LINGUAGENS QUÂNTICAS E A CONEXÃO COM A LINGUAGEM CLÁSSICA

CAIO COSTA N. A. DA SILVA[1]
GEISON FREITAS DA SILVA[2]
MARIANGELA FERREIRA FUENTES MOLINA[3]

RESUMO

Com o objetivo de identificar similaridades entre os modelos computacionais clássico e quântico pelo prisma das linguagens de programação, a presente pesquisa inicialmente buscou compreender os elementos principais do novo contexto computacional quântico incluindo aferir, brevemente, a caminhada histórica da mecânica quântica. Avaliar suas passagens e atores nos permitiu compreender um pouco mais sobre a motivação e aplicação em uma metodologia computacional. Com isso, a proposta em seguida foi de identificar subsídios que conectasse a história das linguagens clássicas as linguagens quânticas, entender como o modelo clássico participa dessa evolução, como influenciou, sobretudo, a primeira linguagem real quântica (QCL - Quantum Computation Language) e estabelecer possíveis heranças estruturais mesmo se tratando de paradigmas que, inicialmente, se comportam de forma tão distintas.

Palavras-chave: Computação Quântica; Linguagem de Programação; Linguagens de Quânticas; Programação Clássica.

ABSTRACT

In order to identify similarities between the classical and quantum computational models in the prism of programming languages, this research initially sought to understand the main elements of the new quantum computational context, including briefly referring to a historical path of quantum mechanics. Valuing your passages and actors allows us to understand a little more about the motivation and application of a computational methodology. As a matter of course, the next step was to identify subsidies that connect to the history of classical languages as quantum languages, to understand how the classical model participates in its evolution, as an influence, above all, to the first quantum real language (QCL - Quantum Computation Language) and establish possible structural inheritances even when dealing with paradigms that, initially, behave in a different way.

Keywords: Quantum Computation; Programming Language; Quantum Languages; Classic Programming.

[1]Graduando, Análise e Desenvolvimento de Sistemas da Faculdade de Tecnologia - FATEC Mogi das Cruzes, SP, Brasil. E-mail: amaro.caio@gmail.com
[2]Graduando, Análise e Desenvolvimento de Sistemas da Faculdade de Tecnologia - FATEC Mogi das Cruzes, SP, Brasil.
[3]Docente, Análise e Desenvolvimento de Sistemas da Faculdade de Tecnologia - FATEC Mogi das Cruzes, SP, Brasil.

INTRODUÇÃO

Um elemento que nos impulsiona é considerar que todo processo científico é provido de um grande esforço coletivo, um imenso volume histórico de pesquisa e, portanto, quando nos deparamos com informações sobre a nova era computacional quântica, estamos “abraçando” quase cento e cinquenta anos de investimentos da física (mecânica quântica) desde Planck (1858-1947) e quase cem anos de pesquisa na área computacional - se considerarmos apenas a partir da geração um. A pretensão desta pesquisa não é fazer um estudo aprofundado a respeito da computação quântica, mas sim identificar aspectos históricos ligados à evolução da linguagem quântica que foram absorvidos de tecnologias anteriores.

O parâmetro de análise aqui aplicado foi, fundamentalmente, a primeira linguagem real de programação quântica QCL em conjunto com um mapeamento dos recursos hoje disponíveis para experimentação quântica denominados “Kit de Desenvolvimento Quântico”. Recursos disponibilizados por diversas organizações como a Microsoft Q#, Amazon Braket, D-Wave Quantum e IBM Experience entre outros.

MATERIAL E MÉTODOS

Com intuito de compreender as similaridades do ponto de vista da semântica e sintaxe, entre as linguagens clássicas e os recursos de programação quânticos, a pesquisa realizou uma imersão no tema proposto utilizando como base para desenvolvimento desse saber, materiais bibliográficos em artigos científicos, publicações científicas diversas e, - para efeito da compreensão prática do objeto mencionado – foram estudados os fundamentos básicos da primeira linguagem de programação quântica QCL e alguns simuladores denominados “Kits de Desenvolvimento Quânticos”.

Importante mencionar que o propósito deste trabalho foi de apenas identificar similaridades estruturais entre o clássico e quântico que apontasse

reutilizações, ou não, da visão tradicional na linguagem quântica QCL, para assim concluir em possíveis heranças de sintaxe, semântica e caminhos que esse novo paradigma indica.

