

**UNIVERSIDADE ESTADUAL PAULISTA**  
**"JÚLIO DE MESQUITA FILHO"**  
**FACULDADE DE ENGENHARIA, CAMPUS DE SÃO JOÃO DA BOA VISTA**

**GABRIELA CRISTINA KENEDI PRESTES**

**ANÁLISE DA UTILIZAÇÃO DE CÓDIGOS DE PARIDADE EM MODELOS DE RAID**

São João da Boa Vista

2023

**Gabriela Cristina Kenedi Prestes**

**ANÁLISE DA UTILIZAÇÃO DE CÓDIGOS DE PARIDADE EM MODELOS DE RAID**

Trabalho de Conclusão de Curso apresentado ao Conselho de Curso de Engenharia Eletrônica e de Telecomunicações da Faculdade de Engenharia do Câmpus de São João da Boa Vista - Unesp, como parte dos requisitos para obtenção do diploma de Graduação em Engenharia Eletrônica e de Telecomunicações .

Orientadora: Prof<sup>a</sup> Dra. Cintya Wink de Oliveira Benedito

São João da Boa Vista  
2023

P936a Prestes, Gabriela Cristina Kenedi  
Análise da utilização de códigos de paridade em modelos de RAID  
/ Gabriela Cristina Kenedi Prestes. -- São João da Boa Vista, 2023  
72 p. : il., tabs.

Trabalho de conclusão de curso (Bacharelado - Engenharia de  
Telecomunicações) - Universidade Estadual Paulista (Unesp),  
Faculdade de Engenharia, São João da Boa Vista  
Orientadora: Cintya Wink de Oliveira Benedito

1. Teoria da informação. 2. Códigos corretores de erros (Teoria da  
informação). 3. Armazenamento de dados. 4. Proteção de dados. 5.  
Recuperação de dados (Computação). I. Título.

Sistema de geração automática de fichas catalográficas da Unesp. Biblioteca da Faculdade de  
Engenharia, São João da Boa Vista. Dados fornecidos pelo autor(a).

Essa ficha não pode ser modificada.

**UNIVERSIDADE ESTADUAL PAULISTA “JÚLIO DE MESQUITA FILHO”  
FACULDADE DE ENGENHARIA - CÂMPUS DE SÃO JOÃO DA BOA VISTA  
GRADUAÇÃO EM ENGENHARIA ELETRÔNICA E DE TELECOMUNICAÇÕES**

**TRABALHO DE CONCLUSÃO DE CURSO**

**ANÁLISE DA UTILIZAÇÃO DE CÓDIGOS DE PARIDADE EM MODELOS DE  
RAID**

Aluno: Gabriela Cristina Kenedi Prestes

Orientador: Prof.<sup>a</sup> Dr.<sup>a</sup> Cintya Wink de Oliveira Benedito

Banca Examinadora:

- Cintya Wink de Oliveira Benedito (Orientadora)
- Edgar Eduardo Benitez Olivo (Examinador)
- Marlon Rodrigues Garcia (Examinador)

A ata da defesa com as respectivas assinaturas dos membros encontra-se no prontuário do aluno (Processo nº 236/2023)

São João da Boa Vista, 15 de dezembro de 2023

## AGRADECIMENTOS

Agradeço à Deus, Nossa Senhora Aparecida e a Nossa Senhora Das Graças, por ouvirem minhas orações e terem me dado forças em cada momento de angústia.

Aos meus pais, Quirino Prestes e Irany Prestes, por todo amor, confiança, proteção e dedicação. O caminho não foi fácil, mas eles nunca mediram esforços para me apoiar durante cada fase da minha vida.

Aos meus irmãos, Thiago Prestes, Daniela Prestes, Eduardo Prestes, Isabella Prestes e Rafaella Prestes, por sempre me incentivar a ser uma pessoa melhor e serem meu alicerce.

Aos meus avós, Elisa Prestes, Eduardo Prestes, José Queiroz e Cacilda Queiroz, por todas as orações, pelas palavras de conforto, pelos conselhos e por todo apoio.

