

PROTOCOLO DE CODIFICAÇÃO SUPERDENSE IMPLEMENTADA EM Q#

João Ataíde¹, Gustavo Leony²; José Vicente Cardoso Santos³

¹ Centro Universitário Jorge Amado (UNIJORGE) Graduando em Ciência da Computação

² Centro Universitário Jorge Amado (UNIJORGE) Graduando em Ciência da Computação

³ Centro Universitário Jorge Amado (UNIJORGE) Professor do Curso Ciência da Computação

Abstract. Today's computer architecture has limitations, with this imminent danger of becoming stagnant and obsolete, as thermocomputing grows stronger. Using the concepts of quantum physics, a new type of machine appears, the Quantum Computer, with a capacity immeasurably superior to current machines and with an innovative architecture concept. That said, this article has the general objective of describing the implementation logic of a quantum algorithm, explain how its numerical sets work, and how a quantum computer behaves, using an algorithm that makes use of the properties of the quantum bit described throughout the article, this algorithm called "Superdense Coding Protocol", is a classic quantum computing algorithm used to exchange two classical bits for the price of one quantum bit. The methodology used in the production of this bibliographic, we searched in articles published on the WEB, information about the need to maintain quality, its advantages over traditional maintenance, and what use it makes.

Resumo. A arquitetura de computadores atual possui limitação, com esse perigo iminente de se tornar estagnada e obsoleta, cada vez mais o termo Computação Quântica vem ganhando força. Utilizando os conceitos da física quântica, um novo tipo de máquina surge, o Computador Quântico, com uma capacidade imensuravelmente superior as máquinas atuais e com um conceito de arquitetura inovador. Dito isto, este artigo tem como objetivo geral descrever a lógica de implementação em um computador quântico, e como objetivos específicos explicar como funciona seus conjuntos numéricos e como um computador quântico se comporta. utilizando um algoritmo que faça uso das propriedades do bit quântico descritas ao longo do artigo, algoritmo esse denominado de "Protocolo de codificação superdensa", é um algoritmo clássico da computação quântica utilizado para trocar dois bits clássicos pelo preço de um bit quântico. A metodologia utilizada na produção deste artigo foi a metodologia bibliográfica, na qual buscamos em artigos publicados na WEB, informações sobre a necessidade do surgimento da computação quântica, suas vantagens sobre a computação clássica, e quais propriedades o qubit faz uso.

Palavras-chave: Arquitetura; Computação Quântica; Física Quântica; Bit Quântico; Protocolo de codificação superdensa.

1. Introdução

Com a utilização da Computação Quântica, a redução de tempo necessário para executar certas tarefas é de tal ordem que alguns problemas que levariam tempos impraticáveis em supercomputadores podem ser resolvidos em tempos normais em computadores quânticos, ou seja, problemas praticamente insolúveis para computação clássica passam a ser solúveis em computação quântica.[ALVES2003].

Além disso, o estudo da Computação Quântica, sob as lentes da Teoria da Computação, podem trazer novos resultados para ela mesma. O estudo de um problema sob o ponto de vista de um novo modelo computacional pode trazer novas ideias para solucioná-lo sem tais recursos. [BREDARIOL 2014]. Ou seja, impulsionar o avanço da computação quântica, acarreta em benefícios para a própria computação atual.

Como a computação clássica usa os valores 1 e 0, conhecidos como bits para sua comunicação, os computadores quânticos também se baseiam nele, porém como bits quânticos chamados de Qubits. Essa tecnologia oferta uma revolução na segurança no ambiente digital, em virtude que o computador quântico está apto a descriptografar dados, possibilitando uma maior segurança das informações, diminuindo riscos e oferecendo mais segurança contra violações.

Uma das dificuldades sobre ter acesso a esta tecnologia é o propósito da sua aplicação e o público alvo que usufruía-la. As alternativas presentes estão ao proporcionar competição cibernética, renovação em seções da computação como Inteligência Artificial, Machine Learning, criptografia.

No decorrer deste trabalho, será explicado como um computador quântico se comporta, suas bases numéricas e sua programação.

2. Objetivos Geral e Específicos

Este artigo tem como objetivo geral descrever a lógica de implementação de um algoritmo quântico e os objetivos específicos consistem em detalhar o funcionamento dos seus conjuntos numéricos e os comportamentos de um computador quântico.

