

# Implementação de Computação Tolerante a Falhas na Plataforma de Programação Quântica Ket

Gabriel R. Cervi<sup>1</sup>, Jerusa Marchi<sup>1</sup>, Evandro C. R. da Rosa<sup>1</sup>

<sup>1</sup> Grupo de Computação Quântica - GCQ - UFSC  
Departamento de Informática e Estatística  
Universidade Federal de Santa Catarina (UFSC) – Florianópolis, SC – Brazil  
{gabriel.cervi@grad., jerusa.marchi@, evandro.crr@posgrad.}ufsc.br

**Abstract.** *The development of techniques for fault-tolerant computing is essential in the context of quantum computing when we consider real-world systems, which are highly sensitive to noise and, consequently, susceptible to errors. Incorporating these techniques into quantum compilers allows for increased reliability of results obtained of algorithm executions, in a transparent manner to the programmer. In this context, this project aims to study and implement quantum error correction mechanisms in the Ket quantum programming platform, starting with the Steane code implementation and aiming to expand support for fault-tolerant computing techniques within the platform.*

**Resumo.** *O desenvolvimento de técnicas para realizar computação tolerante a falhas é essencial dentro do contexto de computação quântica ao consideramos os sistemas reais, que são altamente sensíveis a ruídos, e consequentemente, suscetíveis a erros. A incorporação dessas técnicas em compiladores para as máquinas quânticas permite aumentar a confiabilidade dos resultados obtidos na execução de algoritmos, de forma transparente ao programador. Nesse contexto, este projeto tem como objetivo estudar e implementar mecanismos de correção de erros quânticos na plataforma de programação quântica Ket, iniciando pela implementação do código de Steane e visando a ampliação do suporte às técnicas de computação tolerante a falhas na plataforma.*

## 1. Introdução

Tecnologias atuais para a implementação de computadores quânticos são altamente sujeitas a ruídos, tornando-se cada vez mais imperativo o uso de técnicas de computação tolerante a falhas. Embora simuladores permitam obter resultados conforme o esperado, o mesmo não se aplica aos sistemas físicos reais. A alta sensibilidade desses dispositivos a ruídos externos os torna extremamente vulneráveis a erros, e portanto, a implementação de mecanismos de tolerância a falhas é fundamental para mitigar esses efeitos e viabilizar a computação quântica prática. [Nielsen and Chuang 2010, §10]

O uso de computação tolerante a falhas confere maior confiabilidade nos resultados obtidos ao executar algoritmos. Além disso, quando integrada diretamente aos compiladores, essa proteção pode ser aplicada de maneira automatizada. Isso garante que a robustez do sistema seja aumentada de forma transparente, sem que o programador precise alterar manualmente a lógica do seu código para se adaptar às camadas de correção.

A partir dessas premissas, este trabalho dedica-se à implementação de mecanismos de correção de erros quânticos na plataforma Ket, projeto reconhecido com o Selo de Inovação da SBC em 2025. O trabalho, que se encontra em sua fase inicial, utiliza o código de Steane primariamente como um modelo de validação pedagógica e arquitetural. Este mapeamento teórico serve para consolidar os requisitos de infraestrutura da plataforma Ket, pavimentando o caminho para a futura inclusão de simulações com modelos de ruído e suporte a arquiteturas de códigos de superfície.

## 2. Correção de Erros Quânticos

A correção de erros clássica, embora robusta [Patterson et al. 2014], é de transposição inviável para o domínio quântico devido a restrições fundamentais. Enquanto sistemas clássicos lidam apenas com inversões discretas de bits (bit-flip), o cenário quântico impõe erros contínuos via superposição e a vulnerabilidade exclusiva do erro de fase (phase-flip). Adicionalmente, o teorema da não-clonagem e o colapso do estado quântico impedem a medição direta dos qubits, exigindo métodos para detectar erros sem observar o dado codificado [Nielsen and Chuang 2010, §10.1].

A solução fundamental para esses obstáculos reside no uso de estabilizadores: um grupo de operações que, quando aplicados a um estado quântico específico não altera o mesmo. Quando integrados com qubits auxiliares, esses operadores permitem extrair a chamada “síndrome do erro”, a qual viabiliza a identificação de erros de maneira indireta.