Ao Caio Elias, por me acompanhar em tantas aventuras e situação inesperada. Seu apoio e carinho foram fundamentais para superar os obstáculos que surgiram na minha trajetória durante últimos anos.

Aos meus amigos Alessandra Ramalho, Nicolas Puñales, Rosely Carvalho e Vanessa Santana pelo apoio, conselhos e carinho.

Aos meus amigos Ana Letícia Coradi, Homero Assaloni, Diógenes Ferreira, Vanessa Martão e André Vinhas que estiveram ao meu lado desde a minha primeira semana de aula. Naquele momento em que eu estava extremamente perdida e sozinha, eles foram os primeiros a me estender a mão. Agradeço profundamente por isso.

Aos meus amigos, Amanda Belchior, Carlos Aleixo, Débora Beatriz Zanitti, Grazielle Cossa, Jéssica Damaglio, Júlia Duarte, Larissa Yida, Leoanardo Bobadilla, Leonardo Marques, Letícia Borges, Letícia Dal'Cól, Nicholas Martins e a todos aqueles que passaram pela minha vida ao longo do curso, tornando os dias em São João mais divertidos e memoráveis.

Aos meus mestres Felipe e Rodrigo, por cada puxão de orelha e por sempre acreditarem no meu potencial. O apoio constante e os sábios conselhos que recebi de ambos foram elementos indispensáveis para que eu pudesse concluir mais um ciclo da minha vida.

À minha orientadora, Prof. Dra. Cintya Wink de Oliveira Benedito, por ter aceitado conduzir esse trabalho, com tanta paciência, persistência e carinho. A Cintya é um modelo a ser seguido, não apenas por sua excelência acadêmica, mas também por sua humanidade e empatia. Tenho certeza de que seu trabalho como educadora deixará um legado de gratidão em todos os que tiveram a sorte de serem orientados por você.

Aos professores da banca examinadora, Prof. Dr. Edgar Eduardo Benitez Olivo e Prof. Dr. Marlon Rodrigues Garcia, por aceitar o convite para avaliar este trabalho.

A todos os professores e professoras que passaram pela minha vida, expresso minha profunda admiração e respeito pela nobre profissão que vocês desempenham.

À UNESP, por transformar minha vida, por meio da educação.

Sou eternamente grata por Deus ter colocado cada uma dessas pessoas no meu caminho. Fico imensamente feliz em saber que tenho tantas pessoas e especiais para agradecer.

*“Nada vos será impossível.”  
(Matheus 17:20)*

## RESUMO

Vivemos na era digital, onde a importância dos dados para as empresas e a sociedade cresce a cada dia, tornando os dados um ativo de extrema relevância, pois a perda de dados pode resultar em grandes prejuízos. Diante desse cenário, são desenvolvidos métodos para enfrentar os desafios associados à perda de dados, em especial quando estamos utilizando sistemas de armazenamento distribuído (DSS, *Distributed Storage System*), e um desses métodos é conhecido como Arranjo Redundante de Discos Independentes (RAID, *Redundant Array of Independent Disks*). O RAID é um tipo de sistema de armazenamento de dados distribuído que agrupa várias unidades individuais, representadas por unidade de disco rígido (HDD, *hard disk drive*) ou unidade de estado sólido (SSD, *solid state drive*), em uma única unidade lógica para melhorar a taxa de transferência de leitura e gravação, confiabilidade, disponibilidade ou uma combinação destes. O RAID é fundamentado em três técnicas principais: distribuição, espelhamento e paridade. Essas técnicas visam proporcionar uma maior segurança e eficiência no armazenamento e recuperação de dados. O objetivo deste trabalho é apresentar diferentes níveis de RAID, explorando os códigos de bloco utilizados na implementação desses métodos para a proteção de dados. Entre os códigos de bloco discutidos, estão o código de Hamming, códigos de paridade em forma matricial, código de paridade linha-diagonal (RDP, *row-diagonal parity*) e código EVENODD. Aprofundando a discussão sobre RAID, o trabalho analisa e compara as principais funcionalidades de cada nível de RAID. Além disso, será implementada uma máquina virtual composta por dois arranjos configurados em RAID 5 e 6. Este experimento prático visa explorar em detalhes como cada sistema responde em termos de desempenho, contribuindo para uma compreensão mais aprofundada a esses níveis de RAID.