3. Metodologia

A metodologia bibliográfica foi a escolhida para o desenvolvimento deste trabalho, pois foram feitas pesquisas em artigos científicos, monografias, dissertações, e livros sobre o tema, a fim de aprimorar o conhecimento da necessidade da criação da computação quântica, sua base de concepção, seus comportamentos e diferenças quanto a computação clássica. Logo após ter compreendido suas propriedades, tais quais destacam-se a sobreposição e o emaranhamento, foi aplicado um algoritmo quântico usando o Protocolo de Codificação Superdensa, para provar na prática que o ganho do uso da computação quântica é largamente maior que o uso da computação clássica.

4. Justificativa

Fomentar o interesse pela computação quântica e suas competências, a fim de entender gradativamente o desenvolvimento do Protocolo de Codificação Superdensa com a explicação do ganho de recursos computacionais a respeito da computação clássica, aplica-se um algoritmo quântico junto com a IDE bastante conhecida e utilizada, o Visual Studio Code, com o intuito de esclarecer que a computação quântica está mais próxima e acessível do que parece ser.

No momento atual, o conceito existente de computador é baseado na arquitetura de Von Neumann, que afirma serem necessários, memória, processador, dispositivos de entrada e saída de dados e barramento de comunicação para ter-se um computador [LOMONACO 2002].

Utilizando o poder dos computadores quânticos foram desenvolvidos vários algoritmos. O mais relevante deles foi criado pelo pesquisador Peter Shor em 1994, afim de solucionar o problema da fatoração de números inteiros extensos. O algoritmo faz uso da sobreposição quântica para diminuir o tempo da solução do problema. Amostra do algoritmo destacou-se na sociedade científica, por poder usar a mecânica quântica na computação e possibilitar a quebra de certas chaves criptográficas [LOMONACO 2002].

Sendo assim, criptografias atuais que levariam cerca de bilhões de anos para serem quebradas, com a utilização de um computador quântico, levariam muito menos tempo para serem decifradas.

Segundo [ALVES 2003] um computador convencional precisaria executar por bilhões de anos para fatorar um número de 400 dígitos. Uma máquina quântica poderia fazer o serviço em cerca de um ano.

Uma propriedade da computação quântica de extrema importância é o bit quântico, chamado também de Qubit, que consegue assumir os bits 0 e 1 ao mesmo tempo. Segundo [MATTIELO et al. 2012] No entanto, se as moedas se comportassem como os objetos microscópicos, que obedecem aos princípios da mecânica quântica, as faces cara e coroa poderiam ser vistas ao mesmo tempo, ou seja, no lançamento de uma moeda, o resultado cara e coroa coexistiriam.

Esse ganho computacional só é possível devido a uma propriedade da mecânica quântica, a sobreposição. Segundo [MATTIELO et al. 2012] Se novamente, uma moeda é um objeto quântico, os resultados '1' e '0' são denominados de estados. [RIEFFEL 2000].

O estado de um qubit é um vetor unitário em seu espaço de estados $E = C^2$. Em geral, utilizamos a base formada pelos vetores $|0\rangle = (1, 0)$ e $|1\rangle = (0, 1)$. Assim, que $|\alpha|^2 + |\beta|^2 = 1$ para que $|\varphi\rangle$ seja um vetor unitário. Mas também podemos usar a base ortonormal $|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ e $|-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$. Dessa forma seu estado seria dado por $|\psi\rangle = \delta|+\rangle + \gamma|-\rangle$. E para que $|\psi\rangle$ tenha norma 1, exigimos que $|\delta|^2 + |\gamma|^2 = 1$ [ADEMAR et al. 2020].

Se o estado do sistema é $\varphi\rangle = \alpha|0\rangle + \beta|1\rangle$, a medição $\{|0\rangle, |1\rangle\}$ resulta: 0 com probabilidade $|\alpha|^2$, 1 com probabilidade $|\beta|^2$ [GOZI et al. 2019]. Ou seja, o bit quântico trabalha com possibilidades, ele não fornece um resultado exato, num cenário de previsões o qubit fornecerá uma taxa X de possibilidade de tal evento ocorrer ou não.

Para realçar a conexão próxima do estado citado acima, Rigolin afirma que [RIGOLIN 2008].

Este emaranhamento é fundamentado principalmente a não localidade de Bell. Segundo [GOMES 2018], seu conceito está intimamente ligado à não- localidade de Bell, de tal maneira que é preciso que um estado quântico seja necessariamente emaranhado para observarmos correlações não-locais associadas a ele.