Por fim, porém não menos importante, a presente pesquisa tratou de alinhar três aspectos para o contorno geral do artigo: paisagem histórica da computação como caminho evolutivo conectado, uma breve passagem sobre os pilares quânticos para ilustrar os fundamentos físicos da era quântica e sobretudo a identificação de similaridades de códigos escritos em QCL com o paradigma procedural da linguagem clássica C e com isso identificar elementos que apontem para a transmissão do conhecimento clássico para as linguagens quânticas.

Trajetória da programação

Nessa trajetória é notável o aperfeiçoamento e a complexidade que as linguagens de programação foram submetidas, refletindo naturalmente nos algoritmos com suas sequências lógicas e instruções, sempre mantendo um claro objetivo de realizar os cálculos impossíveis pela mente humana. As linguagens de programação aplicadas à arquitetura computacional a partir de von Neumann até os dias de hoje, iniciam sua materialização a partir da década de 40 com um volume absolutamente incrível de contribuições, por isso vale ressaltar que a lista que segue abaixo representa apenas uma parcela dos esforços somados em quase 80 anos da sua história como menciona Kowaltowski (1996, p.238):

> As grandes invenções tecnológicas dificilmente aparecem de maneira independente. A ideia de automatizar os cálculos vem desde a antiguidade e começou com a utilização de pedras e outros dispositivos que deram origem aos ábacos, progredindo durante vários séculos até o aparecimento de computadores digitais na década de 1940.

Tabela 1. Linha história das linguagens de programação.

Década de 1940	**PLANKALKÜL** 1943/45 – **ENIAC coding system** 1943/46 – **ARC Assembly** 1947 – **CPC** 1948.
Década de 1950	**BIRKBECK ASSEMBLER** 1950 - **FORTRAN** 1954 – **LISP** 1956/58 – **COBOL** 1959.
Década de 1960	**ALGOL** 60 1960 – **SNOBOL** 1962 – **BASIC** 1964 – **EXAPT** 1967
Década de 1970	**PASCAL** 1970 – **PROLOG** 1972 – **SMALLTALK** 1972 - **SCHEME** 1975 – **MODULA** 1975 – **C Shell** 1978
Década de 1980	**ADA80** 1980 – **C with class** 1980 – **IBM BASICA** 1981 - **C++** 1983 – **CLIPPER** 1984 – **MIRANDA** 1986 -**PERL** 1987
Década de 1990	**PYTHON** 1991 – **VISUAL BASIC** 1991 – **LUA** 1993 – **JAVA** 1995 – **PHP** 1995 – **RUBY** 1995 – **JAVASCRIPT** 1995 – **VBScript** 1996 – **QCL 1998** ***primeira linguagem quântica***
Década de 2000	**C#** 2000 – **JOIN JAVA** 2000 - **VB .NET** 2001 – **ADA2005** 2007 - **COBRA** 2007 – **GO** 2009
2010 a 2021	**DART** 2011 – **KOTLIN** 2011 – **CRYSTAL** 2014 – **HACK** 2014 – **Q#** 2017 – **C++20** 2020 - **MS POWER FIX** 2021

Fonte: Rigaux [s.d.].

Trajetória da mecânica quântica

Segundo Chibeni (2008), o nascimento da física quântica ocorre na virada para o século passado e está totalmente associado a um contexto de dificuldade da física clássica para explicar diversos fenômenos. Dentre as limitações estavam o efeito fotoelétrico, espectro discreto dos átomos e moléculas e, a radiação do corpo negro. Em 1900, o físico e matemático alemão Max Karl Ernest Ludwig Planck (1858-1947**)**, propôs que a energia não era contínua, mas sim que a radiação é absorvida ou emitida por um corpo aquecido não sob a forma de ondas, mas por meio de pequenos "pacotes" de energia. Esses pequenos "pacotes" de energia Max Planck deu o nome de quantum (quanta - quantidade elementar de algo); Plank criou uma função que permitia determinar a radiação das partículas oscilantes em um corpo negro (HEWITT, 2011):

$E = n . h . v$

n = número inteiro positivo - h = constante de Planck (6,626 . 10-34 J .) - v = frequência da radiação emitida.