**PALAVRAS-CHAVE:** RAID. Sistema de Armazenamento Distribuído. Proteção de dados. Recuperação de dados. Paridade. Códigos Corretores de erro.

## **ABSTRACT**

We live in the digital age, where the importance of data for businesses and society is growing every day, making data a crucial asset. The loss of data can result in significant losses. In this context, methods are developed to address the challenges associated with data loss, especially when utilizing Distributed Storage Systems (DSS), and one of these methods is Redundant Array of Independent Disks (RAID). RAID is a type of distributed data storage system that combines multiple individual units, represented by hard disk drives (HDD) or solid state drives (SSD), into a single logical unit to enhance read and write transfer rates, reliability, availability, or a combination of these factors. RAID is based on three main techniques: distribution, mirroring, and parity. These techniques aim to provide greater security and efficiency in data storage and recovery. The objective of this work is to present different RAID levels, exploring the block codes used in the implementation of these methods for data protection. Among the discussed block codes are Hamming code, matrix-based parity codes, row-diagonal parity (RDP) codes and EVENODD codes. This work aims not only to deepen the understanding of data protection mechanisms provided by RAID but also to highlight the importance of block codes in preserving data integrity in computer systems. Deepening the discussion about RAID, the study analyzes and compares the main functionalities of each RAID level. Additionally, a virtual machine composed of two arrays configured in RAID 5 and 6 will be implemented. This practical experiment aims to explore in detail how each system responds in terms of performance, contributing to a more in-depth understanding of these RAID levels.

**KEYWORDS:** RAID. Distributed Storage System. Data Protection. Data Recovery. Parity. Error-Correcting Codes.

## LISTA DE ILUSTRAÇÕES

|           |                                                                                          |    |
|-----------|------------------------------------------------------------------------------------------|----|
| Figura 1  | Expectativa de crescimento de dados a nível global. . . . .                              | 15 |
| Figura 2  | Estrutura básica de código de bloco com redundância. . . . .                             | 17 |
| Figura 3  | Código de paridade unidimensional. . . . .                                               | 25 |
| Figura 4  | Código matricial de paridade bidimensional. . . . .                                      | 25 |
| Figura 5  | Código matricial de paridade diagonal $A(3, 3)$ . . . . .                                | 26 |
| Figura 6  | Exemplo de um código matricial com paridade diagonal $A(4, 4)$ . . . . .                 | 27 |
| Figura 7  | Código RDP para $C_p = 5$ . . . . .                                                      | 28 |
| Figura 8  | Demonstração do cálculo da paridade $P$ e $Q$ via código RDP. . . . .                    | 29 |
| Figura 9  | Arquitetura geral do código EVENODD. . . . .                                             | 30 |
| Figura 10 | Distribuição mensagem $u$ no arranjo de disco para matriz com $C_p = 5$ . . . . .        | 31 |
| Figura 11 | Código EVENODD para $C_p = 5$ . . . . .                                                  | 31 |
| Figura 12 | Exemplo distribuição de um arquivo. . . . .                                              | 34 |
| Figura 13 | Exemplo espelhamento local. . . . .                                                      | 34 |
| Figura 14 | Exemplo espelhamento remoto. . . . .                                                     | 35 |
| Figura 15 | Exemplo arquitetura tradicional de um <i>cluster</i> . . . . .                           | 37 |
| Figura 16 | Infraestrutura de <i>cloud</i> . . . . .                                                 | 37 |
| Figura 17 | Infraestrutura de Data Center tradicional. . . . .                                       | 38 |
| Figura 18 | Principais componente internos de um disco rígido. . . . .                               | 39 |
| Figura 19 | Esboço da estrutura física de um disco rígido. . . . .                                   | 39 |
| Figura 20 | Estrutura interna de um disco SSD. . . . .                                               | 41 |
| Figura 21 | Comparação entre a estrutura física de um disco HD e disco SSD . . . . .                 | 42 |
| Figura 22 | Sistema de RAID com discos individuais fragmentados em bloco. . . . .                    | 43 |
| Figura 23 | Funcionamento geral do RAID . . . . .                                                    | 43 |
| Figura 24 | RAID 0. . . . .                                                                          | 45 |
| Figura 25 | RAID 1. . . . .                                                                          | 46 |
| Figura 26 | RAID 2. . . . .                                                                          | 47 |
| Figura 27 | Distribuição dos bits de informação da mensagem $u$ . . . . .                            | 47 |
| Figura 28 | Resultado do exemplo sobre RAID 2. . . . .                                               | 48 |
| Figura 29 | RAID 3. . . . .                                                                          | 49 |
| Figura 30 | RAID 4. . . . .                                                                          | 50 |
| Figura 31 | Distribuição de dados no arranjo de disco em RAID 4. . . . .                             | 50 |
| Figura 32 | RAID 5. . . . .                                                                          | 51 |
| Figura 33 | Distribuição dos bits de informação no arranjo de disco configurado em RAID 5. . . . .   | 52 |
| Figura 34 | Distribuição dos bits de informação e paridade no arranjo configurado em RAID 5. . . . . | 52 |
| Figura 35 | Arranjo de discos configurados em RAID 5 com a presença de um disco danificado. . . . .  | 53 |
| Figura 36 | Adição do disco novo ao arranjo. . . . .                                                 | 53 |
| Figura 37 | Resultado da recuperação dos dados em um sistema configurado em RAID 5. . . . .          | 53 |