6. Protocolo de Codificação Superdensa Implementado em Q#

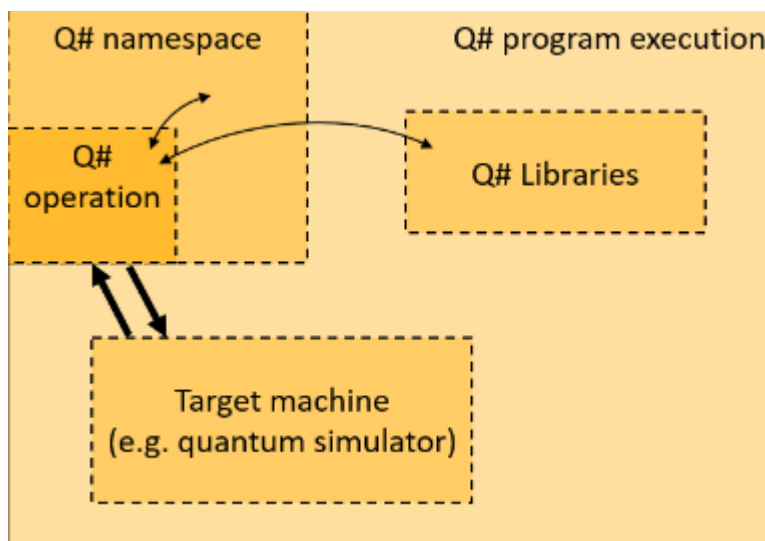
Neste capítulo será abordado o desenvolvimento do protocolo de codificação superdensa, como citado acima, este protocolo tem como objetivo fazer uma comunicação quântica, onde dois indivíduos devem trocar dois bits de informação, no qual cada um possuindo um dos dois qubits, utilizando um emaranhamento quântico e usa como base o protocolo de teletransporte quântico.

O princípio da codificação Superdensa descreve, por dedução, que um canal quântico não permite transmitir mais informações por qubit do que um canal convencional e, portanto, deve-se passar dois qubits para transmitir e codificar a mensagem, devido a importância do estado emaranhado, no qual é capaz transformar dois bits da computação clássica em um único bit quântico.

A linguagem de programação *open-source* Q# é bastante usada para algoritmos quânticos. Foi programada para não necessitar dos componentes do hardware e aprimorar a execução direcionada a toda gama de aplicativos quantum. Além do mais, a Linguagem Q# oferecer auxílio na anexação da computação clássica avançada e da quântica. Na figura 1, consta o diagrama de funcionamento de um programa implementado em Q#:

Figura 1: Diagrama

Fonte: Microsoft



O software foi desenvolvido utilizando a plataforma Visual Studio Code, juntamente com a utilização da extensão *Microsoft Quantum Development Kit*, para que fosse possível desenvolver na linguagem Q#, que trata-se de uma linguagem quântica criada pela Microsoft. Para a construção do projeto foram necessárias a utilização das bibliotecas *Microsoft.Quantum.Intrinsic* e *Microsoft.Quantum.Measurement*.

O desenvolvimento foi dividido em quatro partes:

Passo 1: deverá ser criado o par entrelaçado de bits quânticos que serão enviados os dois bits clássicos;

Passo 2: mandar a mensagem;

Passo 3: decodificar a mensagem enviada;

Passo 4: unir os passos anteriores e criar o protocolo superdenso.

Na figura 2, as bibliotecas necessárias para a construção do projeto foram adicionadas da forma abaixo, a biblioteca *microsoft.quantum.intrinsic* serve para auxiliar quanto a aplicação de portas quânticas e realizar a função de chamada de mensagem. Já a biblioteca *microsoft.quantum.measurement*, possui operações para realizar medições além do básico oferecido pela *quantum.intrinsic*.

Figura 2: Bibliotecas Fonte: Autoria Própria

```
open microsoft.quantum.intrinsic;  
open microsoft.quantum.measurement;
```

Na figura 3, foi realizado o passo 1 descrito acima, mas antes disto foi realizada a criação do *namespace*, geralmente todo arquivo em Q# é iniciado com um *namespace*, ele é um organizador das funcionalidades relacionadas, e seu uso se torna muito importante ao utilizar bibliotecas Q#.

No passo 1, foi realizada a operação de criação dos dois qubits, o q1 e q2, e o seu emaranhamento realizado através da porta Hadamard(H). Esta porta é utilizada para permitir que o qubit entre em estado de sobreposição. Os dois qubits iniciam no estado de Bell $|\Phi^+\rangle$.

