Revista Brasileira de Ensino de Física, vol. 47, e20240469 (2025)
www.scielo.br/rbef
DOI: https://doi.org/10.1590/1806-9126-RBEF-2024-0469

Artigos Gerais

Geradores de números aleatórios: da pseudoaleatoriedade à verdadeira aleatoriedade na era da segunda revolução quântica**

Random number generators: from pseudorandomness to true randomness in the era of the second quantum revolution

Filippo Ghiglieno*[1,2,3], Luiz Roncaratti[3], Lucca Rodrigues Cunha[3], Pedro Calligaris Delbem[4], Rodrigo Silva de Almeida[5], Antonio Mauro Saraiva[6], Alexandre Delbem[5]

[1]Universidade Federal de São Carlos, Departamento de Física, São Carlos, SP, Brasil.
[2]Universidade Federal de São Carlos, Departamento de Física, Laboratório de Óptica, LAser e Fotônica, São Carlos, SP, Brasil.
[3]Universidade de Brasília, Instituto de Física, Brasília, DF, Brasil.
[4]Universidade de São Paulo, Instituto de Física, São Carlos, SP, Brasil.
[5]Universidade de São Paulo, Instituto de Ciências Matemáticas e de Computação, Departamento de Sistemas de Computação, São Carlos, SP, Brasil.
[6]Universidade de São Paulo, Escola Politécnica de São Paulo, Departamento de Engenharia de Computação e Sistemas Digitais, São Paulo, SP, Brasil.

Recebido em 24 de dezembro de 2024. Revisado em 25 de março de 2025. Aceito em 01 de abril de 2025.

Neste artigo, exploramos o conceito de aleatoriedade, desde suas origens históricas em jogos de azar até seu papel crucial em aplicações modernas, como criptografia, simulação científica e segurança da informação. Discutimos as limitações dos geradores pseudoaleatórios (PRNGs), que, por serem baseados em algoritmos determinísticos, não atendem às demandas de segurança crítica. Em contrapartida, analisamos os geradores verdadeiramente aleatórios (TRNGs), que aproveitam fenômenos físicos imprevisíveis, como ruído térmico, e destacamos os geradores quânticos de números aleatórios (QRNGs), que utilizam propriedades da mecânica quântica, como superposição e incerteza, para alcançar níveis superiores de entropia e segurança. Além disso, apresentamos uma descrição detalhada de dois tipos de qubits utilizados na geração quântica de números aleatórios: qubits supercondutores e qubits fotônicos. Discutimos os princípios físicos subjacentes a cada abordagem e suas aplicações práticas em contextos estratégicos e tecnológicos. O artigo também enfatiza a necessidade de testes rigorosos para validar a qualidade da aleatoriedade gerada, sublinhando a relevância desses dispositivos para a educação e o desenvolvimento científico e tecnológico.
Palavras-chave: Geradores verdadeiramente aleatórios (TRNGs), Geradores pseudoaleatórios (PRNGs), Geradores quânticos de números aleatórios (QRNGs), Tecnologia quântica, Entropia e aleatoriedade.

In this paper, we explore the concept of randomness, from its historical origins in games of chance to its crucial role in modern applications such as cryptography, scientific simulations, and information security. We discuss the limitations of pseudo-random number generators (PRNGs), which, being based on deterministic algorithms, do not satisfy critical security requirements. In contrast, we analyze true random number generators (TRNGs) that harness inherently unpredictable physical phenomena, such as thermal noise. Furthermore, we highlight quantum random number generators (QRNGs), which utilize quantum mechanical properties like superposition and uncertainty to achieve superior levels of entropy and security. Additionally, we provide a detailed description of two types of qubits employed in quantum random number generation: superconducting qubits and photonic qubits. We discuss the underlying physical principles of each approach and their practical applications in strategic and technological contexts. The article also emphasizes the necessity of rigorous testing to validate the quality of the generated randomness, underscoring the relevance of these devices for education and scientific and technological development.
Keywords: True Random Number Generators (TRNGs), Pseudo-Random Number Generators (PRNGs), Quantum Random Number Generators (QRNGs), Quantum Technology, Entropy and Randomness.

*Endereço de correspondência: filippo.ghiglieno@df.ufscar.br
Editor-Chefe: Marcello Ferreira

** Dedicado a todos os encontros casuais que transformaram caminhos.

1. O Que é Aleatoriedade?

O conceito de aleatoriedade é amplamente familiar. Seja ao jogar dados ou ao comprar bilhetes de loteria, todos já tiveram contato com essa ideia. A necessidade de aleatoriedade é antiga; mesmo nos épicos literários, encontram-se jogos fundamentados em resultados aleatórios [1]. De fato, loterias e jogos de azar, em suas variadas formas, fizeram parte de praticamente todas as civilizações ao longo da história.

Com o tempo, a relevância dos números aleatórios se expandiu, e novas aplicações foram surgindo, desde a previsão do tempo até a simulação de Monte Carlo [2–4] e da criptografia à amostragem estatística [5, 6]. Em nossa vida cotidiana, usamos saídas de geradores de números aleatórios sem nem perceber. Por exemplo, senhas de uso único (OTPs) enviadas para diversos fins [7], os PINs fornecidos pelos bancos e os CAPTCHAs [8] que aparecem na tela ao acessar sites são todos esperados como saídas de geradores de números aleatórios. Então, como são gerados números aleatórios?

Em uma primeira experiência com aleatoriedade, é comum notar que o botão RAND em uma calculadora gera um número diferente a cada vez que é pressionado. Isso frequentemente causa surpresa e questionamentos, principalmente pela falta de conhecimento sobre eletrônica e suas limitações. Calculadoras comuns não geram números verdadeiramente aleatórios, mas utilizam listas grandes de números pré-programados que são percorridos sequencialmente. Essas listas exibem uma *boa aleatoriedade* e simulam uma distribuição uniforme, onde os números estão distribuídos em proporções iguais, o que significa que, ao observar os números gerados, há uma quantidade equilibrada de cada dígito (como tantos 1s quanto 9s, tantos 1s quanto 2s, e assim sucessivamente, respeitando, portanto, o princípio da uniformidade). Essas listas, no entanto, não atendem a requisitos de segurança.

A Figura 1 exibe números aleatórios do alfabeto binário (0,1), gerados por um algoritmo da biblioteca *Random* do Python (versão 3.9.7), executado em um computador DELL (processador 11$^{\text{th}}$ Gen Intel(R) Core(TM) i7-1165G7 @ 2.80GHz 1.69 GHz). O resultado é uma sequência pseudoaleatória.

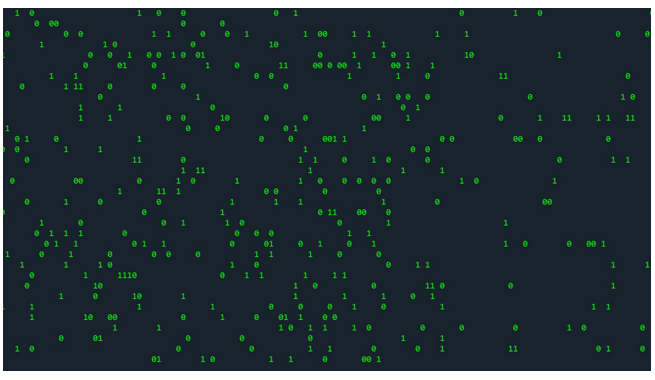

Figura 1: Sequência pseudoaleatória de números binários gerada em um computador DELL utilizando um código em Python, desenvolvido por meio de um prompt no ChatGPT.

1.1. Aleatoriedade na segurança, nas simulações e nos jogos

No contexto moderno, a aleatoriedade é crucial em campos como a **criptografia**, onde a segurança das comunicações digitais depende de chaves criptográficas robustas, geradas a partir de números aleatórios. Essas chaves devem ser verdadeiramente imprevisíveis para evitar ataques e garantir a confidencialidade dos dados. Em **simulações científicas**, métodos numéricos baseados em números aleatórios permitem modelar fenômenos complexos, avançando a compreensão de sistemas dinâmicos. Já nos **jogos de azar** e no **metaverso**, a integridade dos jogos online e físicos depende de geradores de números aleatórios (Random Number Generators – **RNGs**) que garantam integridade e imprevisibilidade.

A saída ideal de um gerador de números aleatórios deve obedecer aos princípios de **uniformidade** e **independência**, ou seja, os números gerados devem ser equiprováveis e sem correlações entre si. Além disso, outro fator crucial, especialmente em aplicações de criptografia, é a **privacidade**. A partir dos pilares anteriores (uniformidade, independência e privacidade), o teorema de Shannon demonstra que sistemas como o *one-time pad*, que utiliza uma chave única e aleatória com comprimento igual ao da mensagem para criptografá-la, podem alcançar segurança teoricamente perfeita, ou seja, a impossibilidade de um adversário decifrar a mensagem sem conhecer a chave, independentemente dos recursos computacionais disponíveis [9]. Para isso, é necessário que a chave seja totalmente aleatória, tenha o mesmo comprimento da mensagem e seja usada apenas uma vez. Nessas condições, o texto cifrado não revela nenhuma informação sobre o conteúdo original sem o conhecimento da chave, garantindo um nível de segurança absoluto [10]. A impossibilidade prática de utilizar chaves com essas características nos modernos sistemas de telecomunicações tem direcionado os esforços de pesquisadores teóricos e experimentais de diversas áreas para o desenvolvimento de geradores de números aleatórios (RNGs) que combinem uniformidade, independência e privacidade, além de serem compactos, integrados e capazes de operar com alta taxa de transferência (*throughput*).

A Figura 2 reagrupa as categorias de RNGs baseados em soluções de hardware e software, abrangendo diversas tipologias.Curiosamente, muitos dos geradores utilizados em nossas atividades cotidianas não produzem aleatoriedade verdadeira, sendo mais apropriadamente classificados como *geradores de números pseudoaleatórios* (Pseudo Random Number Generators – **PRNGs**). PRNGs são baseados em algoritmos determinísticos que convertem uma pequena sequência inicial em uma

 DOI: https://doi.org/10.1590/1806-9126-RBEF-2024-0469

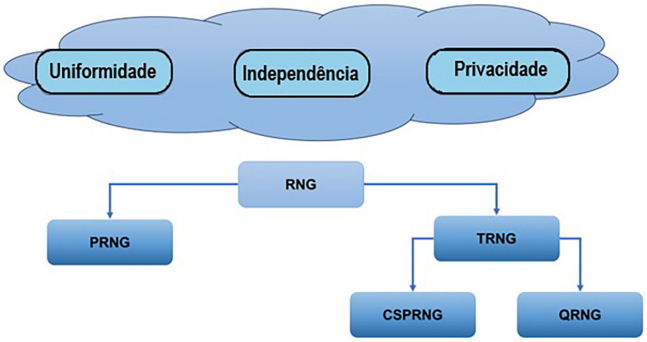

Figura 2: Diferentes famílias de RNGs, classificadas de acordo com soluções baseadas em hardware e software de diversas tipologias.

sequência maior de números [11–13] que passam em testes específicos, como o teste NIST [14] e o teste Die-Hard [15], para serem úteis em situações específicas. Esses geradores são rápidos, mas, como dependem de um algoritmo e uma sequência inicial previsível, os números gerados podem ser antecipados, o que os torna inadequados para aplicações onde a privacidade é essencial, mas adequados para simulações e previsões meteorológicas.

Por outro lado, *geradores de números verdadeiramente aleatórios* (True Random Number Generators – **TRNGs**) baseiam-se em processos físicos de natureza imprevisível. Para gerar números genuinamente aleatórios, esses dispositivos exploram a aleatoriedade de fenômenos naturais, como ruído atmosférico [16], radiação cósmica [17], ruído térmico [18], ruídos em circuitos eletrônicos [19] e sistemas caóticos [20]. No entanto, extrair aleatoriedade de ruídos pode ser um processo lento e complexo, e o desenvolvimento de geradores de números aleatórios eficientes continua sendo uma área de grande interesse científico e tecnológico. Diversos tipos de geradores foram recentemente propostos e construídos, com aplicações que vão desde sistemas de segurança até amostragens estatísticas e simuladores avançados.

Na sede da Cloudflare [21], em São Francisco, há uma parede conhecida como LavaRand, composta por lâmpadas de lava que produzem formas de cera imprevisíveis. Uma câmera capturando essas formas permite a conversão das imagens em bytes aleatórios. A aleatoriedade usada em diversas aplicações é geralmente gerada pelo sistema operacional, que coleta dados de diferentes fontes dentro do próprio dispositivo, conhecidas como **fontes de entropia**. Essas fontes incluem eventos imprevisíveis ou variáveis que ocorrem naturalmente durante o uso do hardware e do software, contribuindo com dados para os geradores de números aleatórios. Entre as fontes de entropia mais comuns estão as interrupções de hardware, como o tempo dos movimentos do mouse ou o ritmo de pressionamento das teclas, que são difíceis de prever com precisão, e também as interrupções de software, que ocorrem em momentos aleatórios devido à execução simultânea de diversos processos e eventos no sistema. Outro exemplo é o tempo de busca do disco rígido, que varia ligeiramente conforme as condições operacionais do sistema.

Esses dados, provenientes de fontes de entropia, são combinados e processados pelo sistema operacional para fornecer uma aleatoriedade prática e efetiva. Esse tipo de aleatoriedade, gerada a partir das atividades internas do dispositivo, é amplamente útil em aplicações cotidianas e para fins de segurança moderada.

É importante notar que, embora esses fenômenos produzam números não determinísticos, a qualidade desses números é de difícil mensuração, pois uma teoria futura poderia modelar esses processos com precisão, tornando-os previsíveis. Por isso, é ideal buscar fontes com aleatoriedade intrínseca e garantida.

A mecânica quântica é inerentemente não determinística [22, 23], o que torna os sistemas quânticos ideais para a geração de números realmente aleatórios. As leis da mecânica quântica possibilitam a quantificação precisa da qualidade dessa aleatoriedade. Com os avanços na tecnologia quântica, hoje é possível gerar números aleatórios de alta qualidade, ideais para aplicações em criptografia com segurança incondicional [24].

Os geradores de números aleatórios quânticos (Quantum Random Number Generators – **QRNGs**) utilizam fenômenos quânticos para produzir aleatoriedade comprovável [25, 26]. Entre as propriedades notáveis dos sistemas quânticos estão a superposição de estados, o colapso de estado durante a medição, o emaranhamento, que mantém correlações não-locais entre partículas, e a indistinguibilidade de partículas – características que desafiam a intuição clássica, mas que garantem a verdadeira aleatoriedade desses sistemas.

Os primeiros QRNGs se baseavam no princípio da incerteza da mecânica quântica, explorando fenômenos como a radioatividade e o ruído de disparo em circuitos eletrônicos. Esses métodos, entretanto, apresentavam limitações como o manuseio complexo de fontes radioativas e a necessidade de diferenciar entre ruído de disparo e ruído térmico, além de serem relativamente lentos. Para superar esses desafios, a tecnologia avançou na direção da óptica quântica, criando geradores baseados em detectores de fótons únicos e detecção fotográfica macroscópica [27]. Sistemas atômicos, como o ruído de spin, também foram investigados, comprovando a capacidade dos QRNGs. Com controle aprimorado sobre esses sistemas, há propostas de QRNGs que aprimoram os testes de aleatoriedade pública [14, 15], elevando a confiabilidade e a qualidade da aleatoriedade.

1.2. Entropia: quantificando a imprevisibilidade

A **entropia** é uma característica fundamental dos RNGs e pode ser definida como uma medição de incerteza (ou desordem, em termos de sua definição clássica) em um sistema.

A entropia de uma distribuição de probabilidades, como a dos números gerados por um RNG, pode ser

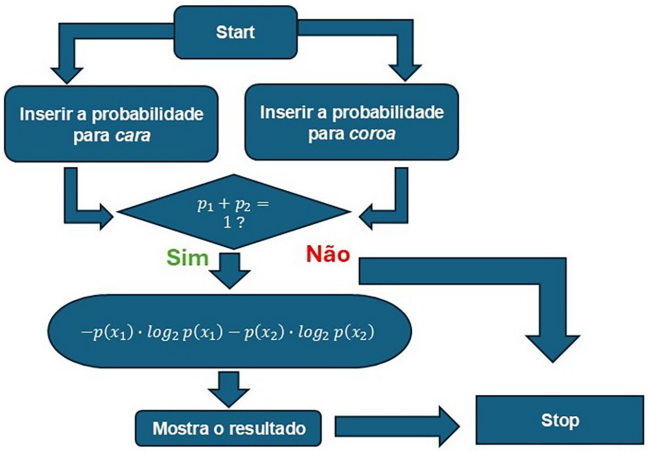

Figura 3: Fluxograma para calcular a entropia de uma moeda. Fonte: Filippo Ghiglieno.

calculada pela fórmula:

$$\begin{aligned} H(X) = & -\sum_{x\epsilon X} p(x) \cdot \log_2\ p(x) \\ = & -p(x_1) \cdot \log_2 p(x_1) - p(x_2) \cdot \log_2 p(x_2) - \cdots \\ & - p(x_n) \cdot log_2 p(x_n) \end{aligned} \quad (1)$$

onde $p(x_1)$, $p(x_2)$, etc., são as probabilidades de cada possível resultado. No caso de um lançamento de moeda, consideram-se apenas p_{cara} e p_{coroa}.