Uma das implementações mais emblemáticas do uso de estabilizadores é o código de Steane [Nielsen and Chuang 2010, §10.4.2], proposto originalmente em 1996. O código utiliza a estrutura do código clássico de Hamming, uma técnica de correção de erros que utiliza três bits de paridade para proteger quatro bits de dados, para mapear um único qubit lógico em sete qubits físicos. Essa arquitetura, fundamentada em códigos lineares clássicos, permite tratar erros de inversão de bit e de fase de maneira independente, simplificando os processos de detecção e correção de erros sem a necessidade de operações complexas entre os tipos de falha.

Essa independência é refletida diretamente na organização de seus operadores: o código de Steane é composto por um conjunto de estabilizadores  $Z$ , responsáveis por detectar inversões de bit (bit-flips), e um conjunto simétrico de estabilizadores  $X$ , dedicados à verificação de inversões de fase (phase-flips). Essa simetria não apenas simplifica a detecção, mas estabelece a base para as operações tolerantes a falhas. Formalmente, os seis geradores independentes que definem o grupo de estabilizadores são:

$$\begin{aligned}
 S_1 &= IIIXXXX & S_4 &= IIIZZZZ \\
 S_2 &= IXXIIXX & S_5 &= IZZIIZZ \\
 S_3 &= XIXIXIX & S_6 &= ZIZIZIZ
 \end{aligned} \tag{1}$$

A partir dessa definição, os estados  $|0\rangle_L$  e  $|1\rangle_L$  do subespaço lógico, detalhados em (2) e (3) são construídos de modo que a aplicação de qualquer estabilizador  $S_i$  não altere a informação lógica codificada. Isso ocorre porque os estados lógicos são auto-estados de todos os geradores do grupo, com autovalor +1. Fisicamente, a aplicação de um estabilizador promove apenas uma permutação ou troca entre os estados físicos internos, mantendo o estado global invariante. Essa propriedade permite que a verificação da síndrome seja realizada continuamente sem corromper a informação quântica.

$$|0\rangle_L = \frac{1}{\sqrt{8}}(|0000000\rangle + |1010101\rangle + |0110011\rangle + |1101100\rangle + |0001111\rangle + |1011010\rangle + |0111100\rangle + |1101001\rangle) \quad (2) \quad |1\rangle_L = X^{\otimes 7} |0\rangle_L \quad (3)$$

### 3. Codificação, Verificação e Correção

Há diversas maneiras de codificar os qubits físicos para o qubit lógico, uma delas é o método de projeção por estabilizadores, que possui alta extensibilidade para outros códigos quânticos. O processo consiste em iniciar os sete qubits físicos no estado  $|0\rangle^{\otimes 7}$  e utilizar qubits auxiliares em superposição para aplicar operações controladas baseadas nos geradores do grupo de estabilizadores. Esse procedimento força os qubits físicos a assumirem as correlações exigidas pelo código de Steane, consolidando o estado lógico. A decodificação segue o processo inverso para recuperar a informação original.

Para a realização da síndrome do erro, utiliza-se um processo semelhante ao de projeção inicial: usam-se qubits auxiliares para aplicar as operações controladas dos estabilizadores, agora seguidas pela medição desses auxiliares nos eixos computacionais. Conforme pode ser observado na Figura 1 para a síndrome de *phase-flip*, os resultados  $M_1$  a  $M_3$  compõem a síndrome, onde o valor 1 indica a detecção de uma inconsistência de paridade e 0 a sua ausência. Para o bit-flip, realizam-se operações análogas, alterando apenas os estabilizadores e a operação de correção para  $X_j$ . A localização exata do erro é determinada pelas fórmulas (4) e (5), onde o índice  $i$  identifica o qubit físico que deve receber uma porta de correção  $Z$ , enquanto  $j$  define o alvo da porta  $X$ . Esta estratégia tem como base o fato dos erros anti-comutarem com determinados estabilizadores, o que inverte o autovalor do estado para -1 e permite a detecção das falhas via medição.