O comando "CNOT(q1, q2);" é utilizado para aplicar a porta lógica CNOT (Controlled-NOT), esta porta atua como a porta lógica clássica NOT e é usada com o primeiro qubit como controle e o segundo como alvo, mas somente se o primeiro estiver no estado $|1\rangle$, caso contrário não realiza alterações.

Figura 3: Emaranhamento Fonte: Autoria Própria

```
namespace protocolo.codificacao.superdensa {  
    open Microsoft.Quantum.Intrinsic;  
    open Microsoft.Quantum.Measurement;  
    //Tarefa 1 - Emaranhamento  
    operation ParEmaranhado (q1: Qubit, q2: Qubit) : Unit {  
        H(q1);  
        CNOT(q1, q2); // com isso, tem-se o estado:  $(|00\rangle + |11\rangle) / \sqrt{2}$   
    }
```

Na figura 4, foi realizado o passo 2, também conhecido como tarefa de Alice, é onde ocorre o envio da mensagem, esta etapa inicia-se com o par de qubit emaranhados no estado quântico $|\Phi^+\rangle = (|00\rangle + |11\rangle)/\sqrt{2}$, construída a partir da base produtoppor operações unitárias sobre os qubits. Logo após Alice decidir que mensagemquer enviar, basicamente, ela aplica uma operação unitária em seu qubit de transformaçãoe manda para Bob. Em seguida, obtêm-se o seguinte resultado

```
//Tarefa 2 - Envio da mensagem(Alice)
operation CodificarMensagem(qAlice : Qubit, message : ProtocolMessage) : Unit {
    if (message::Bit1) {
        Z(qAlice);
    }

    if (message::Bit2) {
        X(qAlice);
    }
}
```

Figura 4: Codificação Fonte: Autoria Própria

"00"tal como I e $|\Phi^+\rangle = (|00\rangle + |11\rangle)/\sqrt{2}$ "01"tal como X e $|\Psi^+\rangle = (|01\rangle + |10\rangle)/\sqrt{2}$ "10" tal como Z e $|\Phi^-\rangle = (|00\rangle - |11\rangle)/\sqrt{2}$ "11" tal como Y e $|\Psi^-\rangle = (|01\rangle - |10\rangle)/\sqrt{2}$

Na figura 5, foi realizado o passo 3, também conhecido como tarefa de Bob, é onde ocorre a decodificação da mensagem enviada por Alice. Sendo necessário recuperar o estado mantendo a ordem dos bits. O mais interessante e que não é possível decodificar mais de um bit de informação em apenas um qubit, o que ocorre é que no momento em que Bob recebe a mensagem de Alice codificada, esta troca de mãos faz com que o qubit consigarealizar a codificação de dois bits clássicos de informações com um único qubit.

Figura 5: DecodificaçãoFonte: Autoria Própria

```
//Tarefa 3 - Decodificar a mensagem do passo anterior

operation DecodificarMensagem(qAlice : Qubit, qBob : Qubit) : ProtocolMessage {
    Adjoint ParEmaranhado(qAlice, qBob);
    return MensagemCodificada(MResetZ(qAlice) == Um, MResetZ(qBob) == Um);
}
```

Na figura 6, foi realizado o passo 4, é um compilado dos passos anteriores, é necessário realizar uma alocação temporária para os qubits possam ser executados. É realizado novamente o passo 1, emaranhando os qubits e Bob e Alice recebendo um dos pares cada, novamente o passo 2, onde Alice consegue codificar a mensagem de dois bits clássicos, manipulando o próprio qubit que influência o estado quântico do par emaranhado, e envia para Bob, acionando assim o passo 3, onde nesta troca de mãos Bob mede o qubit de Alice e o seu, e consegue decodificar a mensagem.

Figura 6: Protocolo de Codificação SuperdensaFonte: Autoria Própria

```
operation ProtocoloSuperdense (message : ProtocolMessage) : ProtocolMessage {
    use (q1, q2) = (Qubit(), Qubit());
    ParEmaranhado (q1, q2); //Passo 1 novamente
    CodificarMensagem (q1, mensagem); //Passo 2 novamente
    return ProtocolMessage (q1, q2); //Passo 3 novamente
}
```

Para que seja possível compilar e executar o protocolo, que possui a extensão. QS, é necessário utilizar um arquivo HOST escrito em Python, importando o *namespace* e realizando a chamada das operações, e para que seja possível utilizar um arquivo Python como HOST e necessário importar o pacote QSHARP através do gerenciador de pacotes do Python(PIP) bem como o pacote IQ#.