A soma de todas as probabilidades em um espaço de probabilidade é sempre igual a 1, indicando que, embora cada lado tenha 50% (0.5) de chance, há 100% de chance de que um dos lados seja o resultado. Cada valor de probabilidade é inferior a 1, fazendo com que seu logaritmo seja negativo. Para calcular um valor positivo de entropia total para o sistema, utiliza-se a soma dos logaritmos negativos multiplicados pelas probabilidades (1).

Em um lançamento de uma moeda justa, onde ambas as faces têm chances iguais de ocorrer, a entropia é igual a 1 bit. Considerando *cara* como 1 e *coroa* como 0, a incerteza permanece equilibrada sobre qual valor esse único bit terá. Em uma sequência de 10 lançamentos de moeda justa, a entropia seria de 10 bits.

Caso uma moeda não seja justa, a entropia resultante será inferior a 1 bit. Quanto maior o viés, menor será a entropia. Um exemplo extremo ocorre em uma moeda com o mesmo símbolo dos dois lados, cuja entropia seria de 0 bits. Uma moeda onde a *cara* tem 75% de probabilidade e a coroa 25%, a entropia do lançamento seria de aproximadamente 0.8 bits.

O cálculo da entropia de uma moeda tendenciosa pode ser feito com um programa. A Figura 3 apresenta o fluxograma dos passos necessários para realizar esse cálculo. Esses passos podem ser implementados em Python da seguinte maneira:

```
"""
Created on Wed Nov 20 12:09:01 2024
@author: Quantum ready
"""

import math

def calcular_entropia():
    try:
      caras = float(input("Digite a probabilidade
       de caras (entre 0,0 e 1,0): "))
      coroas = float(input("Digite a probabilidade
       de coroas (entre 0,0 e 1,0): "))
      if not (0 <= caras <= 1) or not
       (0 <= coroas <= 1):
        raise ValueError("As probabilidades devem
         estar entre 0,0 e 1,0.")
      if abs(1 - (caras + coroas)) > 0.01:
        raise ValueError("A soma de P(caras) e
         P(coroas) deve ser aproximadamente 1.")
    except ValueError as e:

      print(e)
      return

    entropia = sum(-p * math.log2(p) for p in
     [caras, coroas] if p > 0)
    print(f"P(caras)={caras:.2f}, P(coroas)=
     {coroas:.2f}, Entropia: {entropia:.2f} bits")

if __name__ == "__main__":
    calcular_entropia()
```

O código executado acima acompanha os seguintes passos lógicos:

1. recebem-se dois números decimais como entrada, representando as probabilidades de ocorrer *cara* e *coroa*, respectivamente;
2. as entradas são validadas antes de efetuar os cálculos: a soma das duas probabilidades deve ser igual a 1; no entanto, devido a possíveis imprecisões de ponto flutuante, em vez de comparar a soma diretamente com 1, é recomendável medir a proximidade;
3. a fórmula da equação da entropia é então aplicada aos valores fornecidos, e o resultado é exibido.

Aqui estão alguns resultados obtidos para diferentes probabilidades:

- P(*cara*) = 0.5, P(*coroa*) = 0.5, entropia: 1 bit;
- P(*cara*) = 0.75, P(*coroa*) = 0.25, entropia: 0.81 bit;
- P(*cara*) = 0.8, P(*coroa*) = 0.2, entropia: 0.72 bit;
- P(*cara*) = 0.1, P(*coroa*) = 0.9, entropia: 0.47 bit.

Como pode ser observado, embora ainda seja obtido um bit de saída (ou seja, se o resultado foi *cara* ou *coroa*) a cada lançamento da moeda, a entropia da saída diminui à medida que o lançamento se torna mais enviesado. Outra forma de entender isso é notar que, quando um lançamento de moeda possui entropia de 1 bit, prever o resultado é tão difícil quanto possível para um lançamento de moeda. Com uma entropia de 0.47 bits, um dos resultados torna-se mais provável que o outro, facilitando a previsão.

A Figura 4 mostra como a entropia (a linha curva sólida) muda conforme o lançamento da moeda se torna mais enviesado. As linhas pontilhadas representam as probabilidades de sair cara ou coroa. Observe que a soma dessas probabilidades permanece sempre igual a

 DOI: https://doi.org/10.1590/1806-9126-RBEF-2024-0469

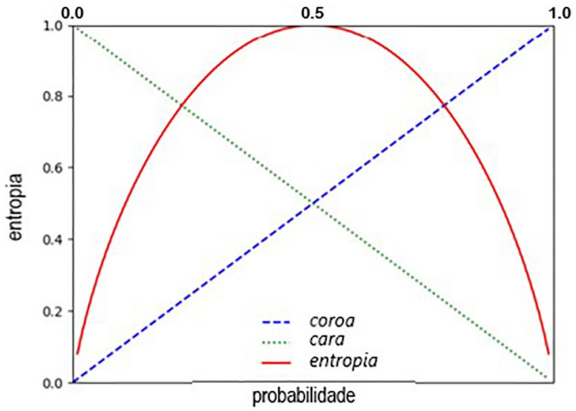

Figura 4: Entropia e probabilidade para a moeda.

1, pois representam o espaço total de probabilidades, ou seja, não há um terceiro resultado possível. A entropia é máxima (o pico no meio) quando as probabilidades de cara e coroa são ambas de 50%. Esse é o ponto em que se torna mais difícil prever qual lado da moeda sairá.

Como a entropia se relaciona com os RNGs? A entropia em RNGs mede o grau de imprevisibilidade dos números gerados. Quanto maior a entropia, mais difícil é prever os próximos valores, garantindo segurança e confiabilidade. Nos QRNGs, a entropia é alta devido à natureza intrinsecamente aleatória dos fenômenos quânticos. Por outro lado, nos PRNGs, a entropia depende da qualidade da semente inicial e da função geradora, como será discutido no próximo parágrafo.

Em contextos criptográficos, uma alta entropia é fundamental para assegurar a imprevisibilidade e a segurança dos dados gerados.

2. Geradores de Números Verdadeiramente Aleatórios (TRNG)

Fenômenos como o lançamento de moedas, rolagem de dados, decaimento nuclear, ruído térmico em resistores e até mudanças no clima são exemplos de processos que geram valores imprevisíveis, podendo servir como fontes de aleatoriedade com diferentes níveis de qualidade (entropia) e velocidade (taxa de produção de novos números). A *velocidade de geração* é uma característica fundamental, pois indica o quão rapidamente um RNG pode produzir novos números. Por exemplo, a decisão de levar ou não um guarda-chuva poderia ser baseada na aleatoriedade do fenômeno físico da chuva, mas essa decisão não precisaria ser atualizada a cada milissegundo. A taxa de geração de aleatoriedade está limitada tanto pelo fenômeno que se está amostrando (a ocorrência de chuva, por exemplo) quanto pela frequência com que as condições físicas mudam (como o tempo necessário para a chuva começar ou parar, que é, no mínimo, de alguns minutos).

Em geral, é desejável que os TRNGs atendam às seguintes propriedades:

- **proteção contra ataques físicos:** TRNGs devem ser projetados para proteger contra ataques físicos, como tentativas de manipulação por invasores com acesso ao dispositivo, que possam tentar prever ou influenciar a saída dos números gerados.
- **modelo físico para previsibilidade de desempenho:** TRNGs devem contar com um modelo físico que permita prever a taxa de geração e a entropia dos bits gerados com base nas propriedades fundamentais dos fenômenos físicos subjacentes. Esses *testes de integridade* (teste de uniformidade, teste de entropia, teste de sequência repetitiva, teste de viés, teste de correlação) são ideais para detectar falhas na operação do TRNG, podendo interromper automaticamente o funcionamento caso algum problema seja identificado. Vale ressaltar que, embora o modelo ajude a quantificar características operacionais do RNG (como a taxa de geração e a entropia dos bits), ele não prevê os bits específicos gerados. Em essência, ele responde a perguntas como:
  - *os bits gerados são suficientemente aleatórios?*
  - *os bits estão sendo gerados na velocidade necessária*?

mas sem indicar o valor exato de cada bit produzido pelo TRNG.

TRNGs amostram o mundo físico para gerar valores que, na prática, são imprevisíveis. Embora se possa discutir filosoficamente que o universo pode ser determinístico e que nada é realmente ***imprevisível***, isso não é relevante para aplicações criptográficas, pois o que importa é que os valores sejam impossíveis de prever por adversários contemporâneos.

A lista seguinte apresenta alguns dos fenômenos físicos mais comuns empregados por TRNGs, cada um com suas características de custo, confiabilidade e segurança, mostrando como esses dispositivos aproveitam a imprevisibilidade natural para produzir aleatoriedade em níveis criptograficamente seguros:

- **Decaimento radioativo:** geradores de números aleatórios que utilizam o decaimento radioativo produzem números verdadeiramente aleatórios, pois o processo é intrinsecamente imprevisível no nível atômico. Em outras palavras, não há como prever quando exatamente um átomo específico irá decair. No entanto, quando se observa uma grande quantidade de átomos idênticos, é possível calcular a taxa média de decaimento com base no conceito de meia-vida, que é o tempo necessário para que metade dos átomos no grupo decaia ($N_0 e^{-\lambda t}$) [28]. Para gerar números aleatórios, esse processo probabilístico pode ser monitorado por um contador Geiger, que detecta os eventos de decaimento e converte esses eventos em bits digitais [29]. Cada detecção é um evento aleatório,

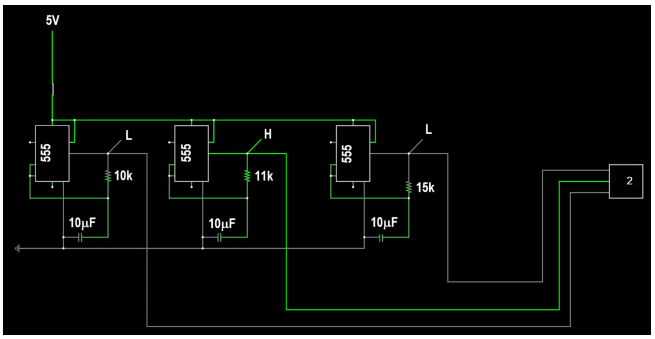

Figura 5: Gerador de números aleatórios com 3 circuitos osciladores do tipo [30].

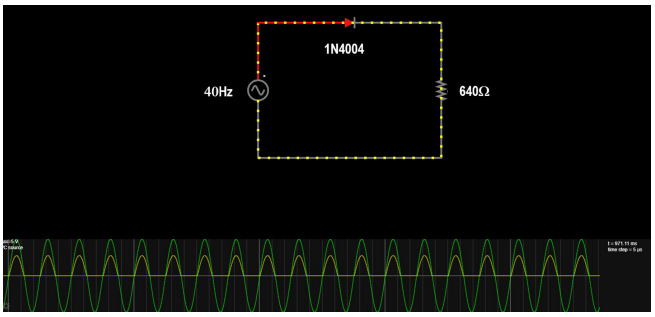

Figura 6: Circuito elétrico com diodo, em baixo em verde a tensão de entrada no circuito e em amarelo a corrente medida depois do componente 1N4004 (Half-Wave Rectifier).

criando uma sequência de bits que reflete a natureza imprevisível do decaimento. Apesar de sua eficácia em produzir aleatoriedade, o uso dessa técnica enfrenta limitações. Ela é cara devido à complexidade de amostrar esses eventos de forma confiável, e porque exige o uso de fontes radioativas específicas para garantir a taxa de geração desejada de números aleatórios.

- **Ruído atmosférico captado por receptores de rádio:** esse método é econômico e fácil de implementar, mas é vulnerável a ataques externos, pois um adversário pode influenciar a saída do RNG por meio de interferências eletromagnéticas.
- **Desvios na temporização de sinais de relógio:** essa técnica é de baixo custo. Os sinais de relógio já são componentes essenciais em praticamente todos os processadores modernos, o que elimina a necessidade de adicionar novo hardware. A Figura 5 mostra um circuito para gerar um número aleatório com uso do circuito integrado (IC) **555** [30], utilizado em várias aplicações, como temporizadores, geradores de onda e osciladores. Foi lançado em 1972 pela fabricante americana Signetics (mais tarde adquirida pela Philips) com os nomes comerciais de **NE555T** (invólucro metálico) e **NE555V** (invólucro DIP), e foi apelidado de *The IC Time Machine* (*A Máquina do Tempo num Chip*). Este componente continua em pleno uso graças a sua simplicidade, versatilidade, baixo preço e boa estabilidade às variações de temperatura e de tensão. É composto por 24 transistores de junção bipolar, 2 diodos e 16 resistores em um encapsulamento duplo em linha (DIP) de 8 pinos. No entanto, implementar corretamente essa técnica exige muito cuidado, pois medir as variações (ou desvios, em inglês *jitter*) no tempo dos sinais de relógio é uma tarefa complexa, já que esses sinais não foram originalmente projetados para gerar números aleatórios. Além disso, seu comportamento é facilmente influenciado por invasores, seja por meio de acesso físico (como a introdução de ruídos na fonte de alimentação) ou de acesso remoto ao processador (executando outras aplicações no mesmo dispositivo).
- **Ruído elétrico gerado pelo efeito avalanche ou Zener:** esse tipo de ruído ocorre em diodos, que são componentes comuns em circuitos elétricos. Os diodos normalmente funcionam como *válvulas*, permitindo que a corrente flua em apenas uma direção, o que ajuda a proteger outros componentes no circuito (veja Figura 6).
  Alguns diodos específicos, chamados diodos de avalanche ou Zener, possuem uma propriedade interessante: quando submetidos a uma certa tensão, eles criam um ruído elétrico altamente aleatório devido a processos físicos internos. No caso do efeito avalanche, quando a tensão ultrapassa um limite, os elétrons se movem de forma caótica dentro do material, gerando picos de corrente imprevisíveis. De maneira similar, o efeito Zener em certos tipos de diodos, sob tensão reversa, também resulta em um ruído elétrico imprevisível. No entanto, esses fenômenos, que são explorados como fontes de aleatoriedade, são geralmente considerados uma desvantagem nos usos tradicionais dos diodos. Os picos de corrente e o ruído irregular comprometem a estabilidade elétrica, sendo frequentemente indesejados em dispositivos eletrônicos convencionais. Por esse motivo, os fabricantes tendem a projetar diodos que minimizem ao máximo esse ruído, em vez de amplificá-lo, pois sua presença é incompatível com a maioria das aplicações que exigem precisão e eficiência nos circuitos eletrônicos. Nos próximos parágrafos, o assunto será abordado com mais detalhes.
- **Osciladores de anel:** essa técnica é semelhante ao método de deriva de clock, pois também depende do *jitter* (pequenas variações) presente nos sinais de temporização, mas ao invés de medir diretamente essas pequenas variações de tempo, a técnica usa um conjunto de portas lógicas NOT, componentes simples que invertem o sinal de entrada. Essas portas NOT são organizadas em um circuito circular, formando um “anel” com um

 DOI: https://doi.org/10.1590/1806-9126-RBEF-2024-0469

número ímpar de portas. Quando um sinal percorre esse anel, ele passa por cada porta NOT e muda de estado a cada vez (de 0 para 1, de 1 para 0, e assim por diante). Como há um número ímpar de portas, o sinal nunca se estabiliza; ele continua alternando, ou oscilando, entre dois níveis de voltagem (alto e baixo), criando um sinal de saída que é essencialmente instável e aleatório. Esse sinal oscilante serve como uma fonte de aleatoriedade, pois o jitter no circuito faz com que o tempo exato das oscilações varie de forma imprevisível. Essas oscilações aleatórias são então convertidas em bits digitais, proporcionando uma fonte de números aleatórios com base nas características naturais e imprevisíveis do jitter presente no circuito [31] (nos próximos parágrafos o assunto será abordado com maior detalhe).