Partindo do princípio de Plank, no ano de 1905, Albert Einsten oferece uma contribuição avigorando o trabalho de Plank. Para elucidar o efeito fotoelétrico, Einsten nos afirmou que a luz deveria ser composta por diminutas partículas

(depois denominadas de fótons) de massa zero. Einsten, com isso, sugere uma natureza corpuscular para radiação, ou seja, natureza dual da radiação (para a luz - para as ondas eletromagnéticas).

Resumo de algumas contribuições da mecânica quântica

Niels Bohr, em 1913, publicou a teoria atômica / espectro de linha (princípio da quantização de Planck). Estudo com base no discreto do átomo de hidrogênio.

Arthur Holly Compton, em 1922, publicou o efeito Comptom.

De Broglie, em 1923, pesquisou o que ficou conhecido como a Natureza Dual da Matéria.

Werner Heisenberg, em 1927, anuncia o Princípio da Incerteza que dá início ao formalismo da teoria quântica. Afirma que é impossível saber a posição exata de um elétron na eletrosfera de um átomo.

Erwin Schordienger, em 1926, publica equação de onda da matéria. Propôs também, em 1933, o experimento conhecido como o Gato de Schrödinger.

Stern e Gerloch em 1922 realizam experimento aplicando um feixe de luz sobre um campo magnético não uniforme, o resultado foi que esse feixe se dividiu em dois, isso explica pela interação do “Spin” com o campo magnético.

Paul Dirac 1928 propõe o que ficou conhecido como a Equação de Dirac. Seu trabalho observou partículas de carga oposta do elétron, o que possibilitou o descobrimento do pósitron (1932 observado por **Carl David Anderson**).

Sin-Itiro Tomonaga, Julian Schwinger e Richard Feynman desenvolveram com base nos princípios de Dirac, a teoria quântica de campos do eletromagnetismo.

Emily Grumbling e Mark Horowits (2019, p. 25) propõem que:

> [...] um objeto quântico geralmente não existe em um estado completamente determinado e conhecível. De fato, cada vez que se observa um objeto quântico, ele se parece com uma partícula, mas quando não está sendo observado, se comporta como uma onda. Essa *chamada* dualidade onda-partícula leva a muitos fenômenos físicos interessantes.

Computação quântica – Conceitos fundamentais

Segundo De Melo e Christofoletti, no início dos anos 1980, Richard Feynman (1982) constata que um computador quântico seria uma ferramenta eficaz para resolver problemas de física e química, dado que é exponencialmente caro simular grandes sistemas quânticos com computadores clássicos; teorizava que a viabilidade para simulação de sistemas quânticos seria possível se realizado em computadores quânticos. Outro renomado cientista, Paul Benioff (1980), um dos precursores do modelo quântico computacional argumentou sobre as possíveis barreiras que a miniaturização dos circuitos integrados apresentava, indicando os possíveis potenciais ganhos no desenvolvimento da computação quântica pelos resultados no processamento e também quanto ao processo físico do computador**,** transversalizando com um debate conexo desencadeado pela publicação do famoso artigo *Cramming More Components Onto Integrated Circuits***,** do autor Gordon E. Moore (resultou na Lei de Moore), na qual questionava a possibilidade de antecipar o que aconteceria na indústria de semicondutores nos próximos 10 anos (até 1975), na qual afirmou:

> A complexidade para um mínimo custo de componentes tem crescido numa taxa que se aproxima do fator dois por ano. Certamente, a curto prazo, espera-se que essa taxa continue, senão aumente. A longo prazo, a taxa de crescimento pode ser um pouco mais incerta, apesar de que não há razão para acreditar que ela permaneça constante por pelo menos 10 anos. (MOORE, 1965, p.2, tradução nossa).

Em 1985 a possibilidade de materializar efetivamente um computador quântico começa receber contornos reais com o cientista David Deutsch, quando ele concebe uma versão da máquina Turing nos princípios quânticos e afirma em seu artigo *Quantum theory, the Church-Turing principle and the universal quantum computer***:**

> Já vimos que o computador quântico universal Q pode simular perfeitamente qualquer máquina de Turing e pode simular com precisão arbitrária qualquer computador ou simulador quântico. (DEUTSCH, 1985, p.10, tradução nossa).