|           |                                                                               |    |
|-----------|-------------------------------------------------------------------------------|----|
| Figura 38 | RAID 5: Comparação entre a capacidade física dos discos e a capacidade lógica | 54 |
| Figura 39 | RAID 50. . . . .                                                              | 54 |
| Figura 40 | RAID 6 com paridade distribuída. . . . .                                      | 55 |
| Figura 41 | RAID 6 com discos exclusivos para armazenamento de paridade. . . . .          | 56 |
| Figura 42 | Código RDP: distribuição de dados no arranjo de disco em RAID 6. . . . .      | 56 |
| Figura 43 | Código RDP: adição dos bits de paridade $P$ ao sistema em RAID 6. . . . .     | 57 |
| Figura 44 | Arranjo de disco em RAID 6 com paridades geradas via código RDP. . . . .      | 57 |
| Figura 45 | Código EVENODD: distribuição de dados no arranjo de disco em RAID 6 . .       | 58 |
| Figura 46 | Código EVENONDD: demonstração do cálculo da paridade $Q$ . . . . .            | 59 |
| Figura 47 | Arquitetura macro da simulação de um servidor de arquivo. . . . .             | 62 |
| Figura 48 | Demonstração dos arranjos de discos configurados em RAID 5 e 6. . . . .       | 64 |
| Figura 49 | Resultados: taxa de transferência de gravação. . . . .                        | 66 |
| Figura 50 | :Resultados: taxa de transferência de gravação. . . . .                       | 66 |
| Figura 51 | Resultados: IOPS. . . . .                                                     | 67 |
| Figura 52 | Resultados: latência média. . . . .                                           | 68 |

## **LISTA DE TABELAS**

|                                                                                  |    |
|----------------------------------------------------------------------------------|----|
| Tabela 1 – Principais técnicas e códigos dos Níveis de RAID analisados . . . . . | 60 |
| Tabela 2 – Comparação de desempenho entre os Níveis de RAID . . . . .            | 60 |

## **LISTA DE ABREVIATURAS E SIGLAS**

|        |                                                 |
|--------|-------------------------------------------------|
| CD     | Compact Disc                                    |
| CPU    | Central Processing Unit                         |
| SDS    | Software Defined Storage                        |
| DRAM   | Dynamic Random Access Memory                    |
| DSS    | Distributed Storage System                      |
| ECC    | Error Correcting Code                           |
| E/S    | Entrada/saída                                   |
| GB     | Gigabytes                                       |
| Gbit/s | Gigabits per sec.                               |
| GPU    | Graphics Processing Unit)                       |
| HA     | High Availability                               |
| HBA    | Host Bus Adapter                                |
| HD     | Hard Disk                                       |
| HDD    | Hard Disk Drive                                 |
| IBGE   | Instituto Brasileiro de Geografia e Estatística |
| IDC    | International Data Corporation                  |
| I/O    | Input/Output                                    |
| KB     | kilobytes                                       |
| MB/s   | Megabytes por sec.                              |
| MLC    | Multi Level Cell                                |
| ms     | Milissegundo                                    |
| NAND   | Not-And                                         |
| QLC    | Quad Level Cell                                 |
| RAID   | Redundant Array of Independent Disks            |
| RAM    | Random Access Memory                            |