- **Multiplicação de Entropia Modular (MEM):** MEM é uma técnica inovadora para gerar números aleatórios a partir de ruído analógico, desenvolvida inicialmente por Peter Allan no final dos anos 1990 e depois, de forma independente, por Bill Cox na década de 2010. O método começa com uma fonte de ruído analógico, como o ruído térmico ou elétrico. Esse ruído, uma vez amplificado, serve como fonte de entropia, ou seja, de imprevisibilidade. A grande inovação da MEM está em um processo de multiplicação de entropia que aplica um conjunto de regras muito simples para amplificar as variações do ruído. Isso cria flutuações de voltagem intensas e aleatórias, que são então convertidas em dados digitais ou bits para a geração de números aleatórios. Esse sistema de regras e amplificação é desenhado para ser de baixo custo e resistente a interferências eletromagnéticas, uma vantagem importante para proteger a integridade dos dados gerados. Além disso, a MEM inclui um modelo físico que permite monitorar a "saúde" do RNG, ou seja, a qualidade da aleatoriedade gerada. Esse modelo avalia se o dispositivo está operando corretamente, tornando a técnica não apenas econômica, mas também segura e confiável para aplicações onde a aleatoriedade verdadeira é essencial (nos próximos parágrafos o assunto será abordado com maior detalhe).

2.1. Diodos e geração de números aleatórios

Diodos são componentes eletrônicos projetados para restringir o fluxo de corrente em apenas uma direção, o que os torna úteis para proteger circuitos, especialmente em casos em que a polaridade da fonte de alimentação é invertida. Em situações normais, quando a corrente flui no sentido "natural" do diodo, o diodo está *polarizado diretamente*. Já quando a voltagem é aplicada no sentido oposto, o diodo (idealmente) interrompe o fluxo de corrente e é chamado de *polarizado reversamente*. O bloqueio da corrente em polarização reversa é o que torna

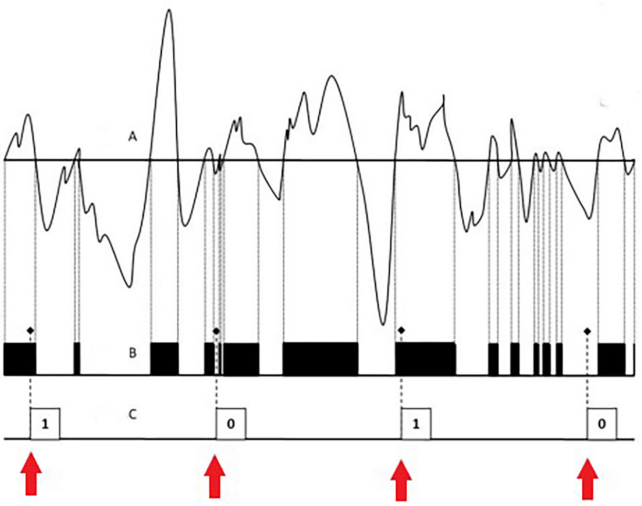

Figura 7: Esquema do processo de geração de números aleatórios utilizando o ruído de um diodo Zener. (**A**) A linha contínua representa o valor médio da tensão em uma janela de tempo definida. (**B**) O sinal amplificado é enviado a um circuito comparador que verifica se a tensão está acima ou abaixo do valor médio. (**C**) Em intervalos regulares, o circuito codifica o resultado em bits: *0* para tensões abaixo do valor médio e *1* para tensões acima.

os diodos tão úteis. No entanto, alguns tipos de diodos apresentam efeitos secundários indesejáveis conhecidos como *efeitos parasíticos*. Embora geralmente esses efeitos não sejam desejáveis, eles podem ser aproveitados para a geração de números aleatórios.

Esse é o caso do ***efeito avalanche*** e do ***efeito Zener***, dois fenômenos físicos distintos que geram ruído no circuito elétrico. Esse ruído pode ser amplificado e convertido em sinais digitais por um conversor analógico-digital (ADC), criando uma fonte de aleatoriedade (veja Figura 7).

Apesar de amplamente utilizados, diodos Zener não são ideais para TRNGs por várias razões:

- **design voltado para baixa geração de ruído:** os diodos Zener são projetados para minimizar o ruído de avalanche, o que os torna uma fonte pobre de ruído eletrônico, na verdade, um uso comum dos diodos Zener é a regulação de tensão, onde o ruído é extremamente indesejável;
- **efeito Zener não parametrizado pelo fabricante:** o efeito Zener em polarização reversa não costuma ser medido nem parametrizado pelos fabricantes, eles priorizam o controle de qualidade para o funcionamento padrão dos diodos Zener, sem foco em efeitos secundários indesejados;
- **variações significativas de ruído entre dispositivos:** o nível de ruído do efeito Zener varia substancialmente de um diodo para outro, podendo ser até 10 vezes maior ou menor dependendo do fabricante;

- **sensibilidade à temperatura e ao envelhecimento:** o ruído gerado pelo efeito Zener é bastante sensível à temperatura e pode mudar ao longo do tempo, conforme o circuito envelhece;
- **ausência de modelo físico confiável:** não existe um modelo físico claro que correlacione bem o ruído Zener para avaliar a *saúde* de um TRNG, dificultando a garantia de uma fonte de aleatoriedade consistente e de alta qualidade.

Em resumo, embora o efeito Zener possa gerar ruído útil para TRNGs, ele não oferece a confiabilidade e previsibilidade desejadas para aplicações que exigem uma fonte consistente de aleatoriedade.

2.2. Osciladores de anel com portas NOT

Uma porta NOT, ou inversora, é um componente básico que inverte o sinal de entrada: se a entrada estiver em nível alto (1), a saída será em nível baixo (0), e vice-versa. Em um oscilador de anel, um número ímpar de portas NOT é conectado em sequência, formando um circuito fechado onde a saída final é realimentada à entrada inicial. Esse circuito gera uma oscilação contínua entre altos e baixos níveis de voltagem, produzindo um sinal que alterna de forma regular. A aleatoriedade no circuito é originada pelo *jitter*, ou atraso temporal, produzido por cada anel oscilador. Esse atraso consiste em variações imprevisíveis no sinal periódico em relação a um clock de referência.

Embora o *jitter* seja geralmente considerado uma característica indesejada em sistemas eletrônicos, no contexto de um TRNG ele se torna uma propriedade fundamental para a geração de sinais aleatórios. O *jitter* apresenta uma distribuição Gaussiana ao redor de cada transição de clock, entre os estados lógicos baixo e alto, proporcionando a entropia necessária para a criação de números verdadeiramente aleatórios [32]. Na Figura 8 é proposto um TRNG com 114 anéis osciladores, com cada anel possuindo 13 inversores (NOT). A frequência de operação do circuito é de 40 MHz, e o flip-flop atua como um amostrador síncrono, capturando o estado lógico do sinal proveniente dos osciladores de anel. Nesta implementação um estágio de pós processamento

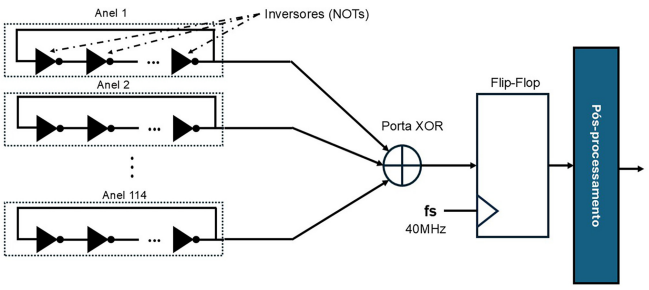

Figura 8: TRNGs baseado em anéis osciladores (B. Sunar, W.J. Martin e D.R. Stinson, IEEE Transaction Computers **56**, 109, 2007 [34]).

é requerido, pois todas as saídas dos anéis osciladores são conectadas em uma única porta XOR, consequentemente, todos os sinais são assíncronos, forçando a porta XOR a trabalhar em estado sobrecarregado. Em [33], propõe-se um aprimoramento da estrutura apresentada em [34], alcançando uma frequência de operação de 100 MHz. Nesta abordagem, são colocados flip-flops para a amostragem após cada saída de anel oscilador, tornando-a síncrona, não sendo necessário um estágio de pós-processamento.

Entretanto, osciladores de anel apresentam várias limitações como fonte de aleatoriedade:

- **dependência de efeitos parasíticos:** como os diodos Zener, que também são usados para gerar aleatoriedade, os osciladores de anel baseiam-se em efeitos que surgem como consequência do *design* e não como objetivo principal do processo de fabricação, o que torna difícil garantir que essas características aleatórias sejam consistentes e confiáveis;
- **pouca defesa física:** osciladores de anel são vulneráveis a ataques externos: um invasor com um gerador de onda senoidal pode aplicar um sinal de frequência semelhante à do oscilador na fonte de alimentação do *chip*, fazendo com que o oscilador *sincronize* com essa frequência externa permitindo ao invasor prever a saída do TRNG, comprometendo a aleatoriedade (ataque de injeção de falhas).

Para melhorar a segurança e confiabilidade dos TRNGs baseados em osciladores de anel, seguem algumas práticas recomendadas:

1. **uso de um contador binário:** adicionar um contador binário simples na saída do TRNG (pós-processamento) permite monitorar a frequência das oscilações. Se o número de 1 for muito maior do que o de 0 em uma janela de tempo (como o último minuto), essa diferença pode indicar uma falha de funcionamento;
2. **transparência e monitoramento de saúde:** divulgar o projeto do TRNG e permitir o acesso público à contagem completa dos bits gerados facilita uma avaliação independente da saúde do dispositivo, esse tipo de monitoramento é útil para detectar falhas no funcionamento do dispositivo ao longo do tempo.
3. **verificação externa de entropia:** para TRNGs com amostragem de atraso fixo (a solução mais comum), recomenda-se o uso de um verificador externo para estimar a entropia (ou seja, o nível de imprevisibilidade) de cada amostra, isso ajuda a garantir que o TRNG esteja gerando bits suficientemente aleatórios (a recomendação 2 foca na *integridade do funcionamento* do TRNG, e a recomendação 3 foca na *qualidade da aleatoriedade* dos números gerados);

 DOI: https://doi.org/10.1590/1806-9126-RBEF-2024-0469

4. **proteção física:** osciladores de anel são vulneráveis a ataques de injeção de ruído, se a segurança do sistema permitir essas interferências, o uso pode ser aceitável, caso contrário, recomenda-se adicionar uma barreira física, como um encapsulamento (*potting*) ou uma blindagem sobre o IC, para dificultar o acesso direto de invasores ao oscilador;
5. **integração com CSPRNG e Flash Seguro:** caso o *chip* disponha de uma memória *flash* segura (não facilmente acessível a invasores), uma solução robusta é combinar a saída do TRNG com um gerador de números aleatórios criptograficamente seguro (CSPRNG) e uma semente armazenada nessa memória. A semente serve como ponto de partida para alimentar o CSPRNG, que, por sua vez, utiliza tanto a semente quanto a saída do TRNG para gerar números aleatórios mais imprevisíveis e consistentes. Com o tempo, a qualidade da saída do TRNG pode ser afetada por fatores como desgaste físico ou variações de temperatura. O CSPRNG compensa essas variações, reforçando a aleatoriedade com a combinação da semente e da saída do TRNG. Esse método oferece uma camada adicional de segurança, garantindo uma fonte de números aleatórios confiáveis, mesmo em condições variáveis, e reduzindo o impacto de potenciais degradações na saída do TRNG.

Essas recomendações aplicam-se a diferentes tipos de TRNGs, mas são especialmente relevantes para osciladores de anel, dada sua vulnerabilidade a interferências externas e a ausência de um modelo físico robusto que garanta a quantidade de entropia gerada.

2.3. TRNGs baseados na Multiplicação Modular de Entropia (MEM)

O circuito baseado na Multiplicação de Entropia Modular (MEM) é composto por vários elementos. Primeiro, uma fonte de ruído térmico (R), como um resistor que gera ruído térmico. Esse ruído é então amplificado por um amplificador de baixo ruído (LNA). O sinal amplificado é comparado a uma tensão de referência (V_{ref}) por um comparador. Dependendo da saída do comparador, uma lógica de controle decide se dobra a tensão ou o excesso sobre o ponto médio, de forma que a voltagem flutua de maneira imprevisível, mas permanece dentro seu intervalo.

A saída da lógica de controle é realimentada no sistema através de um conversor digital-para-analógico (DAC). Para garantir a verdadeira aleatoriedade dos *bits* de saída, uma função de branqueamento, como uma função *hash* criptográfica, é aplicada. Finalmente, um monitor de saúde verifica o nível de entropia para assegurar que permanece dentro dos limites esperados, garantindo a aleatoriedade da saída. Este circuito é essencial para gerar números aleatórios verdadeiros, necessários em diversas aplicações de segurança e criptografia. O

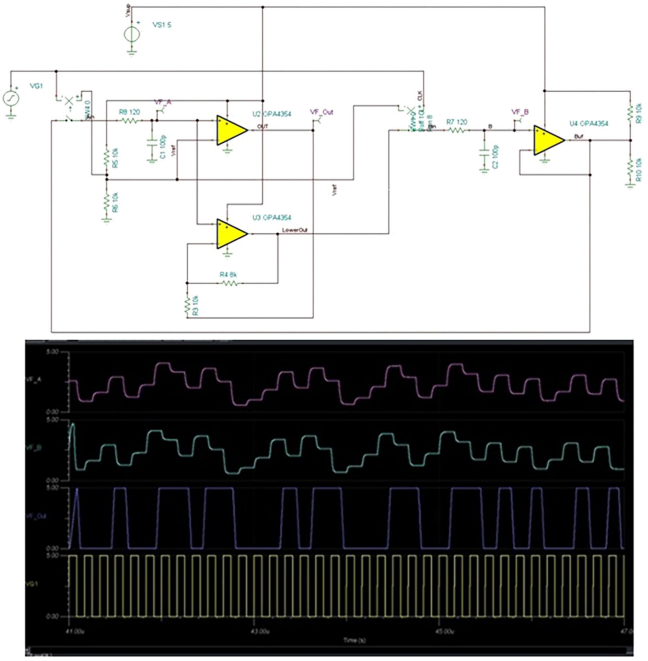

Figura 9: Acima – esquema elétrico do *Infinity Noise* MEM (GITHUB, *waywardgeek/infnoise*, disponível em: https://github.com/waywardgeek/infnoise, acessado em: 04/11/2024 [35]), em baixo – simulação dos sinais de saída.

método MEM oferece diversas vantagens específicas para TRNGs:

1. **resistência a interferências:** com base no comparador a técnica é resistente a ataques por injeção de ruído eletromagnético e a interferências capacitivas ou indutivas, tornando o sistema mais seguro;
2. **modelo físico para monitoramento:** a MEM fornece um modelo físico que permite avaliar continuamente a integridade do RNG, garantindo que o dispositivo esteja funcionando corretamente;
3. **baixo custo e simplicidade:** os componentes necessários para a arquitetura MEM são baratos e em pequena quantidade, e o *design* não possui restrições de patentes;
4. **disponibilidade de projetos abertos:** existem alguns esquemas gratuitos para implementar o MEM, como o projeto *infnoise* de Bill Cox [35] (veja Figura 9) e o redesenho *REDOUBLER* de Peter Allan [36];
5. **alta velocidade:** o método MEM é rápido, com o i*nfnoise* operando a velocidades superiores a 100 Mbit/segundo. No entanto, é importante lembrar que a velocidade não deve ser o principal critério para TRNGs, pois sua saída geralmente é usada apenas para fornecer a semente para geradores de números pseudoaleatórios criptograficamente seguros (Cryptographically-Secure Pseudo-Random Number Generators – CSPRNGs), assunto do próximo paragrafo. Em geral, 512 bits aleatórios de um TRNG são suficientes para semear um

CSPRNG, desde que ele ofereça garantias de segurança apropriadas.