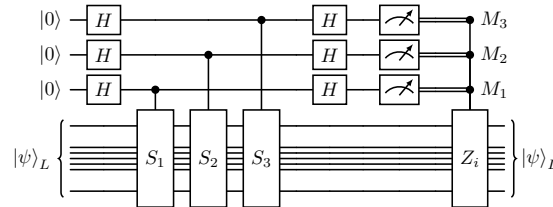


Figura 1. Circuito de síndrome e correção de erro de fase.

$$i = 1 \cdot M_3 + 2 \cdot M_2 + 4 \cdot M_1 \quad (4) \quad j = 1 \cdot M_6 + 2 \cdot M_5 + 4 \cdot M_4 \quad (5)$$

### 4. Computação Tolerante a Falhas

Uma vez que a informação encontra-se codificada no estado lógico, a manutenção da tolerância a falhas exige que as portas quânticas sejam adaptadas para suas respectivas implementações lógicas. Cumpre ressaltar que a aplicação transversal de portas e a extração de síndromes isoladas constituem apenas blocos fundamentais; a computação tolerante a falhas estrita exigirá, em etapas subsequentes do projeto, a modelagem de ciclos repetidos de medição, circuitos de preparação de ancillas protegidas e decodificadores robustos à propagação de erros. No contexto do código de Steane, as operações são categorizadas em dois grupos fundamentais: as transformações pertencentes ao grupo de Clifford e as operações não-Clifford. Essa distinção é crucial, pois a complexidade de implementação varia significativamente entre essas duas classes de portas.

O grupo de Clifford é definido pelo conjunto gerador  $\{H, S, \text{CNOT}\}$ , e sua relevância para o código de Steane reside na propriedade da transversalidade. Portas transversais são inerentemente tolerantes a falhas, pois sua versão lógica é decomposta na aplicação individual da porta em cada um dos sete qubits físicos, impedindo que erros em um único qubit se propaguem para outros dentro do mesmo bloco lógico. Por exemplo a porta  $X$  lógica ( $X_L$ ), no código de Steane equivale à operação  $X^{\otimes 7}$  nos qubits físicos.

Apesar das vantagens das portas transversais, o teorema de Eastin-Knill demonstra que a transversalidade por si só não é suficiente para universalidade [Eastin and Knill 2009]. Para implementar operações fora do conjunto de Clifford, como a porta  $T$ , utilizam-se outros métodos, sendo o mais comum a injeção de estados mágicos [Bravyi and Kitaev 2005], fundamentada no teletransporte de portas representado na Figura 2 para a porta  $T$ . Para preparar o estado  $T|+\rangle_L$ , usa-se a chamada destilação de estados mágicos, técnica que extrai estados mágicos de alta fidelidade a partir de fontes ruidosas, garantindo a universalidade do conjunto de portas.

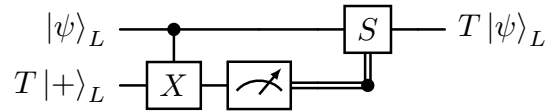


Figura 2. Injeção da porta  $T$  em um estado lógico  $|\psi\rangle_L$

## 5. Considerações Finais

Visto que este projeto encontra-se em suas etapas iniciais, há ainda uma vasta fundamentação a ser explorada. Os próximos passos delineiam-se pelo aprofundamento nos protocolos de destilação de estados mágicos, dada a complexidade e especificidade inerentes a cada estado, seguidos pela investigação da comutação de códigos. Subsequentemente, será realizada uma análise de viabilidade técnica de cada abordagem no âmbito da computação tolerante a falhas utilizando o código de Steane.

Uma vez consolidados esses estudos, o foco voltará-se à implementação na plataforma Ket, com o objetivo de introduzir funcionalidades de correção de erros provendo uma base sólida para a futura expansão rumo a outras arquiteturas e técnicas de computação quântica resilientes a ruído.

**Agradecimento:** Registra-se o agradecimento à Fundação Amparo à Pesquisa e Inovação do Estado de Santa Catarina (FAPESC) pelo apoio financeiro através do projeto 2024TR002672 e também o apoio financeiro do Instituto Nacional de Ciência e Tecnologia em Computação Quântica Aplicada através do processo CNPq nº 408884/2024-0.

## Referências

- Bravyi, S. and Kitaev, A. (2005). Universal quantum computation with ideal Clifford gates and noisy ancillas. *Physical Review A*, 71(2):022316.
- Eastin, B. and Knill, E. (2009). Restrictions on Transversal Encoded Quantum Gate Sets. *Physical Review Letters*, 102(11):110502.
- Nielsen, M. A. and Chuang, I. L. (2010). *Quantum computation and quantum information*. Cambridge university press, Cambridge, 10th anniversary edition edition.
- Patterson, D. A., Hennessy, J. L., and Alexander, P. (2014). *Computer organization and design: the hardware/software interface*. Elsevier Morgan Kaufmann, Amsterdam Heidelberg, 5. ed edition.