Figura 7: Hostpy Fonte: Autoria Própria

```
p > quantum2 > host.py
import qsharp

from protocolo.codificacao.superdensa import ParEmaranhado
from protocolo.codificacao.superdensa import CodificarMensagem
from protocolo.codificacao.superdensa import DecodificarMensagem
from protocolo.codificacao.superdensa import ProtocoloSuperdense

ParEmaranhado.Simulate()
CodificarMensagem.Simulate()
DecodificarMensagem.Simulate()
ProtocoloSuperdense.Simulate()
```

7. Considerações Finais

Ao longo deste artigo foi possível observar a necessidade do advento da computação quântica, suas influências e importância na esfera computacional contemporânea.

Para a ciência, é bastante promissor o surgimento das novas tendências tecnológicas, dentre elas a computação quântica, destacando-se suas ferramentas de implementação lógica, dado que o principal objetivo deste é possibilitar maiores processamentos de informações que um computador clássico.

Por ora a concretização da aplicação de um computador quântico trata-se da alternativa mais viável, é esperado que este tome ascensão futuramente, em razão do seu objetivo principal não ser efetivamente a substituição dos modelos disponíveis no mercado de consumo comum.

Sanando o objetivo geral e os específicos, demonstra-se inequivocamente o comportamento dos computadores quânticos, as suas bases numéricas, que são fundamentadas na sobreposição e no emaranhamento, conforme explanado. A lógica de implementação e os ganhos reais do qubit foram vistos através do projeto prático, traduzido na construção do algoritmo “protocolo de codificação superdensa” efetuando a transformação de dois bits convencionais em um único bit quântico.

8. Referências Bibliográficas

ADEMAR, C., BIZ, A., SCOTTI, A., and PEZZI, D. (2020). Computação quântica.

ALVES, F. (2003). Computação quântica: Fundamentos físicos e perspectivas.

BREDARIOL, A. (2014). Computação quântica e teoria da computação. <https://abgrilo.org/master.pdf>.

BURTON, A., ESPINOZA, R., KUZMIN, R., GRANADE, C., Prawiroat-modjo., G., and PAZ, A. (2022). Microsoft.quantum.intrinsic namespace. <https://learn.microsoft.com/pt-br/qsharp/api/qsharp/microsoft.quantum.intrinsic>.

ESPINOZA, R., BURTON, A., KUZMIN, R., GRANADE, C., GUEN, P., and PAZ, A. (2022). Microsoft.quantum.measurement namespace. <https://learn.microsoft.com/en-us/qsharp/api/qsharp/microsoft.quantum.measurementdescription>.

GOMES, V. d. S. (2018). Não-localidade baseada em realismo. Tese(Doutorado) — Universidade Federal do Paraná, Curitiba. ^

GOZI, G., HOFSTATTER, B., ANDRADE, K., and OLIVEIRA, M. (2019). A matemática de um bit quântico. <https://econtents.bc.unicamp.br/eventos/index.php/pibic/article/view/2962/2955>.

LOMONACO, S. J. (2002). A rosetta stone for quantum mechanics with an introduction to quantum computation. Washington, DC volume 58 of Proc. Sympos. Appl. Math.,pages 3-65. Amer. Math. Soc., Providence, RI, 2002.

MATTIELO, F., GOMES, G., AMORIM, R., and BARBOSA, W. (2012). Decifrando a computação quântica. Microsoft (2022). Maneiras de executar um programa em q#. <https://learn.microsoft.com/pt-br/azure/quantum/user-guide/host?programs?source=recommendationstabs=tabid-python>.

OLIVEIRA, I. S., AZEVEDO, E. R., and FREITAS, J. C. (2003). C. de computação quântica. ACM Computing Surveys,Vol 32, No.3, September 2000, pp.300-335. ^

RIEFFEL, E. W. (2000). P. an introduction to quantum computing for non- physicists. ACM Computing Surveys,Vol 32, No.3, September 2000, pp.300-335.

RIGOLIN, G. (2008). Emaranhamento quântico. Instituto de Física “GlebWataghin”, Universidade Estadual de Campinas,Unicamp, 13083-970, Campinas, São Paulo, Brasil.

UniversosQuanticos (2018). Serie: Portas lógicas quânticas 7 – cnot gate.
<https://universosquanticos.wordpress.com/2018/05/11/serie-portas-logicas-quanticas7-gate/>.

Vaughan, D. (2019). Mensagens quânticas com q# e blazor.
<https://learn.microsoft.com/pt-br/archive/msdn-magazine/2019/september/quantum-computing-quantum-messaging-with-qsharp-and-blazor>