2.4. Diretrizes para o projeto de TRNGs

A segurança criptográfica depende diretamente da qualidade dos números aleatórios, e isso começa com os geradores verdadeiros de números aleatórios. Embora não exista um *caminho único* para projetar um TRNG ideal, aqui estão algumas diretrizes importantes:

- **modelo físico e entropia comprovável**: Um bom TRNG deve ser fundamentado em modelos físicos robustos, baseados em áreas como a física do estado sólido, fotônica ou mecânica quântica, capazes de demonstrar, mesmo aos mais céticos, a taxa de geração e a entropia dos bits gerados com base em propriedades físicas fundamentais. Essa confiabilidade, no entanto, não é plenamente assegurada em TRNGs que utilizam ruído Zener ou osciladores de anel, devido às limitações inerentes dessas abordagens.
- **defesa contra ataques físicos simples**: TRNG incorporados no *chip* devem ser protegidos contra ataques físicos básicos, como a injeção de ruído na fonte de alimentação, por exemplo, o TRNG da Intel, utilizado pela instrução RDRAND, parece ser extremamente sensível a esse tipo de interferência, segundo esquemas publicados e simulações SPICE;
- **proteção para TRNGs externos (USB)**: TRNG independentes, como os dispositivos USB, devem se proteger contra *hosts* mal-intencionados. Muitos TRNG baseados em USB podem ser facilmente atacados pelo *host*, o que permite ao invasor prever todos os *bits* aleatórios gerados, sem que o usuário perceba o ataque. Esse tipo de vulnerabilidade pode ocorrer durante o transporte do dispositivo ou quando qualquer pessoa tem acesso físico ao aparelho [37];
- **verificação de integridade a cada inicialização**: o *firmware* interno do TRNG deve verificar a integridade do dispositivo em cada inicialização, garantindo que ele esteja funcionando conforme esperado;
- **verificação durante o processo de produção**: mesmo que se assume que o TRNG não falhará nem degradará após a produção, recomenda-se ao menos testar sua saúde durante a fabricação, usando o modelo físico das propriedades subjacentes;
- **desativação em caso de mau funcionamento**: um verificador de integridade deve interromper o acesso ao TRNG caso detecte mau funcionamento, mesmo que isso paralise o sistema, esse bloqueio ajuda a prevenir o uso de dados potencialmente inseguros.

2.5. Remoção de viés na saída do TRNG com extratores de aleatoriedade

A saída dos TRNGs geralmente passa por um *extrator de aleatoriedade* antes de ser usada em aplicações práticas, uma vez que a fonte física pode não estar gerando valores com entropia suficientemente alta. Um exemplo clássico de extrator de aleatoriedade foi proposto por John von Neumann, onde o algoritmo do extrator (implementado em hardware ou software) avalia *bits* sucessivos gerados pelo TRNG. Se dois *bits* consecutivos forem iguais, nada é gerado; se forem diferentes, apenas o primeiro *bit* é usado. Isso transforma uma sequência como

$$00\ \ 11\ \ 00\ \ 10\ \ 01\ \ 01\ \ 00\ \ 00\ \ 10\ \ 00\ \ 01\ \ 10\ \ 10\ \ 01\ \ 00$$

em

$$1\ 0\ 0 \qquad 1 \qquad 0\ 1\ 1\ 0.$$

Embora essa saída tenha menos *bits*, ela possui maior entropia, tornando o resultado mais imprevisível. Um exemplo mais sofisticado de extração de aleatoriedade envolve o uso de matrizes de Toeplitz [38]. Essas matrizes atuam como extratores de aleatoriedade ao transformar uma sequência de bits bruta e possivelmente correlacionada, proveniente de um gerador quântico de números aleatórios (QRNG), em uma nova sequência de bits com alta entropia e distribuição uniforme.

A característica fundamental das matrizes de Toeplitz ($T \in \mathbb{C}^{n\times n}$) é que cada diagonal descendente da esquerda para a direita tem valor constante, o que permite realizar uma combinação linear eficiente dos bits de entrada:

$$T = \begin{pmatrix} t_0 & t_{-1} & t_{-2} & t_{-3} & \cdots & t_{-(n-1)} \\ t_1 & t_0 & t_{-1} & t_{-1} & \cdots & t_{-(n-2)} \\ t_2 & t_1 & t_0 & t_{-1} & \cdots & t_{-(n-3)} \\ t_3 & t_2 & t_1 & t_0 & \cdots & t_{-(n-4)} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ t_{(n-1)} & t_{(n-2)} & t_{(n-3)} & t_{(n-4)} & \cdots & t_0 \end{pmatrix} \tag{2}$$

Essa multiplicação gera uma sequência final que dilui padrões ou correlações presentes na entrada, resultando em uma saída estatisticamente segura e imprevisível.

Além disso, graças à sua estrutura repetitiva, a multiplicação utilizando matrizes de Toeplitz pode ser realizada com eficiência computacional de ordem $O(n \cdot \log n)$, o que torna sua aplicação prática e rápida. Por esse motivo, elas são amplamente utilizadas no pós-processamento de QRNGs e em protocolos de criptografia quântica, garantindo uma geração de números verdadeiramente aleatórios e altamente seguros.

Essas diretrizes ajudam a orientar o design de TRNGs mais seguros e confiáveis, garantindo que a aleatoriedade essencial para a criptografia seja protegida e monitorada em todos os momentos.

 DOI: https://doi.org/10.1590/1806-9126-RBEF-2024-0469

3. Geradores Pseudoaleatórios de Números (PRNG)

Quando a geração de números verdadeiramente aleatórios apresenta limitações de velocidade ou complexidade, os geradores de números pseudoaleatórios (PRNGs) oferecem uma solução eficiente, combinando rapidez e praticidade ao emular aleatoriedade para diversas aplicações.

PRNGs são algoritmos que recebem um número inicial (ou uma série de números) chamado de **semente** (*seed*) como entrada, realizam cálculos sobre ele e geram uma longa sequência de números "aleatórios" com base nessa semente. São considerados determinísticos, pois a mesma semente sempre fará com que o PRNG produza exatamente a mesma sequência de números.

O mesmo valor de semente em um PRNG sempre gerará a mesma sequência de saída. Isso contrasta com os TRNGs, onde é impossível replicar a saída, pois as entradas são processos físicos estocásticos, em vez de um único valor inicial. Mas para responder a esta contradição em termos, Hamming esclarece da seguinte forma:

> *Para fins práticos, somos forçados a aceitar o conceito desconfortável de 'relativamente aleatório', que significa que, em relação ao uso proposto, não vemos motivo para que os resultados não se comportem como se fossem aleatórios (como geralmente é exigido pela teoria). Este conceito é altamente subjetivo e pouco aceitável para os puristas, mas é algo ao qual os estatísticos recorrem regularmente ao selecionar uma 'amostra aleatória'. Uma vez escolhida, essa amostra é finita e definida, deixando de ser aleatória – espera-se, no entanto, que os resultados obtidos tenham propriedades aproximadamente equivalentes às de um cômputo completo de todo o espaço amostral considerado pela teoria* [39].

3.1. Exemplo: implementando geradores congruências lineares

Um PRNG bem simples pode ser criado usando apenas a eq. (3) [40]:

$$X_{n+1} = (aX_n + c)\ mod[m] \tag{3}$$

em que X é a sequência de valores "*aleatórios*" gerados, e:

- m, onde $0 < m$, é o *módulo*,
- a, onde $0 < a < m$, é o *multiplicador*,
- c, onde $0 < c < m$, é o *incremento*,
- X_0, onde $0 < X_0 < m$, é a *semente* ou valor inicial.

Essa equação é chamada de *gerador congruencial linear* (Linear Congruential Generator – LCG) porque os novos números são gerados a partir de valores anteriores de forma linear. O código apresentado a seguir corresponde a um gerador de números aleatórios (RNG) baseado em LCGs. Uma vez que um LCG é um PRNG, ele é determinístico, o que significa que podemos usar um RNG de referência para comparar nossa saída. Desde que usemos a mesma semente, nossa saída deve corresponder à saída gerada por um RNG similar.

A biblioteca padrão do C++ inclui diversos PRNGs, que são acessados por meio do cabeçalho #include <random>. Entre esses geradores, o *minstd_rand* é um PRNG que usa o método LCG, o qual aplica uma fórmula matemática simples para produzir uma sequência de números baseados em uma semente inicial. Essa biblioteca foi projetada para permitir a geração consistente e controlada de números pseudoaleatórios em diferentes aplicações.

A criação de um objeto *minstd_rand* exige uma semente inicial, que é usada para gerar uma sequência determinística de números pseudoaleatórios. A semente pode ser um número qualquer (o número 42 é usado como exemplo no código) e, ao reutilizar a mesma semente, a sequência gerada será idêntica em cada execução. Isso é especialmente útil para testes e comparações.

Uma vez inicializado, o gerador *minstd_rand* pode produzir números chamando o operador. Cada chamada gera o próximo número da sequência, de acordo com o algoritmo de LCG.

```
"""
Created on Thu Nov 28 18:22:05 2024
@author: Quantum ready
"""

import random

# Criando o gerador minstd\_rand (LCG equivalente
    em Python)

lcg_rand = random.Random()
lcg_rand.seed(42)  # Configurando a semente

# Gerando os n\'{u}meros aleat\'{o}rios

random_numbers = [lcg_rand.randint(0, 2**31 - 1)
    for _ in range(40)]

# Imprimindo os n\'{u}meros gerados

print(", ".join(map(str, random_numbers[:-1])) + f
    ", {random_numbers[-1]}")
```

Ao usar *random* (equivalente ao minstd_rand em C++) com uma semente fixa, é possível gerar uma sequência de referência para validar implementações em diferentes plataformas ou linguagens. Essa prática é útil em testes de consistência, em que a previsibilidade da sequência ajuda a identificar eventuais discrepâncias.

O *random* em Python (como *minstd_rand* em C++) é um gerador rápido e eficiente, mas sua qualidade de aleatoriedade é limitada para aplicações mais complexas, como criptografia ou simulações de alta precisão:

$$m = 2^{31} - 1, \quad a = 48271, \quad c = 0. \tag{4}$$

O valor $a = 7^5 = 16807$ foi originalmente projetado para uso na família de computadores IBM 360 [41].

Este gerador é amplamente utilizado e foi submetido a testes mais extensivos do que qualquer outro gerador de números aleatórios. É frequentemente recomendado para operações estatísticas e simulações [42, 43].

A vulnerabilidade do algoritmo é que, embora passe nos testes estatísticos de aleatoriedade, ele não possui nada de realmente aleatório. Na criptografia, se um terceiro souber que está sendo utilizado o algoritmo de congruência linear, bastaria conhecer uma pequena parte da sequência para determinar os parâmetros da equação 2. Uma vez descoberto um único número, seria possível determinar todos os números subsequentes.

Conhecendo X_0, X_1, X_2, X_3 com base no algoritmo:

$$X_1 = (aX_0 + c) \ \ mod[m], \tag{5}$$

$$X_2 = (aX_1 + c) \ \ mod[m], \tag{6}$$

$$X_3 = (aX_2 + c) \ \ mod[m] \tag{7}$$

seria possível para um espião resolver as equações (5–7) em a, c, m.

Portanto, embora seja conveniente utilizar um bom gerador de números aleatórios, é desejável garantir que a sequência utilizada não seja reprodutível, de forma que o conhecimento de uma parte da sequência não permita determinar os elementos subsequentes. Esse objetivo pode ser alcançado por meio de várias estratégias. Por exemplo, [44] sugere o uso do relógio interno do sistema para modificar o fluxo de números pseudorrandômicos. Uma maneira de utilizar o relógio consiste em reiniciar a sequência a cada N números, usando como semente o $mod[m]$ do valor corrente do relógio. Outra abordagem é simplesmente adicionar a cada número pseudorrandômico o $mod[m]$ do valor atual do relógio.

4. Tecnologia Quântica 2.0 e seu Impacto em Aplicações Práticas para os Geradores de Números Aleatórios

A mecânica quântica veio como uma tempestade para os pilares fundamentais de nossa compreensão clássica do mundo físico. Desde os seus primórdios na década de 1920, não só contribuiu para o nosso conhecimento, mas também produziu avanços tecnológicos que estão na base da nossa evolução. A teoria revolucionou a eletrônica e a ciência da informação com o transistor [45, 46], a comunicação óptica com o laser [47] e mais recentemente a química quântica com novos materiais [48] e drogas [49]. Atualmente, para muitos países os produtos quânticos representam uma parte importante do Produto Interno Bruto (PIB) e um grande negócio mundial [50].

Novos investimentos (veja Figura 10) estão movimentando a economia internacional, apostando em novas soluções tecnológicas e sistemas em que a tecnologia quântica é o principal motor do avanço.

No entanto, assim como as equações de Maxwell de 1863 levaram um século para dominar e começar a florescer em uma tecnologia generalizada, também é discutível que o mais recente progresso experimental na criação e manipulação de estados quânticos abra caminho para uma segunda revolução quântica que

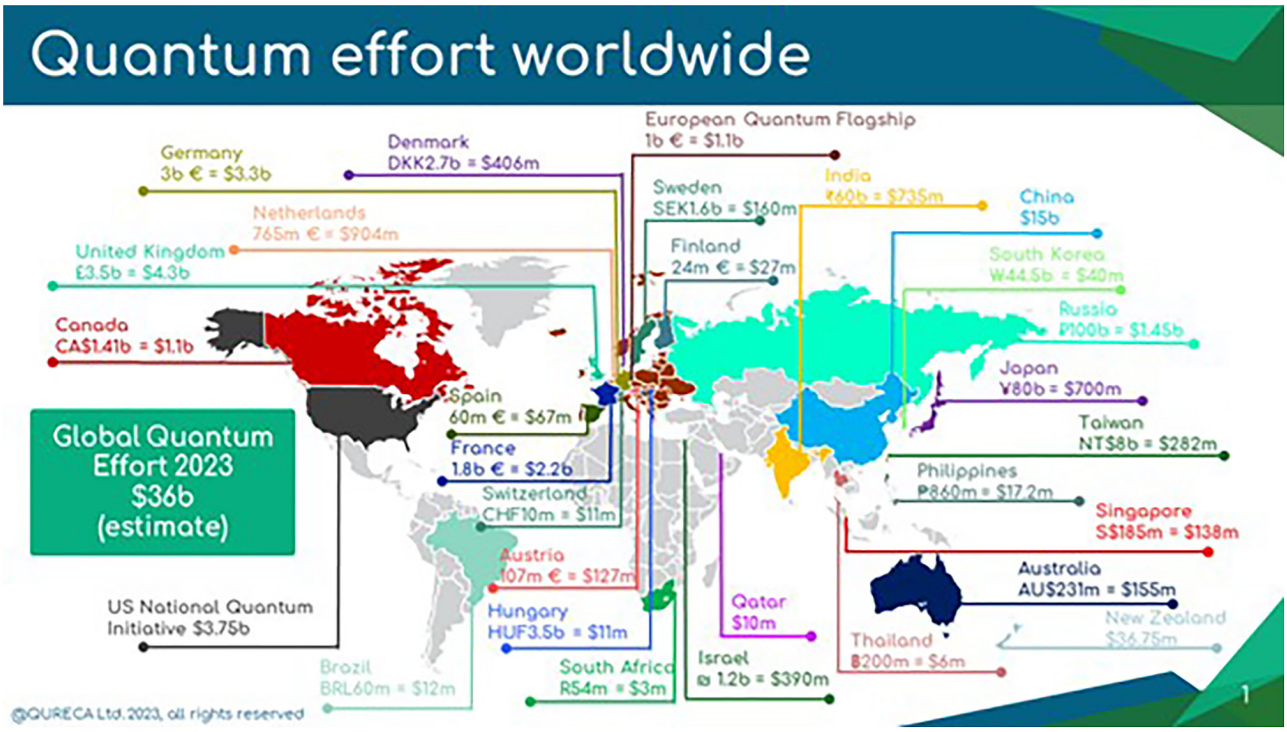

Figura 10: Os investimentos no mundo em pesquisa em tecnologia quântica – 2023 levantamento da empresa QURECA Ltd (Quantum initiatives worldwide – update 2023 – Qureca).

 DOI: https://doi.org/10.1590/1806-9126-RBEF-2024-0469

poderia moldar, novamente, nossa sociedade e economia. Demorou mais de meio século para a mecânica quântica começar a evoluir de uma teoria puramente física, com implicações até mesmo no nível filosófico, para uma ciência com um profundo impacto sobre como entendemos e processamos a informação; a base da nossa tecnologia da informação. O antigo paradigma de Shannon, segundo o qual o *bit* puramente matemático pode ser implementado no mundo físico, foi agora substituído pela visão de Landauer [51], na qual a informação não pode ser separada de sua corporificação física. Em 1982, o chamado "teorema da não clonagem" [52] confirma que novas regras devem ser consideradas quando a informação é codificada quanticamente. No mesmo ano, Feynman previu claramente a supremacia do computador quântico sobre as máquinas clássicas [53].

Em 1984, o primeiro protocolo de criptografia quântica foi publicado por Bennett e Brassard [54], permitindo a criação de chaves simétricas com sigilo perfeito entre os pontos finais de um canal capaz de transmitir correlações quânticas. Sua primeira implementação, embora primitiva, em 1989 [55] em um espaço livre de 25 cm, deu credibilidade à sua afirmação de ser uma tecnologia útil e iniciou na prática o campo da distribuição quântica de chaves (QKD).

No entanto, devido ao mesmo princípio de não clonagem que oferece sua segurança, a amplificação fiel dos sinais quânticos não é possível, limitando assim o QKD na distância máxima alcançável. É importante mencionar que o *no-cloning* não exclui a existência de repetidores quânticos, um dispositivo capaz de estabelecer correlações quânticas em distâncias ilimitadas sem realmente copiar estados, portanto, também capaz de liberar QKD de qualquer restrição física de distância ou perdas [56].