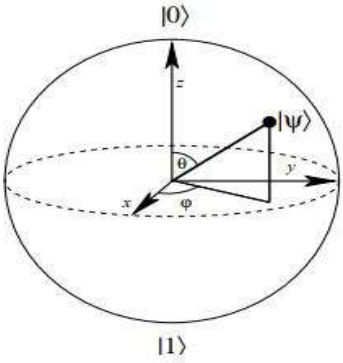
Anos depois outra grande contribuição marca o desencadeamento histórico dos computadores quânticos, só que essa vez no campo dos algoritmos, o cientista Peter Shor (1994) demonstra a possível capacidade do pensamento aplicado a computação quântica com o algoritmo que leva o seu próprio nome,

descrevendo logaritmos discretos e fatoração, provando que - em especial a fatoração - teria desdobramentos muito mais positivos na perspectiva quântica do que na clássica. Em seguida, outro algoritmo chama atenção do mundo científico, o algoritmo quântico de Grover (1996), que permite uma aceleração na busca, resolvendo problemas que a computação clássica precisaria de muito esforço para solucionar.

Qubit e as Portas Quânticas

Para José, Piqueira e Lopes (2013) o computador clássico representa os dados a serem processados como Bit (a menor unidade de valor armazenado ou transmitido por um computador), que significa “Binary Digit”. O bit pode ser alocado como carga elétrica ou por outras formas como polarização magnética e, assume apenas um de dois valores: 0 ou 1. Já o Qubit, denominado “Quantum Bit”, funda uma diferença significativa que amplia a capacidade de processamento exponencialmente para realizar cálculos paralelos, pois ele pode assumir ambos os valores de 0s ou 1s ao mesmo tempo (estado de “superposição”) além dos fenômenos quânticos de emaranhamento (o emaranhamento é uma correlação extremamente forte que existe entre partículas quânticas. Einstein a descrever o emaranhado como "ação assustadora à distância") e encapsulamento. No processo quântico os qubits são compostos por partículas de níveis atômicos e ou subatômicas - elétrons, fótons entre outros.

Figura 1. Bloch Sphere.



Fonte: Omer (2003).

O qubit é essencialmente um vetor que normalmente tem sua representação associada a notação do cientista Dirac, denominada como “braket” (**Exemplo: |ψ > α|*0* + β|1 >).** Resumidamente na abordagem da computação quântica, se olharmos para um único qbit ou qubit podemos dizer que é uma esfera que tem em um dos seus polos o estado 0 e em outro polo o estado 1, porém é possível assumir a posição de qualquer estado dessa esfera, ou seja, qualquer posição na esfera é um estado possível, conforme ela avança do 0 para 1 aumenta a probabilidade de 1.

Desse modo, se sairmos do entendimento de um único qubit e partirmos para compreensão do seu uso na realização computacional, avançaremos para inclusão de novos elementos como as importantes portas lógicas quânticas, que realiza processos de forma bastante similar as portas lógicas clássicas, porém diferencia-se pelo fato de que as portas lógicas quânticas resultam na rotação da esfera mencionada(Qubit), combinando e gerando novos estados como no caso da “gate Hadamard” que possibilita a dita superposição dos estados ou a “Pauli X” que rotaciona em pi (π) para mudar simplesmente o estado 0 para 1 ou vice versa (NIELSEN, CHUANG, 2001).

Hardware Quântico

Sobre a perspectiva do hardware ainda não há computadores quânticos de uso geral muito menos amplamente distribuídos pelo mercado, nossa disponibilidade atual são máquinas quânticas desenvolvidas com arquiteturas diversas que exploram os fundamentos da mecânica quântica visando resolver problemas bastante específicos, geralmente ligados a sistemas quânticos, problemas matemáticos de alta complexidade que a computação clássica não resolve, análises de materiais complexos, cálculos químicos complexos ou para estudos acadêmicos em instituições que possuem frentes de pesquisas quânticas. Vale pontuar que existem também empresas atuando no mercado oferecendo o poder da computação quântica para solucionar problemas complexos e específicos como o caso da empresa canadense D-Wave fundada em 1999, talvez uma das pioneiras no assunto.