|      |                             |
|------|-----------------------------|
| RPD  | Row-diagonal parity         |
| RPM  | Revolutions per minute      |
| SLC  | Single Level Cell           |
| SLED | Single Large Expensive Disk |
| SDS  | Software Defined Storage    |
| SSD  | Solid State Drive           |
| TB   | Terabytes                   |
| TLC  | Triple Level Cell           |
| USB  | Universal Serial Bus        |
| VM   | Virtual Machine             |
| XOR  | Ou exclusivo                |
| ZB   | Zettabytes                  |

## LISTA DE SÍMBOLOS

|               |                                 |
|---------------|---------------------------------|
| $b$           | Bloco                           |
| $\mathcal{C}$ | Código de bloco lineares        |
| $c$           | Palavra-código                  |
| $C_p$         | Controle de paridade            |
| $d$           | Distância mínima de Hamming     |
| $d_h$         | Distância de Hamming            |
| $F_q$         | Corpo Finito                    |
| $G$           | Matriz geradora                 |
| $H$           | Matriz de paridade              |
| $h$           | Fator de paridade               |
| $I_k$         | Matriz identidade de dimensão k |
| $k$           | Bits de informação              |
| $m$           | Bit de paridade                 |
| $n$           | Comprimento da palavra-código   |
| $P$           | Paridade                        |
| $Q$           | Paridade dupla                  |
| $q^k$         | Quantidade de palavra-código    |
| $R_c$         | Taxa de codificação             |
| $t_d$         | Capacidade de detecção de erro  |
| $t_c$         | Capacidade de correção de erro  |
| $w$           | Peso mínimo de Hamming          |
| $w_h$         | Peso de Hamming                 |
| $u$           | Mensagem a ser codificada       |