A criptografia quântica realmente ganhou mais atenção em 1994 com o lançamento do algoritmo quântico de Peter Shor [57], quando cientistas da computação e criptógrafos de todo o mundo entenderam que os computadores quânticos, se pudéssemos fazê-los, provavelmente quebrariam as cifras de chave pública existentes e outra criptografia. E se pudéssemos criá-los, seria apenas uma questão de quantos anos depois as quebras criptográficas começariam a acontecer. Enquanto um computador quântico com 53 qubits realizou em minutos cálculos que levariam milhares de anos em um computador clássico [58], os sistemas de distribuição de chaves quânticas já estão em estágio comercial e muitas empresas e governos têm investido pesadamente em pesquisas tecnologias de informação quântica, incluindo também aspectos importantes como sensoriamento e metrologia. Embora ainda existam questões importantes a serem resolvidas para sua ampla adoção, parece inquestionável que as tecnologias quânticas são chamadas a desempenhar um papel relevante no panorama tecnológico das próximas décadas.

A criptografia quântica finalmente entrou na era da engenharia e, de todas as tecnologias de informação quântica, é possivelmente a mais avançada [59]. No entanto, questões desafiadoras devem ser superadas antes que a rede de fibra óptica possa hospedar naturalmente essa inovação disruptiva para a segurança cibernética. Na prática, ao usar fibra óptica na terceira janela, conhecida como janela de érbio (1530 nm a 1565 nm) [60], as perdas ópticas de propagação são em torno de 0.2 dB/km no melhor caso. Isso significa que, após 15 km, a probabilidade de o sinal, tanto clássico quanto quântico, atingir a outra extremidade é de apenas 50%.

Além disso, em infra-estrutura óptica passiva, divisores, filtros, multiplexadores e outros componentes introduzem perdas adicionais, reduzindo a distância a áreas essencialmente metropolitanas. Se sinais clássicos e quânticos compartilham a mesma camada física, mistura de quatro ondas, espalhamento, reflexões ópticas e outros fenômenos ópticos complicam o cenário [61]. A comunicação óptica padrão realmente regenera o sinal por amplificação óptica ou por conversão eletro-óptica, mas infelizmente o processo de medição destrói o comportamento quântico no sinal e o teorema de não clonagem mencionado acima torna impossível copiar fielmente *bits* quânticos desconhecidos (note que se pode inicializar bits quânticos em um mesmo estado e aplicar as mesmas transformações, gerando bits replicados, mas conhecidos), portanto, em regime quântico as instalações clássicas são completamente inúteis. Finalmente, a copropagação de sinais clássicos e quânticos apresenta ainda outro problema, uma vez que fótons errantes de um pulso clássico que aparecem no canal quântico induzem uma grande taxa de erro que, rapidamente, impossibilita a execução bem-sucedida dos protocolos de comunicação quântica.

Apesar de todos esses problemas, as comunicações quânticas são hoje uma proposta viável que está ganhando cada vez mais força no mercado. A curto prazo, como uma tecnologia que oferece primitivas de segurança com propriedades únicas no nível físico e, a longo prazo, como forma de comunicar elementos de processamento quântico, criando o análogo quântico da Internet atual. Nesta primeira fase, um dos produtos de interesse, relevante tanto para aplicações de criptografia quântica quanto de computação, é o Gerador Quântico de Números Aleatórios (Quantum Random Number Generator – **QRNG**). Este dispositivo é baseado na manipulação do qubit, a unidade básica da computação quântica, para garantir uma geração de números totalmente aleatória.

Quantum Dice, ID Quantique e PsiQuantum são empresas que já estão oferecendo produtos comerciais de hardware baseados em tecnologia quântica para a geração de números aleatórios, enquanto empresas como IBM, Amazon e Microsoft oferecem serviços de software em nuvem para acessar hardware capacitado para atender a determinados tipos de demandas, com um conjunto de transformações de estado úteis para a computação quântica.

No próximo parágrafo, serão apresentadas as propriedades e características fundamentais de um qubit,

juntamente com o formalismo matemático necessário para compreender como a aleatoriedade está intrinsecamente incorporada nessa entidade.

4.1. Qubit: o elemento básico da computação quântica

O *qubit*, ou *bit* quântico, é a unidade fundamental de informação em um sistema quântico e desempenha um papel central na computação quântica, assim como o *bit* clássico é a unidade de informação na computação convencional. No entanto, o qubit possui propriedades exclusivas que o tornam significativamente mais poderoso e versátil em certos contextos computacionais.

4.1.1. Representação matemática do qubit

Assim como um *bit* clássico pode assumir os valores 0 ou 1, um qubit também pode representar esses dois estados, mas com uma diferença crucial: ele pode estar em uma combinação linear desses estados. Usando a *notação de Dirac*, os estados clássicos 0 e 1 são representados pelos *kets*:

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \tag{8}$$

$$|1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}. \tag{9}$$

Um qubit em superposição quântica pode ser descrito como:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle, \tag{10}$$

onde α e β são números complexos, conhecidos como *amplitudes de probabilidade*, que determinam a probabilidade de encontrar o qubit no estado $|0\rangle$ ou $|1\rangle$ após uma medição. A relação de normalização $|\alpha|^2 + |\beta|^2 = 1$ garante que a soma das probabilidades de todos os estados possíveis seja igual a 1.

4.1.2. Propriedades fundamentais do qubit

1. **Superposição quântica**: a superposição é uma das características mais fundamentais do qubit. Diferentemente de um *bit* clássico, que está estritamente em 0 ou 1, o qubit pode estar em um estado intermediário. Essa propriedade permite que o qubit *contenha* simultaneamente informações clássicas dos estados 0 e 1. Por exemplo, se $\alpha = \beta = \frac{1}{\sqrt{2}}$, o qubit está em um estado de superposição equitativa:

$$|\psi\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle. \tag{11}$$

   Nesse estado, a probabilidade de medir $|0\rangle$ ou $|1\rangle$ é igual a 50%, o que é ideal para a geração de números aleatórios.

2. **Entrelaçamento quântico**: quando dois ou mais qubits estão correlacionados, eles podem ser colocados em um estado chamado estado quântico entrelaçado. Nesse estado, a medição de um qubit projeta instantaneamente o estado do outro, independentemente da distância entre eles. Essa propriedade é essencial para muitos algoritmos quânticos e protocolos de comunicação.
3. **Interferência quântica:** qubits podem interferir entre si de maneira controlada, o que é explorado em algoritmos quânticos para amplificar probabilidades de estados desejados enquanto atenuam estados indesejados.
4. **Medição e colapso de estado**: ao medir um qubit, ele colapsa para um dos estados da base na qual a medição foi realizada. Se, por exemplo, a medição do estado (10) for feita na base computacional $\{|0\rangle, |1\rangle\}$, o qubit colapsa com probabilidades dadas por $|\alpha|^2$ e $|\beta|^2$, respectivamente em $|0\rangle$ e $|1\rangle$. Esse colapso reflete a natureza probabilística da mecânica quântica.
5. **Escalabilidade e crescimento exponencial**: na computação clássica, o número de estados possíveis cresce exponencialmente com o número de bits. Por exemplo, 8 bits podem representar $2^8 = 256$ estados, mas apenas um estado pode ser acessado por vez. Já na computação quântica, n qubits podem estar simultaneamente em uma superposição de 2^n estados. Isso significa que:

   - Com 8 qubits, é possível representar simultaneamente todos os 256 estados clássicos possíveis;
   - Com 300 qubits, seria possível representar mais estados do que o número total de átomos no universo observável (2^{300}).

   Essa capacidade de armazenar e manipular informações de maneira exponencial é o que torna a computação quântica tão promissora para resolver problemas que são intratáveis em computadores clássicos.

4.1.3. Implicações físicas e tecnológicas

Embora a teoria do qubit seja fascinante e cheia de potencial, sua implementação prática apresenta desafios significativos, exigindo sistemas físicos que possam representar, manipular e preservar estados quânticos delicados em um ambiente controlado. Duas abordagens amplamente exploradas utilizam *elétrons* e *fótons*, cada uma com características e aplicações distintas. Qubits eletrônicos (como aqueles implementados em sistemas supercondutores, pontos quânticos e íons aprisionados) destacam-se por permitirem interações fortes e controladas entre qubits, sendo ideais para aplicações em computação quântica. No entanto, esses sistemas são particularmente sensíveis à decoerência (isto é, à

 DOI: https://doi.org/10.1590/1806-9126-RBEF-2024-0469

perda de propriedades quânticas), pois interações com o ambiente podem destruir os estados quânticos. Por outro lado, qubits fotônicos, que utilizam propriedades como polarização, estados temporais e modos espaciais dos fótons, oferecem uma robustez natural contra a decoerência devido à baixa interação dos fótons com o ambiente. Isso os torna particularmente adequados para comunicação quântica e redes quânticas distribuídas, onde a transmissão a longas distâncias é essencial. Apesar disso, a implementação de operações lógicas entre qubits fotônicos ainda apresenta desafios consideráveis, exigindo avanços em tecnologias ópticas e mediadores quânticos. Essas abordagens refletem um equilíbrio fundamental entre robustez e capacidade de manipulação, e ambas serão exploradas em maior profundidade nas próximas seções. Nos próximos dois parágrafos, exploraremos a física subjacente a essas duas diferentes abordagens tecnológicas e seu papel na geração de números aleatórios.

4.2. Qubit supercondutor

Os qubits supercondutores foram inicialmente escolhidos na computação quântica devido a uma combinação de fatores técnicos e práticos que os tornaram particularmente atrativos para implementação em grande escala. Sua principal vantagem está na velocidade de operação, com tempos de processamento extremamente curtos, da ordem de nanosegundos, o que permite realizar cálculos complexos de maneira rápida e eficiente.

Além disso, os qubits supercondutores se destacam pela escalabilidade, já que podem ser integrados em chips usando técnicas consolidadas da indústria de semicondutores, como deposição e litografia. Isso garante a fabricação em larga escala, aproveitando o know-how já existente na microeletrônica. Outra característica relevante é o controle preciso dos estados quânticos por meio de pulsos de micro-ondas, que permite a manipulação confiável e a leitura rápida dos resultados.

O pioneirismo da IBM e os avanços experimentais no início dos anos 2000 consolidaram os qubits supercondutores como a principal escolha para sistemas quânticos robustos, como o IBM Quantum System One. Essa escolha prática e eficiente impulsionou a tecnologia, permitindo que ela evoluísse rapidamente para aplicações concretas na computação quântica moderna.

O princípio de funcionamento do qubit supercondutor está na utilização de junções Josephson e circuitos LC supercondutores, que permitem manipular estados quânticos por meio de transições de energia controladas. Para entender a dinâmica de um circuito de qubit supercondutor, é útil, portanto, começar com a descrição clássica de um circuito LC.

Neste sistema, a energia oscila entre a energia elétrica no capacitor C e a energia magnética no indutor L. A energia instantânea e dependente do tempo de cada componente é derivada da corrente e da tensão:

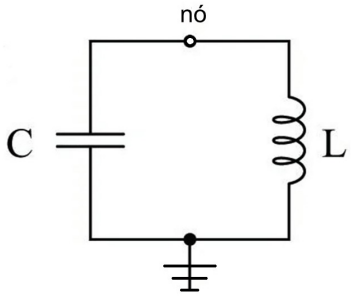

Figura 11: Circuito clássico LC.

$$E(t) = \int_{-\infty}^{t} dt'\, V(t')\, I(t'), \tag{12}$$

onde $V(t')$ e $I(t')$ representam a tensão e a corrente no capacitor ou no indutor.

Para derivar o Hamiltoniano clássico, seguimos a abordagem padrão da mecânica clássica: a formulação de Lagrange-Hamilton [62, 63]. Representamos os elementos do circuito em termos de uma de suas coordenadas generalizadas, como carga ou fluxo. Neste caso, escolhemos o fluxo, definido como a integral temporal da tensão:

$$\Phi(t) = \int_{-\infty}^{t} dt'\, V(t'), \tag{13}$$

em que o potencial no ponto (veja Figura 11) também corresponde à tensão ao longo do elemento e $V_c = V_L$.

Vale notar que poderíamos também associar a energia cinética com a coordenada de momento e a energia potencial com a coordenada de posição, usando a variável de carga Q(t), que é a integral temporal da corrente I(t).

Combinando as Equações (12) e (13), usando as relações $V = -L\frac{dI}{dt}$ e $I = C\frac{dV}{dt}$, e aplicando a fórmula de integração por partes, podemos expressar os termos de energia para o capacitor e o indutor em termos do fluxo no nó:

$$T_L = \frac{1}{2}C\dot{\Phi}^2 \quad U_C = \frac{1}{2L}\Phi^2 \tag{14}$$

Essas relações são fundamentais para estabelecer a correspondência entre a descrição clássica e a descrição quântica em sistemas como os qubits supercondutores, permitindo que, a partir do Hamiltoniano, possamos derivar o comportamento dinâmico do sistema com precisão.

A Lagrangiana é definida como a diferença entre os termos de energia cinética e energia potencial e, portanto, pode ser expressa pela equação (15):

$$\mathcal{L} = T_c - U_L = \frac{1}{2}C\dot{\Phi}^2 - \frac{1}{2L}\Phi^2 \tag{15}$$

A partir da Lagrangiana na equação (15), podemos derivar o Hamiltoniano usando a transformação de

Legendre, que requer o cálculo do momento conjugado ao fluxo, que, neste caso, é a carga no capacitor:

$$Q = \frac{\partial}{\partial \dot{\Phi}} \mathcal{L} = C\dot{\Phi} \tag{16}$$

Assim, o Hamiltoniano do sistema é dado por:

$$H = Q\dot{\Phi} - \mathcal{L} = \frac{1}{2}C\dot{\Phi}^2 + \frac{1}{2L}\Phi^2 \tag{17}$$

como esperado para um circuito elétrico LC. Este Hamiltoniano é análogo ao de um oscilador harmônico mecânico, onde $m = C$ e $\omega = \sqrt{\frac{1}{LC}}$ (frequência ressonante), que em coordenadas de posição x e momento p é expresso como:

$$H = \frac{1}{2m}p^2 + \frac{1}{2}m\omega^2 x^2. \tag{18}$$

O Hamiltoniano descrito acima é clássico. Para descrever o sistema em termos quânticos, é necessário promover as coordenadas de carga e fluxo a operadores quânticos. Enquanto as coordenadas clássicas satisfazem o parêntese de Poisson (expressões que definem relações entre grandezas dinâmicas em sistemas hamiltonianos, indicando como essas grandezas variam no tempo e que são fundamentais para descrever propriedades de conservação e simetrias em física clássica):

$$\{f, g\} = \frac{\partial f}{\partial \Phi}\frac{\partial g}{\partial Q} - \frac{\partial g}{\partial \Phi}\frac{\partial f}{\partial Q} \tag{19}$$

os operadores quânticos correspondentes obedecem a uma relação de comutação:

$$[\hat{\Phi}, \hat{Q}] = \hat{\Phi}\hat{Q} - \hat{Q}\hat{\Phi} = i\hbar \tag{20}$$

onde os operadores são indicados pelo *chapéu*. Em um circuito LC, tanto o indutor L quanto o capacitor C são elementos lineares. Definindo o fluxo reduzido $\hat{\phi} = 2\pi\frac{\hat{\Phi}}{\Phi_0}$ e a carga reduzida $\hat{n} = \frac{\hat{Q}}{2e}$ [64], podemos escrever o Hamiltoniano quântico para o circuito como (a partir das equações 16 e 17):

$$\hat{H} = 4\frac{e^2}{2C}\hat{n}^2 + \frac{1}{2L}\left(\frac{\Phi_0}{2\pi}\right)^2\hat{\phi}^2, \tag{21}$$

em que $E_c = \frac{e^2}{2C}$ é a energia de carga para adicionar cada elétron de um par de Cooper[1] à ilha[2] [65, 66], e $\frac{1}{2L}\left(\frac{\Phi_0}{2\pi}\right)^2$ é a energia indutiva, com $\Phi_0 = \frac{h}{2e}$ sendo o quantum de fluxo magnético supercondutor. O operador $\hat{n}$ é o *número excedente de pares* de Cooper na região isolada do supercondutor (chamada também de ilha[2] e fisicamente localizada entre C e J na Figura 12B), enquanto $\hat{\varphi}$ é a *fase invariante de calibre* no indutor. Estes formam um par conjugado canônico, obedecendo à relação de comutação $[\hat{\phi}, \hat{n}] = i$.