Como indicam Grumbling e Horowitz (2019, p. 114):

> [...] os componentes de hardware necessários para um computador quântico analógico ou baseado em portas, o hardware pode ser modelado em quatro camadas abstratas: o "plano de dados quânticos", onde residem os qubits; o "plano de controle e medição", responsável pela execução das operações e medições nos qubits, conforme necessário; o "plano do processador de controle", que determina a sequência de operações e medições que o algoritmo requer, potencialmente usando resultados de medições para informar as operações quânticas subseqüentes; e o "processador host", um computador clássico que lida com acesso a redes, grandes matrizes de armazenamento e interfaces com o usuário.

Atualmente é possível acessar recursos quânticos de algumas formas, grande parte das disposições estão associados a uma emulação tendo como mediador alguma linguagem clássica, como por exemplo o recurso na nuvem da Amazon, Microsoft, IBM ou Google; todos oferecem uma experiência computacional quântica com bibliotecas bem-organizadas, intuitivas e didáticas.

Linguagens quânticas

As linguagens quânticas são classificadas de três formas, as “linguagens imperativas (procedurais)” uma arquitetura já amplamente estudada na computação clássica que influenciou a história das linguagens; as “linguagens funcionais” que possuem funções com foco nas entradas e saídas de dados em contraponto aos princípios das “linguagens imperativas” que altera o estado dos dados e por fim linguagens classificadas como “outros paradigmas” que são estruturas mistas (JOSÈ, PIQUEIRA, LOPES, 2013). Nas investigações foi encontrado uma única pesquisa – bastante recente - ligada ao paradigma orientado objetos denominado “QJava”, o que indica uma ampliação para esse campo muito em breve.

Em 1998 a primeira linguagem de programação legitimamente quântica foi criada pelo cientista da computação Bernhard Omer, denominada QCL (Quantum Computation Language), baseada no paradigma imperativo (procedural) com fortes influências nas sintaxes da linguagem C. A proposta da QCL é ser executada em um computador clássico que por sua vez controla um computador quântico, com possibilidade de o desenvolvedor definir o número de qubits a

serem utilizados (ÖMER, 1998). A QCL contém um número útil de funções quânticas e mantém uma linguagem de programação clássica posta como sub - linguagem.

Tipos de Linguagens Quânticas

Atualmente existem outras linguagens além das listadas nessa pesquisa, mencionaremos algumas que consideramos importantes pelo propósito histórico.

Imperativas

- **QCL:** Primeira Linguagem de programação real criada para computação quântica. Ela é uma linguagem dita híbrida com forte influência da linguagem clássica C;
- **Linguagem Q:** A segunda linguagem quântica imperativa criada para computação quântica. Ela suporta extensões da linguagem clássica C++;
- **Q#:** Linguagem quântica criada pela Microsoft para dar suporte ao projeto “Quantum Development Kit”;
- **QMA-SM:** Conhecida como Quantum Macro Assembler é possível deduzir que se trata de uma linguagem quântica de baixo nível. Utilizada em “annealers” quânticos;

Funcionais

- **QML:** Uma linguagem funcional quântica com forte influência da Haskell;
- **QFC:** Linguagem de programação quântica com controle clássico, mas pode manipular dados clássicos ou quânticos

Bibliotecas Para Desenvolvimento Quântico

Além das linguagens com características imperativas, funcionais ou mistas, existem outras formas de experienciar a computação quântica através das iniciativas denominadas “Kits de Desenvolvimento Quânticos”**.**

- **PennyLane**: Biblioteca com recursos quânticos que recebe a mediação da linguagem Python. Projeto de código aberto (XANADU, 2013);

- **AWS:** Serviço de computação quântica na nuvem para auxílio de pesquisas de algoritmos quânticos com suporte do Python. Conectada com o projeto PennyLane (AMAZON, 2021);
- **Forest:** Biblioteca com objetivo de criar e manipular circuitos quânticos, além de oferece acesso a um computador quântico na nuvem (RIGETTI COMPUTING, 2020);
- **Qiskit** [s.d.]: Projeto ligado a IBM desenvolvido para operar com linguagem Python utilizando simuladores quânticos IBM e possibilita a geração de programas com operações quânticas entre outros recursos;
- **Azure - Quantum Development Kit:** Projeto da Microsoft que opera com .NET Framework possibilitando a criação de programas que podem ser executados VisualStudio e VSCode (MICROSOFT, 2021);
- **Cirq**: Projeto gerenciado pela Google para criar e manipular programas quânticos utilizado o Python como linguagem mediadora (GOOGLE, [s.d.]);
- **Ocean:** Projeto da empresa D-Wave de código aberto para manipular circuitos quânticos utilizado o Python como linguagem mediadora (D_WAVE, 2021).