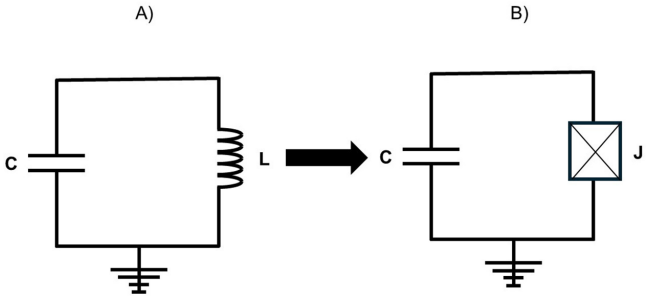

Figura 12: (A) circuito clássico LC linear; (B) circuito com junção Josephson não linear (circuito qubit transmon).

O Hamiltoniano da equação (21) é idêntico ao de uma partícula em um potencial quadrático unidimensional, o oscilador harmônico quântico (QHO). Podemos tratar $\hat{\phi}$ como a coordenada de posição generalizada, de forma que o primeiro termo representa a energia cinética e o segundo, a energia potencial. A forma funcional da energia potencial influencia as soluções de autovalor. Por exemplo, o termo quadrático $\frac{1}{2L}\left(\frac{\Phi_0}{2\pi}\right)^2\hat{\phi}^2$ gera o potencial da Fig. 14, na qual a solução do problema de autovalor resulta em uma série infinita de autoestados $|k\rangle$, com $k = 0, 1, 2, \ldots$, cujas autoenergias E_k estão igualmente espaçadas: $E_{k+1} - E_k = \hbar\omega_r$, com ($\omega_r = \sqrt{\frac{8E_L E_c}{\hbar^2}} = \sqrt{\frac{1}{LC}}$), a frequência ressonante do sistema.

Podemos representar esses resultados de forma mais compacta usando a *segunda quantização* [66] para o Hamiltoniano do QHO:

$$\hat{H} = \hbar\omega_r\left(\hat{a}^+\hat{a} + \frac{1}{2}\right) \tag{22}$$

em que $\hat{a}$ e $\hat{a}^+$ são, respectivamente, os operadores de criação e aniquilação de uma excitação do ressonador e $\hat{n} = \sqrt[4]{\frac{E_L}{32E_c}}i(\hat{a} - \hat{a}^+)$, $\hat{\phi} = \sqrt[4]{\frac{2E_c}{E_L}}(\hat{a} + \hat{a}^+)$ são, respectivamente o operador de número de pares de Cooper (ou operador de carga normalizada) e o operador de fase do supercondutor [67]. O Hamiltoniano na equação (20) é expresso em termos de energia, muitas vezes prefere-se expressar a energia em termos de $\hbar\omega$, onde ω é a frequência angular, pois isso torna as equações mais convenientes na formulação quântica, especialmente ao descrever a dinâmica do sistema em termos de ressonância e acoplamento com outras interações dependentes de frequência. Os operadores $\hat{n}$ e $\hat{\phi}$ representam, respectivamente, as variáveis quantizadas de carga e fase do sistema, desempenhando o papel de observáveis conjugados canônicos no contexto da mecânica quântica.

[1] *par de Cooper*: dois elétrons com spins opostos que se ligam de forma fraca, com energia baixa e em relação a elétrons isolados, podem se mover sem resistência elétrica em material.

[2] *ilha de Cooper*: em um circuito supercondutor é uma região condutora isolada, geralmente feita de um material supercondutor, que está conectada a outras partes do circuito através de junções Josephson ou capacitores. Ela funciona como um reservatório discreto de carga que permite o acúmulo ou remoção de pares de Cooper inteiros, em vez de elétrons individualmente.

 DOI: https://doi.org/10.1590/1806-9126-RBEF-2024-0469

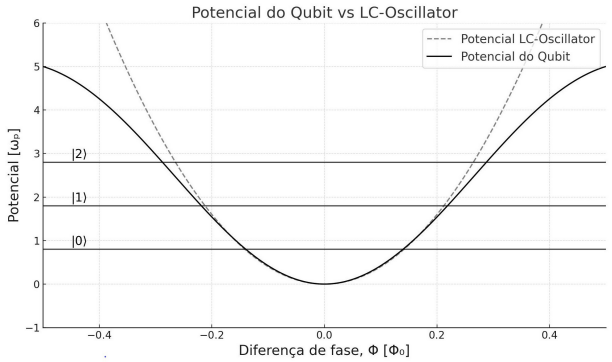

Figura 13: Potencial LC do oscilador clássico e potencial anarmônico do circuito com junção Josephson (circuito transmon).

Esses operadores caracterizam os estados quânticos do sistema, estabelecendo uma relação de incerteza semelhante à posição e ao momento em sistemas clássicos. Como consequência dessa relação de incerteza, mesmo no estado fundamental, ocorrem flutuações de ponto zero associadas a essas variáveis. Essas flutuações refletem a impossibilidade de definir simultaneamente e com precisão a carga e a fase, resultando em distribuições probabilísticas que permanecem não nulas mesmo na ausência de energia externa. Isso é uma manifestação direta do princípio da incerteza de Heisenberg aplicado ao sistema de carga e fase.

Na Figura 12B o indutor é substituído por uma junção Josephson [68].

As junções Josephson são elementos de circuito não lineares e sem dissipação que formam a base de circuitos supercondutores. Uma junção Josephson é formada ao separar dois eletrodos supercondutores por um isolante suficientemente fino, permitindo que pares de Cooper realizem tunelamento através da barreira. O efeito Josephson descreve a *supercorrente* que atravessa a junção, de acordo com as equações:

$$I = I_0 \sin\phi \quad V = \frac{\Phi_0}{2\pi}\frac{d\phi}{dt} \tag{23}$$

onde $\Phi_0 = \frac{h}{2e} = 2 \cdot 10^{-15} Wb$ e I_0 é a corrente crítica da junção.

A junção Josephson modifica o termo de potencial na equação da hamiltoniana quantizada (21):

$$\hat{H} = 4\frac{e^2}{2C}\hat{n}^2 + \frac{1}{2L}\left(\frac{\Phi_0}{2\pi}\right)^2\hat{\phi}^2 \rightarrow = 4\frac{e^2}{2C}\hat{n}^2 - \frac{\Phi_0}{2\pi}I_0\cos(\hat{\phi})$$
$$= 4E_c\hat{n}^2 - E_J\cos(\hat{\phi}) \tag{24}$$

introduzindo uma não linearidade. Para o circuito na Figura 13B, conhecido como *circuito transmon* na literatura, tipicamente os valores correspondentes são $C = 80\ fF$ e, portanto, $\frac{E_c}{h} = 240 MHz$, enquanto $I_c = 40 nA$ com $\frac{E_J}{h} = 20 GHz$. Com a definição de $\omega_p = \sqrt{\frac{8E_J E_C}{\hbar^2}}$ (*frequência de plasma*) e $\eta = \sqrt{\frac{E_c}{8E_J}}$ (*parâmetro*) podemos reescrever a equação da hamiltoniana na seguinte forma:

$$H_{qubit} = \hbar\omega_p 4\eta\hat{n}^2 - 8\hbar\frac{\omega_p}{\eta}\cos(\hat{\phi}). \tag{25}$$

A energia potencial $-E_J\cos(\hat{\phi})$ pode ser aproximada como um potencial harmônico próximo ao mínimo em $\hat{\phi} = 0$ (veja Figura 13), tratando os termos de ordem superior como uma perturbação. Expandindo o cosseno em uma série de Taylor:

$$-E_J\cos(\hat{\phi}) = -E_J\left(1 - \frac{1}{2}\hat{\phi}^2 + \frac{1}{24}\hat{\phi}^4 + \dots\right) \tag{26}$$

o potencial apresenta as seguintes contribuições:

- termo $-E_J$ é constante e pode ser ignorado no Hamiltoniano total;
- o termo quadrático $\frac{E_J}{2}\hat{\phi}^2$ representa o potencial harmônico principal;
- o termo de ordem superior $-\frac{E_J}{24}\hat{\phi}^4$ representa a *perturbação anarmônica*.

O tratamento do potencial como uma perturbação é válido na região onde o termo anarmônico $\left(-\frac{E_J}{24}\hat{\phi}^4\right)$ é pequeno em relação ao termo harmônico $\left(\frac{E_J}{2}\hat{\phi}^2\right)$. Isso acontece quando a amplitude de $\hat{\phi}$ é suficientemente pequena. Em termos físicos:

1. **baixa energia**: quando o transmon está em estados de baixa energia (próximos do estado fundamental ou dos primeiros estados excitados): nessa região, as flutuações quânticas de $\hat{\phi}$ são pequenas;
2. **alta razão** $\frac{E_J}{E_c}$: a energia de Josephson E_J deve ser significativamente maior que a energia de carga $E_J\left(\frac{E_J}{E_c} \gg 1\right)$: essa condição garante que o sistema permaneça confinado próximo ao mínimo do potencial e que as flutuações de $\hat{\phi}$ sejam reduzidas (lembra-se que $\hat{\phi} = \sqrt[4]{\frac{2E_c}{E_L}}(\hat{a} + \hat{a}^+) \Rightarrow \sqrt[4]{\frac{2E_c}{E_J}}(\hat{a} + \hat{a}^+)$). Quando o circuito atinge o regime contrário $\left(\frac{E_J}{E_c} \ll 1\right)$, o qubit passa a ser extremamente sensível ao ruído de carga, como flutuações no número de pares de Cooper ou cargas parasitas induzidas pelo ambiente. Essas flutuações levam a decoerência rápida, comprometendo a estabilidade do qubit.

A perturbação anarmônica permite separar os níveis de energia, eliminando a degenerescência entre sistemas próximos, como os estados $|0\rangle$, $|1\rangle$ e $|2\rangle$. Dessa forma, ela isola um sistema de dois níveis ($|0\rangle$ e $|1\rangle$) de outro sistema ($|1\rangle$ e $|2\rangle$), garantindo que a frequência de transição $\omega_{01}(|0\rangle \rightarrow |1\rangle)$ seja diferente da frequência de transição $\omega_{12}(|1\rangle \rightarrow |2\rangle)$.

Esse desacoplamento gera um subespaço de Hilbert $\mathcal{H}_2$ (veja Figura 14) onde vive o qubit, que pode oscilar entre dois valores de energia distintos: E_0 (associado ao estado $|0\rangle$) e E_1 (associado ao estado $|1\rangle$).

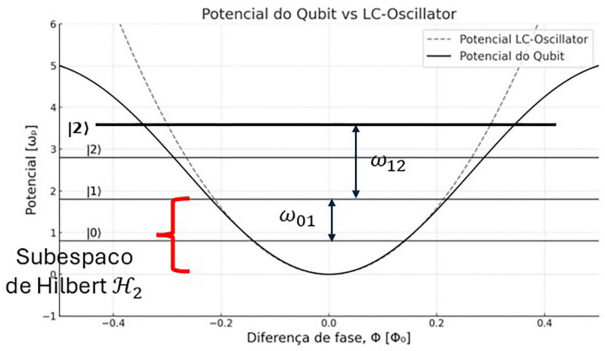

Figura 14: Subespaço de Hilbert $\mathcal{H}_2$ de dois níveis ($|0\rangle$, $|1\rangle$) sem degenerescência em relação ao subespaço adjacente de dois níveis ($|1\rangle$, $|2\rangle$) sendo $\omega_{12} > \omega_{01}$.

A anarmonicidade garante que o qubit permaneça confinado nesses dois níveis de energia, evitando transições indesejadas para o próximo nível $|2\rangle$, o que é fundamental para a estabilidade e controle do sistema quântico.

4.2.1. Manipulando qubits: as portas lógicas em computação quântica

Em computação, uma porta lógica (*gate*) é um elemento básico de processamento de informação, responsável por realizar operações específicas sobre unidades de dados. Na computação clássica, os *gates* operam sobre *bits*, unidades que assumem valores binários discretos, 0 ou 1. Cada gate clássico aplica uma transformação bem definida, como a inversão do valor do bit (**NOT**) ou a combinação lógica de dois bits (**AND**, **OR**) [69].

Na computação quântica, os *gates* desempenham um papel análogo ao dos *gates* clássicos, mas com funcionalidades ampliadas para operar diretamente sobre os estados quânticos dos qubits. Essas operações aproveitam as propriedades únicas dos qubits, como superposição e emaranhamento, possibilitando transformações complexas que habilitam o processamento paralelo de informações. Essa abordagem permite executar algoritmos que ultrapassam as limitações dos sistemas computacionais clássicos, oferecendo maior eficiência em problemas específicos.

Um *gate* quântico é formalmente representado como uma operação matemática que altera o estado de um qubit ou de um sistema de qubits. Esses *gates* são descritos por matrizes unitárias, garantindo que a transformação preserve as propriedades quânticas do sistema, como a normalização da função de onda e a reversibilidade do processo. Essa reversibilidade é uma característica essencial dos sistemas quânticos, em contraste com muitas operações irreversíveis encontradas em sistemas clássicos [70].

Um exemplo emblemático de *gate* quântico é o *Hadamard Gate*, que desempenha um papel crucial na computação quântica ao gerar estados de superposição. Essa operação será detalhada em seções posteriores, ilustrando como os *gates* quânticos são fundamentais para a realização de operações complexas em sistemas quânticos.

4.2.2. O processo de medição na computação quântica

Na computação quântica, o processo de medição desempenha um papel central ao permitir a extração de informações do sistema quântico. Diferentemente da computação clássica, no qual a leitura dos dados é uma operação trivial que não altera o estado do sistema, na mecânica quântica, a medição é um fenômeno intrinsecamente disruptivo e probabilístico [71]. Quando um qubit é medido, ele "*colapsa* de um estado de superposição (10) para um estado clássico definido, de acordo com as probabilidades associadas às amplitudes da função de onda.

Matematicamente, a medição de um estado quântico (10) resulta em (8) com probabilidade $|\alpha|^2$ e (9) com probabilidade $|\beta|^2$. Após a medição, o estado quântico original é irreversivelmente alterado para o estado observado. Esse fenômeno é conhecido como *colapso da função de onda* e reflete a natureza fundamentalmente probabilística da mecânica quântica.

Além de colapsar a superposição, o processo de medição também pode destruir outras propriedades quânticas do sistema, como o emaranhamento entre qubits. Por essa razão, a medição é uma operação que deve ser planejada com extremo cuidado em algoritmos quânticos. Um erro na escolha do momento ou do tipo de medição pode comprometer as informações quânticas processadas, anulando os benefícios do paralelismo quântico.

No contexto da computação quântica, há também diferentes tipos de medições que podem ser realizadas. A mais comum é a *medição computacional*, na qual o estado do qubit é projetado na base $|0\rangle$ e $|1\rangle$. No entanto, medições em outras bases ou mesmo parciais de sistemas multi-qubit podem ser empregadas, dependendo do objetivo do algoritmo ou do circuito quântico.

Por fim, a medição não é apenas um mecanismo de leitura, mas também uma ferramenta de controle. Em alguns algoritmos quânticos, os resultados das medições são usados para guiar operações subsequentes, demonstrando como a interação entre *gates* e medições pode ser explorada para resolver problemas complexos de maneira eficiente. Essa dualidade entre transformação e extração de informação faz do processo de medição um dos pilares mais intrigantes e desafiadores da computação quântica.

4.2.3. O Hadamard gate: criando e explorando superposição

O *Hadamard gate* (**H**) é um dos operadores fundamentais na computação quântica, desempenhando um papel crucial na geração de estados de superposição. Sua importância reside na capacidade de preparar qubits em estados que possibilitam o paralelismo quântico.

 DOI: https://doi.org/10.1590/1806-9126-RBEF-2024-0469

Matematicamente, o Hadamard gate é representado pela matriz unitária:

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \tag{27}$$

Quando aplicado a um qubit no estado $|0\rangle$ ou $|1\rangle$, ele transforma o estado inicial em uma superposição linear uniforme dos dois estados base:

$$|0\rangle \rightarrow \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle), \tag{28}$$

$$|1\rangle \rightarrow \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle). \tag{29}$$

Essa transformação implica que, após a aplicação do Hadamard gate, o qubit terá uma probabilidade igual de 50% de ser medido como $|0\rangle$ ou $|1\rangle$, caso seja submetido a uma medição na base computacional. O Hadamard gate transforma um estado da base computacional em um estado na base de Hadamard e vice-versa. Se um qubit começa no estado $|0\rangle$, a aplicação de Hadamard o coloca em uma superposição $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$, que é um estado indefinido na base computacional com incerteza controlada, mas bem definido na base de Hadamard. De maneira análoga, se um qubit está inicialmente em $|+\rangle$, a porta Hadamard o transforma em $|0\rangle$, que é bem definido na base computacional.

Além de criar superposição, o Hadamard gate desempenha um papel essencial em algoritmos quânticos importantes, como o algoritmo de Grover [72] para busca em bases não ordenadas e o algoritmo de Shor para fatoração de números inteiros. Ele também é utilizado em protocolos de comunicação quântica, como o BB84 para criptografia, no qual a criação e manipulação de estados de superposição são fundamentais.

O Hadamard gate também pode ser visualizado como uma operação que transforma a base computacional $(|0\rangle, |1\rangle)$ em uma base alternativa, conhecida como a base de Hadamard $\left(\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle), \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)\right)$. Essa mudança de base é crucial para explorar a interferência quântica, um fenômeno essencial para a vantagem computacional da computação quântica.

Em suma, o Hadamard gate é um operador central no *design* de circuitos quânticos, sendo indispensável para habilitar o potencial quântico completo dos qubits, seja em algoritmos de otimização, aprendizado de máquina, protocolos de criptografia ou simulações físicas. Sua aplicação simboliza a essência da computação quântica: a combinação única de incerteza e controle, aproveitada para resolver problemas de forma mais eficiente do que qualquer abordagem clássica.

4.2.4. Sequências aleatórias na plataforma IBM com qubits supercondutores

Para gerar sequências de bits mais longas, como 4 ou 8 bits, configura-se um circuito quântico composto por

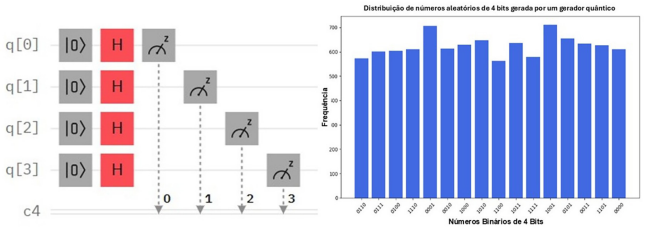

Figura 15: Circuito de geração de 4 bits a partir de 4 qubits na plataforma IBM Quantum (a esquerda), distribuição das ocorrências dos números aleatórios gerados (entre 0 e 15) depois 1024 rodadas do circuito (a direita).

múltiplos qubits. Cada qubit é submetido a uma porta Hadamard, que os coloca em estados de superposição. Após essa transformação, realiza-se uma medição independente de cada qubit, resultando em uma sequência binária aleatória. Esse procedimento é altamente escalável, permitindo a geração de sequências de qualquer comprimento, conforme o número de qubits disponíveis no sistema.

A plataforma *IBM Quantum* [73] fornece acesso remoto a sistemas de computação quântica, possibilitando que usuários desenvolvam e executem algoritmos quânticos via *cloud*. Para a geração de números aleatórios, configura-se um circuito quântico simples, no qual portas Hadamard são aplicadas a múltiplos qubits (veja Figura 13 à esquerda). Essa operação transforma cada qubit de um estado inicial bem definido $|0\rangle$ para um estado de superposição uniforme entre $|0\rangle$ e $|1\rangle$. O processo de medição projeta a sobreposição em um estado definido $|0\rangle$ ou $|1\rangle$ ambos com 50% de probabilidade.

Além de gerar números aleatórios, a IBM Quantum oferece ferramentas para análise dos resultados (veja Figura 15 à direita), permitindo verificar a qualidade da aleatoriedade gerada. Isso possibilita a criação de sequências mais complexas e com alta entropia, atendendo às exigências de aplicações que demandam aleatoriedade robusta e não previsível.

Cada bit de uma sequência gerada por esse método é estatisticamente independente dos demais, tornando essa abordagem altamente segura para aplicações que exigem aleatoriedade genuína. Por se basear em princípios fundamentais da física quântica, o método é inerentemente imprevisível, superando as limitações dos algoritmos pseudoaleatórios convencionais, que podem ser replicados e, eventualmente, decifrados. No material adicional foi disponibilizada uma sequência gerada 1024 vezes com 4 qubits (*sequencia_4QuBits_IBM_1024.txt*).

4.3. Qubit fotônico

O fóton, como quantum fundamental do campo eletromagnético, é um dos pilares da mecânica quântica moderna, representando a menor unidade de energia da luz e de outras formas de radiação eletromagnética.

Introduzido por Einstein em 1905 no contexto do efeito fotoelétrico [74], sua existência foi crucial para corroborar a ideia de quantização da luz, proposta inicialmente por Max Planck [75]. Essa abordagem revolucionária demonstrou que a energia da radiação eletromagnética é proporcional à sua frequência, $E = h\nu$, onde h é a constante de Planck. Além disso, o fóton apresenta características únicas, como *spin* igual a 1 (indicando sua natureza bosônica), ausência de massa de repouso e propagação à velocidade da luz no vácuo [76], características que o diferenciam como uma partícula elementar na física de partículas e na óptica quântica.

Na computação quântica e nas tecnologias quânticas em geral, o fóton desempenha um papel central, especialmente como qubit. A versatilidade do fóton como qubit deve-se à ampla gama de graus de liberdade que ele oferece para codificação de informação quântica, como polarização, fase, frequência, momento angular orbital e tempo de chegada. Esses graus de liberdade podem ser manipulados para representar os estados quânticos ($|0\rangle$ e $|1\rangle$), bem como superposições e emaranhamentos necessários para o processamento de informações quânticas.

Por exemplo, em sistemas baseados na polarização, os estados ortogonais $|H\rangle$ (polarização horizontal) e $|V\rangle$ (polarização vertical) formam uma base quântica para a representação de bits quânticos. Esses estados podem ser facilmente manipulados e medidos usando dispositivos ópticos como polarizadores, moduladores e detectores. Outra aplicação significativa é o uso do momento angular orbital do fóton, permitindo a criação de qubits com estados em dimensões mais altas, conhecidos como qudits [77], que aumentam a capacidade de codificação de informação quântica em comparação aos sistemas binários.

A robustez dos fótons contra a decoerência, devido à sua baixa interação com o ambiente, os torna ideais para aplicações em comunicação quântica, como o protocolo BB84 de criptografia quântica [54], e em redes quânticas, onde permitem a interligação de computadores quânticos através de canais de fibra ótica ou de espaço livre [78]. Sua capacidade de manter coerência quântica em longas distâncias já foi demonstrada experimentalmente, com recordes na transmissão de fótons em redes de fibra e em satélites para comunicação quântica global.

Além disso, tecnologias como fontes de fótons individuais, detectores de fótons e guias de onda integrados estão avançando rapidamente, possibilitando a construção de circuitos fotônicos para computação quântica e criptografia de alta eficiência. Sistemas baseados em fótons também viabilizam experimentos fundamentais, como a demonstração de teletransporte quântico e emaranhamento multipartido, essenciais para o avanço da física quântica e suas aplicações práticas [79].

4.3.1. O fóton e o divisor de feixe (componente óptico)

Considere um divisor de feixe (beam splitter – BS), que é um componente óptico feito de camadas finas de metal ou filmes dielétricos multicamadas depositadas sobre vidro, gerando dois feixes de saída para cada feixe de entrada (veja a Figura 16).

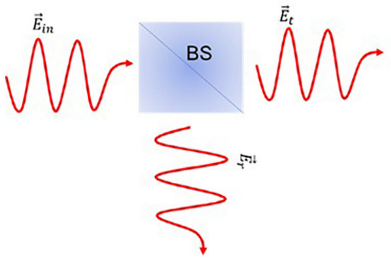

Figura 16: Divisor de feixe com radiação entrante $\vec{E}_{in}$.

$$\vec{E}_r = r\vec{E}_{in} \tag{30}$$

$$\vec{E}_t = t\vec{E}_{in} \tag{31}$$

$$|\vec{E}_{in}|^2 = |\vec{E}_r|^2 + |\vec{E}_t|^2 = (|r|^2 + |t|^2)|\vec{E}_{in}|^2 = 1|\vec{E}_{in}|^2. \tag{32}$$

$\vec{E}_i$ $(i = in, t, r)$ representam os campos elétricos da radiação eletromagnética na formulação clássica, r, t os coeficientes de reflexão e transmissão do divisor de feixe. O caso especial em que $t = r = \frac{1}{\sqrt{2}}$ descreve um divisor de feixe equilibrado 50/50. As fórmulas de (30) a (32) descrevem o comportamento óptico clássico, com conservação de energia, para esse componente passivo [80, 81]. Ao mover para o formalismo da ótica quântica, podemos substituir intuitivamente os campos pelos operadores de aniquilação/criação de bósons [76, 82]:

$$\vec{E}_{in} \propto \hat{a}_{in}, \tag{33}$$

$$\vec{E}_r \propto r\hat{a}_{in} = \hat{a}_r, \tag{34}$$

$$\vec{E}_t \propto t\hat{a}_{in} = \hat{a}_t. \tag{35}$$

As relações de comutação entre os operadores bósons seguem a álgebra de Weyl-Heisenberg, que é expressa pelas seguintes equações de comutação:

$$[\hat{a}_i, \hat{a}_j] = 0, \tag{36}$$

$$[\hat{a}_i^+, \hat{a}_j^+] = 0, \tag{37}$$

$$[\hat{a}_i, \hat{a}_j^+] = \delta_{ij}. \tag{38}$$

No entanto, os novos operadores geram diferentes regras de comutação:

$$[\hat{a}_r, \hat{a}_r^+] = |r|^2[\hat{a}_{in}, \hat{a}_{in}^+], \tag{39}$$

$$[\hat{a}_t, \hat{a}_t^+] = |t|^2[\hat{a}_{in}, \hat{a}_{in}^+], \tag{40}$$

$$[\hat{a}_r, \hat{a}_t^+] = rt^*[\hat{a}_{in}, \hat{a}_{in}^+]. \tag{41}$$

 DOI: https://doi.org/10.1590/1806-9126-RBEF-2024-0469

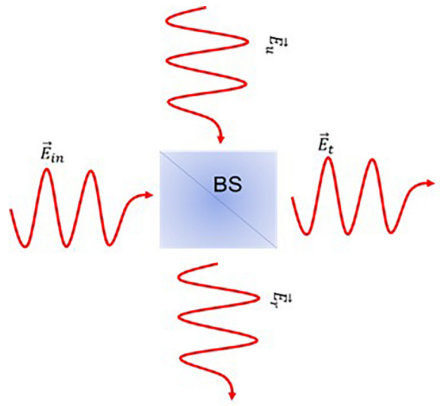

Figura 17: Divisor de feixe (BS) com radiação entrante $\vec{E}_{in}$ e $\vec{E}_u$.

Em outra aplicação, duas ondas são superpostas como em Figura 17.

$$\vec{E}_r = r\vec{E}_{in} + t'\vec{E}_u, \tag{42}$$

$$\vec{E}_t = t\vec{E}_{in} + r'\vec{E}_u, \tag{43}$$

$$\begin{aligned}|\vec{E}_{in}|^2 = |\vec{E}_r|^2 + |\vec{E}_t|^2 = (|r|^2 + |t|^2)|\vec{E}_{in}|^2 \\ + (|r'|^2 + |t'|^2)|\vec{E}_u|^2 + (rt^{'*} + tr^{'*})(\vec{E}_{in}\vec{E}_u^*) \\ + (t'r^* + r't^*)(\vec{E}_u\vec{E}_{in}^*).\end{aligned} \tag{44}$$

A formulação matricial [80] pode compactar a relação entre os feixes de entrada e saída:

$$\begin{pmatrix}\vec{\boldsymbol{E}}_{\boldsymbol{r}} \\ \vec{\boldsymbol{E}}_{\boldsymbol{t}}\end{pmatrix} = \begin{pmatrix}r & t' \\ t & r'\end{pmatrix}\begin{pmatrix}\vec{\boldsymbol{E}}_{\boldsymbol{in}} \\ \vec{\boldsymbol{E}}_{\boldsymbol{u}}\end{pmatrix}. \tag{45}$$

Quando r e t são coeficientes reais, a formulação matricial pode ser simplificada:

$$\begin{pmatrix}\vec{\boldsymbol{E}}_{\boldsymbol{r}} \\ \vec{\boldsymbol{E}}_{\boldsymbol{t}}\end{pmatrix} = \begin{pmatrix}\sqrt{\varepsilon} & \sqrt{1-\varepsilon} \\ \sqrt{1-\varepsilon} & -\sqrt{\varepsilon}\end{pmatrix}\begin{pmatrix}\vec{\boldsymbol{E}}_{\boldsymbol{in}} \\ \vec{\boldsymbol{E}}_{\boldsymbol{u}}\end{pmatrix}. \tag{46}$$

O sinal negativo na última componente matricial pode ser explicado por um deslocamento de fase de 180° em relação às outras ondas. Para uma camada dielétrica simples, isso corresponderia à reflexão da camada com o maior índice de refração. No entanto, essa não é a única escolha possível; diferentes espelhos têm propriedades distintas. Além disso, divisores de feixe reais podem introduzir perdas devido à absorção e dispersão interna. Isso pode ser descrito em termos de perdas $(1 - A)$

$$|\vec{E}_t|^2 + |\vec{E}_r|^2 = (1 - A)(|\vec{E}_{in}|^2 + |\vec{E}_u|^2). \tag{47}$$

Se descrevemos as propriedades de um divisor de feixe através da mecânica quântica, devemos começar assumindo que o feixe de entrada é um feixe descrito pelo operador $\hat{a}_{in}$. O divisor de feixe manterá as propriedades do modo, ou seja, a frequência da luz, o tamanho do feixe e a curvatura da frente de onda são todos preservados. O outro operador que precisamos introduzir é $\hat{a}_u$, para descrever quanticamente os modos de entradas do campo eletromagnético na segunda porta do divisor de feixe. Podemos ignorar a segunda porta de entrada no modelo clássico, mas mesmo que nenhuma energia flua por este segundo porto, no modelo quântico completo há energia de ponto zero no modo de vácuo que entra aqui e contribui para os dois modos de saída:

$$\hat{a}_r = r\hat{a}_{in} + t'\hat{a}_u, \tag{48}$$

$$\hat{a}_t = t\hat{a}_{in} + r'\hat{a}_u, \tag{49}$$

$$\begin{pmatrix}\hat{a}_r \\ \hat{a}_t\end{pmatrix} = \begin{pmatrix}r & t' \\ t & r'\end{pmatrix}\begin{pmatrix}\hat{a}_{in} \\ \hat{a}_u\end{pmatrix}. \tag{50}$$

Agora, analisamos as novas regras de comutação:

$$[\hat{a}_r, \hat{a}_r^+] = |r|^2 + |t'|^2, \tag{51}$$

$$[\hat{a}_t, \hat{a}_t^+] = |t|^2 + |r'|^2, \tag{52}$$

$$[\hat{a}_r, \hat{a}_t^+] = rt^* + t'r^{'*}, \tag{53}$$

$$[\hat{a}_t, \hat{a}_r^+] = tr^* + r't^{'*}. \tag{54}$$

As seguintes relações decorrem do princípio da reciprocidade em óptica e também podem ser derivadas do princípio de conservação de energia:

$$\begin{aligned}|r'| = |r|, \quad |t'| = |t|, \quad |r|^2 + |t|^2 = 1, \\ rt^* + t'r^{'*} = 0, \quad tr^* + r't^{'*} = 0.\end{aligned} \tag{55}$$

Nesse contexto, a álgebra de Weyl-Heisenberg é preservada para $\hat{a}_r$ e $\hat{a}_r^+$:

$$[\hat{a}_r, \hat{a}_r^+] = 1, \tag{56}$$

$$[\hat{a}_t, \hat{a}_t^+] = 1, \tag{57}$$

$$[\hat{a}_r, \hat{a}_t^+] = 0, \tag{58}$$

$$[\hat{a}_t, \hat{a}_r^+] = 0. \tag{59}$$

Uma solução possível para o modelo de divisor de feixe quântico pode ser a seguinte para um dispositivo equilibrado 50/50:

$$\hat{a}_r = \frac{1}{\sqrt{2}}\left(\hat{a}_u + e^{j\frac{\pi}{2}}\hat{a}_{in}\right) = \frac{1}{\sqrt{2}}(\hat{a}_u + j\hat{a}_{in}), \tag{60}$$

$$\hat{a}_t = \frac{1}{\sqrt{2}}\left(e^{j\frac{\pi}{2}}\hat{a}_u + \hat{a}_{in}\right) = \frac{1}{\sqrt{2}}(j\hat{a}_u + \hat{a}_{in}). \tag{61}$$

Quando não há estado na saída, podemos descrever o comportamento do divisor de feixe quântico da seguinte maneira:

$$|0\rangle_{in}|0\rangle_u BS \rightarrow |0\rangle_r|0\rangle_t. \tag{62}$$

Quando um único fóton entra no divisor de feixe $|1\rangle_{in}|0\rangle_u$, devemos considerar a transformação do operador sobre esse estado ao passar pelo componente óptico:

$$\begin{aligned}|1\rangle_{in}|0\rangle_u BS \rightarrow \frac{1}{\sqrt{2}}(ja_r^+ + a_t^+)|0\rangle_r|0\rangle_t \\ = \frac{1}{\sqrt{2}}(j|1\rangle_r|0\rangle_t + |0\rangle_r|1\rangle_t).\end{aligned} \tag{63}$$

A equação (63) descreve um estado emaranhado entre os modos r e t, em que cada modo possui uma probabilidade de 50% de conter um único fóton. Isso significa que, ao realizar uma medição em um dos modos e detectar a presença ou ausência do fóton, o estado do modo restante é imediatamente determinado devido à correlação quântica estabelecida entre eles.