RESULTADO DA DISCUSSÃO

Para exemplificar a proposta de pesquisa foi incluído um trecho de um código escrito em QCL com um algoritmo denominado “Discrete Fourier Transform” na tradução “Transformada Discreta de Fournier”, um problema encontrado nos estudos de sinais, sistemas de comunicação, sistemas de controle entre outros. Nosso objetivo aqui é apenas demonstrar a característica do algoritmo quântico para efeito de exemplificação e, sobretudo, demonstrar o quanto as linguagens quânticas, apesar de toda especificidade do universo tratado se conecta com toda evolução das linguagens clássicas em uma única trajetória.

É possível notar nos algoritmos escritos em QCL que há uma forte influência das linguagens procedurais, especificamente a linguagem C até mesmo nas funções QCL que realiza as operações quânticas como Hadamard (superposição) e as rotações.

Figura 2. Algoritmo DFT.

```
operator dft(qureg q) { // main operator
const n=#q;             // set n to length of input
int i; int j;           // declare loop counters
for i=0 to n-1 {
  for j=0 to i-1 {      // apply conditional phase gates
    CPhase(2*pi/2^(i-j+1),q[n-i-1] & q[n-j-1]);
  }
  Mix (q[n-i-1]);       // qubit rotation
}
flip(q);                // swap bit order of the output
}
```

Fonte: Shajeemohan [s.d.].

Basicamente, “dft.qcl” contém dois loops: O loop externo executa transformações de Hadamard do qubit mais alto para o mais baixo (Mix), enquanto o loop interno realiza mudanças de fase condicionais (CPhase) entre os qubits (SHAJEEMOHAN, s.d.).

Segundo a documentação do Bernhard Ömer (2000), é possível compreender que a herança da linguagem clássica é fundamental na estrutura da linguagem QCL como é possível observar na sintaxe e semântica da estrutura básica da linguagem.

Funções QCL

Figura 3. Funções QCL.

Trigonometric Funct.		Hyperbolic Funct.	
sin (x)	sine of *x*	sinh (x)	hiperbolic sine of *x*
cos (x)	cosine of *x*	cosh (x)	hyperbolic cosine of *x*
tan (x)	tangente of *x*	tanh (x)	hyperbolic tangente of *x*
cot (x)	cotangente of *x*	coth (x)	hyperbolic contangent of *x*
Complex Funct.		**Exponential an related Funct.**	
Re (z)	real part of *z*	exp (x)	*e* raised to the power of *x*
Im (z)	imaginary part of *z*	log (x)	natural logarithm of x
abs (z)	magnitude of *z*	log (x, n)	base-*n* logarithm of *x*
conj (z)	conjugate of *z*	sqrt (x)	square root of *x*

Fonte: Ömer (2000).

Operadores QCL

Figura 4. Operadores QCL.

Op	Description	Argument type
#	register size	quantum types
^	power integer power	all arithmetic int
-	unary minus	all arithmetic
* /	multiplication division integer division	all arithmetic all arithmetic int
mod	integer modulus	int
+	addition	all arithmetic
-	subtraction	all arithmetic
&	concatenation	string, quantum types
== != < <= > >=	equal unequal less less or equal greater greater or equal	all arithmetic, string all arithmetic, string integer, real int, real int, real int, real
not	logic not	boolean
and	logic and	boolean
or	logic inclusive or	boolean
xor	logic exclusive or	boolean

Fonte: Ömer (2000).

Declaração de Variáveis QCL

Figura 5. Declaração de variáveis QCL.

```
qcl> complex z;          // declare complex variable z
qcl> print z;            // z was initialized with 0
: (0.000000,0.000000);
qcl> z=(0,1);            // setting z to i
qcl> print z;
: (0.000000,1.000000);
qcl> z=exp(z*pi);        // assignment to z may contain z
qcl> print z;
: (-1.000000,0.000000);
qcl> input z;
? complex z [(Re, Im)] ? (0.8,0.6)
qcl> print z;
: (0.800000,0.600000)
```

Fonte: Ömer (2000).