Esse comportamento caracteriza um estado de Bell [83], que representa o máximo grau de emaranhamento possível entre dois sistemas quânticos, garantindo que a informação sobre um dos modos defina instantaneamente o estado do outro, independentemente da distância entre eles.

4.3.2. Quantis USB da IDQuantique

O Quantis USB (veja Figura 18) é um gerador de números aleatórios que se baseia em processos quânticos para produzir sequências de bits altamente imprevisíveis.

Com base na documentação oferecida pelo fabricante ID Quantique, ao detectar eventos discretos associados aos fótons, o dispositivo gera números com alta entropia e completa ausência de correlação [84]. Essa abordagem permite a criação de sequências de bits com propriedades ideais para aplicações que exigem máxima segurança e imprevisibilidade.

O dispositivo opera capturando o comportamento de fótons em um processo de divisão e detecção, onde cada fóton possui uma direção final aleatória devido às leis fundamentais da mecânica quântica como descrito no parágrafo anterior. Essa característica garante que os *bits* gerados sejam independentes e imprevisíveis, propriedades essenciais para aplicações como criptografia e simulações que dependem de dados aleatórios. Além disso, o Quantis USB gera até 4 Mbit/s, garantindo alta velocidade na produção de números aleatórios. Ele também conta com controle de erro em tempo real, assegurando a integridade dos dados gerados durante o processo.

O Quantis USB é projetado para ser escalável e de fácil integração, oferecendo uma interface USB para conectividade e fornecendo uma solução prática e eficiente para a geração de números aleatórios de alta qualidade [85].

Sua escalabilidade permite que seja facilmente adaptado a diferentes necessidades e volumes de dados, tornando-o adequado para uma ampla gama de aplicações, desde a criptografia quântica até a geração de sementes para algoritmos de simulação estocástica. Com desempenho consistente e alta capacidade de geração de números aleatórios, o dispositivo atende a diversos requisitos de segurança e eficiência. No material adicional foi disponibilizada uma sequência gerada com Quantis de 1024 bits (*sequencia_bits_quantis_usb_1024.txt*).

Figura 18: Dispositivo Quantis USB.

5. Conclusões

Neste trabalho, apresentamos uma análise abrangente de diferentes soluções tecnológicas para geração de números aleatórios (RNGs), abrangendo abordagens de hardware e software. Embora não tenhamos abordado todas as possibilidades existentes, destacamos como opções promissoras os geradores baseados no princípio de incerteza de Heisenberg, que exploram variáveis contínuas, e os geradores puramente algorítmicos, como aqueles que utilizam funções elípticas no contexto da criptografia pós-quântica. Esses temas, no entanto, serão aprofundados em futuras publicações.

O principal objetivo deste artigo foi organizar e categorizar as diversas tipologias de técnicas de RNG, evidenciando tanto a complexidade técnica do tema quanto sua relevância estratégica. Os RNGs representam uma tecnologia disruptiva em constante evolução, crucial em áreas como criptografia, segurança da informação, simulação científica e aplicações militares. No âmbito educacional, o tema assume uma importância ainda maior, pois proporciona uma plataforma para explorar conceitos fundamentais de aleatoriedade, incentivando estudantes e pesquisadores a desenvolverem soluções inovadoras e competitivas.

Com o mercado global em rápida expansão e a crescente demanda por sistemas mais seguros, eficientes e robustos, a inclusão de RNGs nos currículos e projetos de pesquisa pode estimular o desenvolvimento de tecnologias disruptivas que atendam às necessidades tecnológicas e estratégicas futuras.

Material suplementar

O seguinte material suplementar está disponível online:
Suplemento 1 – Material_complementar_1_sequencia_4QuBits_IBM_1024.
Suplemento 2 – Material_complementar_2_sequencia_bits_quantis_usb_1024.

Referências

[1] E. Moritz, *Games and Gambling in Antiquity: A Historical Survey* (Cambridge University Press, Cambridge, 1956).
[2] N. Metropolis e S. Ulam, J. Am. Stat. Assoc. **44**, 335 (1949).
[3] R.M. Karp, Discret. Appl. Math. **34**, 165 (1991).
[4] R. Motwani e P. Raghavan, ACM Comput. Surv. **28**, 33 (1996).
[5] C.E. Shannon, Bell Syst. Tech. J. **28**, 656 (1949).

[6] R. Gennaro, IEEE Secur. Priv. **4**, 64 (2006).
[7] L.E. Almeida, B.A. Fernández, D. Zambrano, A.I. Almachi, H.B. Pillajo e S.G. Yoo, *One-Time Passwords: A Literary Review of Different Protocols and Their Applications* (Springer, Berlin, 2024).
[8] L. von Ahn, M. Blum, N.J. Hopper e J. Langford. *CAPTCHA: Using Hard AI Problems for security* (Springer, Berlin, 2003).
[9] F. Rubin, *Secret key cryptography: chiphers, from simple to unbreakable* (Manning, Shelter Island, 2022).
[10]
[11] F. James, Comput. Phys. Commun. **60**, 329 (1990).
[12] W. Hörmann, J. Leydold e G. Derflinger, *Automatic nonuniform random variate generation* (Springer, Berlin, 2004).
[13] T.E. Hull e A.R. Dobell, SIAM Rev. **1**, 1 (1962).
[14] A. Rukhin, J. Soto, J. Nechvatal, M. Smid, E. Barker, S. Leigh, M. Levenson, M. Vangel, D. Banks, A. Heckert, J. Dray e S. Vo, *A Statistical Test Suite for random and pseudorandom number generators for cryptographic applications*, disponível em: https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=906762.
[15] DIEHARD, disponível em: https://tams.informatik.uni-hamburg.de/paper/2001/SA_Witt_Hartmann/cdrom/Internetseiten/stat.fsu.edu/diehard.html, acessado em: 27/10/2024.
[16] M. Haahr, *Introduction to randomness and random numbers*, disponível em: https://www.random.org/randomness/, acessado em: 27/10/2024.
[17] J.S. Lee e G.B. Cleaver, Heliyon **3**, e00422 (2017).
[18] H. Zhun e C. Hongyi, em: *ASICON 2001 – 4th International Conference on ASIC* (Xangai, 2001).
[19] M. Hamburg, P. Kocher e M.E. Marson, *Analysis of Intel's Ivy Bridge digital random number generator*, disponível em: https://cdn.atraining.ru/docs/Intel_TRNG_Report_20120312.pdf, acessado em: 27/10/2024.
[20] T. Stojanovski, J. Pihl e L. Kocarev, Circuits Syst. I. **48**, 382 (2001).
[21] CLOUDFLARE, disponível em: https://www.cloudflare.com/, acessado em: 27/10/2024.
[22] M.N. Bera, A. Acín e M. Kuś, Rep. Prog. Phys. **80**, 124001 (2017).
[23] X. Yuan, H. Zhou, Z. Cao e X. Ma. Phys. Rev. A **92**, 022124 (2015).
[24] M.M. Jacak, P. Jóźwiak, J. Niemczuk e J.E. Jacak, Sci. Rep. **11**, 16108 (2021).
[25] M. Herrero-Collantes e J.C. Garcia-Escartin, Rev. Mod. Phys. **89**, 015004 (2017).
[26] X. Ma, X. Yuan, Z. Cao, B. Qi e Z. Zhang, NPJ Quant. Inf. **2**, 1 (2016).
[27] R. Li, em: *Proceedings of the IEEE SoutheastCon 2015 – Fort Lauderdale* (Florida, 2015).
[28] W. Loveland, D.J. Morrisey e G.T. Seaborg, *Modern nuclear chemistry* (John Wiley & Sons, Hoboken, 2006).
[29] Os números aleatórios que guiam nossas vidas e a busca para encontrá-los, BBC News Brasil, São Paulo, 26 de julho de 2024, disponível em: https://www.bbc.com/portuguese/articles/c51y05zev73o, acessado em: 20/11/2024.
[30] R. Pazzetto, *Tutorial: Como fazer um gerador de números aleatórios – Parte 1 (Simulação do circuito)*, disponível em: https://www.youtube.com/watch?v=xgrhkGBh1a8, acessado em: 20/11/2024.
[31] S. Bhunia e M. Tehranipoor, *Hardware Security: A Hands-on Learning Approach* (Morgan Kaufmann, Burlington, 2018).
[32] S.H.M. Kwok e E.Y. Lam, em: *TENCON 2006 – 2006 IEEE Region 10 Conference* (Hong Kong, 2006).
[33] K. Wold e C.H. Tan, em: *International Conference on Reconfigurable Computing and FPGAs* (Cancún, 2008).
[34] B. Sunar, W.J. Martin e D.R. Stinson, IEEE Transaction Computers **56**, 109 (2007).
[35] GITHUB, *waywardgeek/infnoise*, disponível em: https://github.com/waywardgeek/infnoise, acessado em: 04/11/2024.
[36] GITHUB, *alwynallan/redoubler*, disponível em: https://github.com/alwynallan/redoubler, acessado em: 04/11/2024.
[37] IPESI, *Dispositivos USB podem ser a porta de entrada para ataques cibernéticos contra os ambientes de produção*, disponível em: https://ipesi.com.br/dispositivos-usb-podem-ser-a-porta-de-entrada-para-ataques-cibernéticos-contra-os-ambientes-de-producao/, acessado em: 23/12/2024.
[38] A. Böttcher e S.M. Grudsky, *Toeplitz Matrices, Asymptotic Linear Algebra, and Functional Analysis* (Birkhäuser Basel, Basel, 2012).
[39] R.W. Hamming, *The art of probability for scientists and engineers* (Addison-Wesley Publishing, Massachusetts, 1991).
[40] D. Lehmer, em: *Proceedings of a second symposium on large-scale digital calculating machinery* (Massachusetts, 1951).
[41] P. Lewis, A. Goodman e J. Miller, IBM System Journal **8**, 136 (1969).
[42] R. Jain, *The art of computer system performance analysis: techniques for experimental design, measurement, simulation and modeling* (Wiley, New York, 1991).
[43] C. Sauer e K. Chandy, *Computer Systems performances modeling* (Prentice Hall, Englewood Cliffs, 1981).
[44] H. Bright e R. Eninson, ACM Computing Surveys **11**, 357 (1979).
[45] J. Bardeen, Phys. Rev. **71**, 383 (1948).
[46] J. Bardeen e W.H. Brattain, Phys. Rev. **74**, 230 (1947).
[47] T.H. Maiman, Nat. **187**, 494 (1960).
[48] P.K. Barkoutsos, F. Gkritsis, P.J. Ollitrault, I.O. Sokolov, S. Woerner e I. Tavernelli, Chem. Sci. **12**, 4345 (2021).
[49] C.S. Tautermann, *em: Quantum Mechanics in Drug Discovery* (Springer, New York, 2020).
[50] S. Buchholz, D. Golden e C. Brown, *A business leader's guide to quantum technology*, disponível em: https://www2.deloitte.com/us/en/insights/topics/innovation/quantum-computing-business-applications.html/#endnote-sup-1, acessado em: 26/6/2023.
[51] R. Landauer, Physics Today **44**, 23 (1991).
[52] W. Wootters e W. Zurek, Nature **299**, 802 (1982).
[53] R.P. Feynman, International Journal of Theoretical Physics **21**, 467 (1982).

[54] G. Brassard e C.H. Bennett, em: *Proceedings of IEEE International Conference on Computers, Systems and Signal Processing* (New York, 1984).
[55] C.H. Bennett e G. Brassard, SIGACT News **20**, 78 (1989).
[56] H.J. Briegel, W. Dür, J.I. Cirac e P. Zoller, Phys. Rev. Lett. **81**, 5932 (1998).
[57] P.W. Shor, em: *Proceedings 35^{th} Annual Symposium on Foundations of Computer Science* (Santa Fe, 1994).
[58] F. Arute, K. Arya, R. Babbush, D. Bacon, J.C. Bardin, R. Barends, R. Biswas, S. Boixo, F.G.S.L. Brandao e D.A. Buell et al., Nature **574**, 505 (2019).
[59] F. Xu, X. Ma, Q. Zhang, H.K. Lo e J.W. Pan, Rev. Mod. Phys. **92**, 2 (2020).
[60] G.P. Agrawal, *Fiber-Optic Communication Systems* (John Wiley & Sons, New Jersey, 2021).
[61] J.M. Thomas, F.I. Yeh, J.H. Chen, J.J. Mambretti, S.J. Kohlert, G.S. Kanter e P. Kumar, arXiv:2404.10738v4 (2024).
[62] V. Hirvonen, *Lagrangian Mechanics for the Non-Physicist* (Independently published, Washington, 2023).
[63] H. Goldstein, C.P. Poole e J.L. Safko, *Classical Mechanics* (Addison-Wesley, Massachusetts, 2001), 3 ed.
[64] J.M. Martinis, Elsevier **79**, 487 (2004).
[65] S.J. Blundell, *Superconductivity: A Very Short Introduction*? (Oxford University Press, Oxford, 2009).
[66] R. Kleiner, W. Buckel e R. Huebener, *Superconductivity: An Introduction* (Wiley-VCH, Weinheim, 2015).
[67] P. Krantz, M. Kjaergaard, F. Yan, T.P. Orlando, S. Gustavsson e W.D. Oliver, Appl. Phys. Rev. **6**, 021318 (2019).
[68] J.B. Zuber e C. Itzykson, *Quantum field theory* (Dover books on Physics, New York, 2006).
[69] C.W. Kann III, *Digital circuit projects – an overview of digital circuit trough implementing circuits* (LibreTexts, California, 2023).
[70] F. Ghiglieno, P.H.D. Ferreira, V. Tribuzi e O.L. Silva Filho, em: *Systems Engineering – Design, Analysis, Programming, and Maintenance of Complex Systems*, editado por G. Lambert-Torres, G.C.C. Andrade e C.I.A. Costa (IntechOpen, London, 2024).
[71] A. Pres, *Quantum Theory: Concepts and Methods* (Springer, Heidelberg, 1995).
[72] L.K. Grover, em: *Proceedings of the twenty-eighth annual ACM symposium on Theory of Computing Search, Association for Computing Machinery* (Pennsylvania, 1996).
[73] IBM, disponível em: https://www.ibm.com/quantum, acessado em: 12/12/2024.
[74] A. Einstein, Annalen der Physik **322**, 132 (1905).
[75] M. Planck, Annalen der Physik **3**, 553 (1901).
[76] C. Tannoudji e C.E. Jacq, *Photons et atomes* (CNRS Édition, Paris, 1987).
[77] Y. Wang, Z. Hu, B.C. Sanders e S. Kais, Frontiers in Physics **8**, 589504 (2020).
[78] D. Bouwmeester, A. Ekert e A. Zeilinger, *The Physics of Quantum Information: Quantum Cryptography, Quantum Teleportation, Quantum Computation* (Springer, Heidelberg, 2000).
[79] J.L. O'Brien, A. Furusawa e J. Vučković, Nature Photonics **3**, 687 (2009).
[80] E. Hetch, *Optics* (Pearson, London, 2015).
[81] J.P. Pérez e E. Anterrieu, *Optique: fondaments et applications, avec 250 exercices et problèmes résolus* (Dunod, Paris, 2020).
[82] H.A. Bachor e T.C. Ralph, *A Guide to Experiments in Quantum Optics* (WILEY-VCH Verlag GmbH, Weinheim, 2004).
[83] M.A. Nielsen e I.L Chuang, *Quantum computation and quantum information* (Cambridge University Press, Cambridge, 2011).
[84] IDQ, *Quantis QRNG USB*, disponível em: https://www.idquantique.com/random-number-generation/products/quantis-random-number-generator/, acessado em: 14/12/2024.
[85] CENTRO INTERNACIONAL DE FÍSICA, *Nova geração de números aleatórios: uma abordagem quântica – aula 2*, disponível em: https://www.youtube.com/watch?v=ZGEOpHkuTZ0&list=PLukZFARoEvIY2L-PKN2hq5pmKKn3hI3Nw&index=12, acessado em: 15/12/2024.