CONCLUSÃO

A experiência dessa pesquisa nutriu nossa compreensão sobre o quanto esses modelos estão efetivamente emaranhados e quanto eles continuam sendo

colaborativos e complementares. Dentro do campo das linguagens de programação o uso das linguagens clássicas são fundamentais para o fomento quântico quando observamos os Kits de Desenvolvimento Quântico; quando analisado as linguagens é possível também perceber que as suas estruturas - do ponto de vista da sintaxe e semântica - são influenciadas pelos paradigmas clássicos e é possível que essa relação se fortaleça ainda mais, já que é plausível afirmar que não há um modelo melhor do que o outro, mas sim computação para fins diferentes e, no estágio atual, é evidente que a computação clássica auxilia profundamente na construção da musculatura do processo computacional quântico, sobretudo, porque empresta seu volume de conhecimento para acessar os fundamentos desse novo modelo que se apresenta como futuro, mas acima de tudo é a riqueza de uma história contínua.

REFERÊNCIAS BIBLIOGRÁFICAS

AMAZON.AWS, c2021. Disponível em:<https://aws.amazon.com/pt/braket/>. Acesso em: 25 de Nov. de 2021.

BENIOFF, Paul. The computer as a physical system: A microscopic quantum mechanical Hamiltonian model of computers as represented by Turing machines. **Journal of statistical physics**, v. 22, n. 5, p. 563-591, 1980.

CHIBENI, SILVIO SENO. O surgimento da física quântica. **Campinas: Departamento de Filosofia-UNICAMP**, p. 1-8, 2008.

DE MELO, Bruno Leonardo Martins; CHRISTOFOLETTI, Túlio Vinícius Duarte. **Computação Quântica: Estado da Arte**, s.d.

D-WAVE. Ocean. c2021. Disponível em: <https://www.dwavesys.com/solutions-and-products/ocean/>. Acesso em 27 de Nov. 2021.

FEYNMAN R.P. Simulating physics with computers. **Int. J. Theor. Phys**., 21:467, 1982.

GOOGLE. Cirq: Quantum AI. s.d. Disponível em: <https://https://quantumai.google/>. Acesso em: 19 de Nov. de 2021.

HEWITT, Paul G. **Física Conceitual**. Bookman 11ª edição. Porto Alegre, 2011.

JOSÉ, Marcelo Archanjo; PIQUEIRA, José Roberto Castilho; LOPES, Roseli de Deus. Introdução à programação quântica. **Revista Brasileira de Ensino de Física**, v. 35, p. 1-9, 2013.

KOWALTOWSKI, Tomasz. Von Neumann: suas contribuições à Computação. **Estudos Avançados**, v. 10, n. 26, p. 237-260, 1996.

MCGEOCH, C. C. Adiabatic quantum computation and quantum annealing: Theory and practice. Synthesis Lectures on Quantum Computing, **Morgan & Claypool Publishers**, v. 5, n. 2, p. 1–93, 2014.

MICROSOFT. Azure, c2021. Disponível em: <https://azure.microsoft.com/>. Acesso em: 12 de Nov. de 2021;
MOORE, G. E. Cramming more components onto integrated circuits**. Eletronics**, 38(8), 114-117, 1965.

NIELSEN, Michael A.; CHUANG, Isaac L. Quantum computation and quantum information. **Phys. Today**, v. 54, n. 2, p. 60, 2001.

ÖMER, Bernhard**. Quantum programming in QCL**. Dissertação de Mestrado, Institute of Information Systems Technical University of Vienna, 2000.
ÖMER, Bernhard**. Structured Quantum Programming.** PhD thesis, Technical University of Vienna, 2003.

QISKIT. Qiskit, s.d. Disponível em: <https://qiskit.org/>. Acesso em: 17 de Nov. de 2021.

RIGAUX. Rigaux. s.d. Disponível em: <http://rigaux.org/language-study/diagram.html>. Acesso em: 17 de Nov. de 2021.

RIGETTI COMPUTING. Forest, c2020. Disponível em: <https://www.rigetti.com/>. Acesso em: 27 de Nov. de 2021.

ROSA, Evandro Chagas Ribeiro da et al. **QSystem: simulador quântico para Python**. 2019.

SHAJEEMOHAN, B. S. **QCL œ A Programming Language for Quantum computers**. s.d.

XANADU. Pennylane, c2019. Disponível em: <https://https://pennylane.ai/>. Acesso em: 27 de Nov. de 2021.