## SUMÁRIO

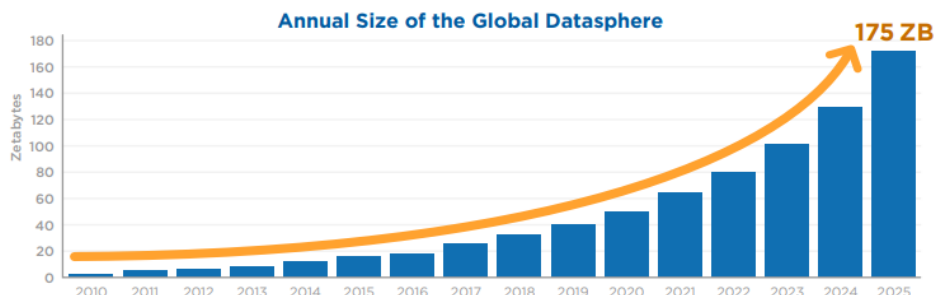
|              |                                                                                 |           |
|--------------|---------------------------------------------------------------------------------|-----------|
| <b>1</b>     | <b>INTRODUÇÃO . . . . .</b>                                                     | <b>15</b> |
| <b>2</b>     | <b>CÓDIGO DE BLOCO . . . . .</b>                                                | <b>17</b> |
| 2.1          | Código de Blocos Lineares . . . . .                                             | 17        |
| <b>2.1.1</b> | <b>Código de Hamming . . . . .</b>                                              | <b>23</b> |
| 2.2          | Códigos Matriciais de Paridade . . . . .                                        | 24        |
| <b>2.2.1</b> | <b>Código Matricial de Paridade Unidimensional . . . . .</b>                    | <b>24</b> |
| <b>2.2.2</b> | <b>Código Matricial de Paridade Bidimensional . . . . .</b>                     | <b>25</b> |
| <b>2.2.3</b> | <b>Código Matricial de Paridade Diagonal . . . . .</b>                          | <b>27</b> |
| <b>2.2.4</b> | <b>Código RDP . . . . .</b>                                                     | <b>27</b> |
| <b>2.2.5</b> | <b>Código EVENODD . . . . .</b>                                                 | <b>30</b> |
| <b>3</b>     | <b>SISTEMAS DE ARMAZENAMENTO DISTRIBUÍDO . . . . .</b>                          | <b>33</b> |
| 3.1          | Princípios dos Sistemas de Armazenamento Distribuído . . . . .                  | 33        |
| <b>3.1.1</b> | <b>Distribuição de Dados . . . . .</b>                                          | <b>33</b> |
| <b>3.1.2</b> | <b>Espelhamento de Dados . . . . .</b>                                          | <b>34</b> |
| <b>3.1.3</b> | <b>Tolerância a Falhas . . . . .</b>                                            | <b>35</b> |
| <b>3.1.4</b> | <b>Disponibilidade, Escalabilidade e Confiabilidade . . . . .</b>               | <b>35</b> |
| 3.2          | Implantações de Sistemas de Armazenamento Distribuídos . . . . .                | 36        |
| 3.3          | Dispositivos de Armazenamento de Dados . . . . .                                | 39        |
| <b>4</b>     | <b>RAID . . . . .</b>                                                           | <b>43</b> |
| 4.1          | Níveis de RAID . . . . .                                                        | 44        |
| <b>4.1.1</b> | <b>RAID 0: Distribuição . . . . .</b>                                           | <b>45</b> |
| <b>4.1.2</b> | <b>RAID 1: Espelhamento . . . . .</b>                                           | <b>45</b> |
| <b>4.1.3</b> | <b>RAID 2: Distribuição e Redundância via Código de Hamming . . . . .</b>       | <b>47</b> |
| <b>4.1.4</b> | <b>RAID 3: Distribuição e Redundância via Paridade Unidimensional . . . . .</b> | <b>48</b> |
| <b>4.1.5</b> | <b>RAID 4: Distribuição e Redundância via Paridade Unidimensional . . . . .</b> | <b>50</b> |
| <b>4.1.6</b> | <b>RAID 5: Distribuição e Paridade . . . . .</b>                                | <b>51</b> |
| <b>4.1.7</b> | <b>RAID 6: Paridade Dupla via Código RDP e Código EVENODD . . . . .</b>         | <b>55</b> |
| <b>5</b>     | <b>ANÁLISE DE DESEMPENHO ENTRE NÍVEIS DE RAID . . . . .</b>                     | <b>60</b> |
| 5.1          | Análise de um ambiente configurado em RAID 5 e RAID 6 . . . . .                 | 61        |
| 5.2          | Resultados da Simulação . . . . .                                               | 66        |
| <b>6</b>     | <b>CONCLUSÃO . . . . .</b>                                                      | <b>69</b> |
|              | <b>REFERÊNCIAS . . . . .</b>                                                    | <b>71</b> |

# 1 INTRODUÇÃO

Nos últimos anos, o avanço tecnológico e a transformação digital têm impulsionado cada vez mais o crescimento do volume de dados processados e armazenados. A tendência de crescimento de dados é reflexo da relevância dos dados para a construção de soluções, análise de comportamento, criação de métricas para tomada de decisão, afeiçoamento de processos e continuidade dos negócios.

De acordo com o relatório "A digitalização do mundo da borda ao núcleo" (*The Digitization of the World From Edge to Core*), a Corporação Internacional de Dados (IDC, *International Data Corporation*) prevê um crescimento global na criação de dados de aproximadamente 175 zettabytes (ZB) até 2025, como representado no gráfico da Figura 1, (REINSEL, David. GANTZ, John. RYDNING, John., 2018).

Figura 1 – Expectativa de crescimento de dados a nível global.



fonte: (REINSEL, David. GANTZ, John. RYDNING, John., 2018).

Nos sistemas de armazenamento e transmissão de dados, é possível ocorrer perdas parciais ou completas da informação armazenada ou transmitida. Essa perda de dados pode ser causada por diversos fatores, como falhas de *hardware*, erros durante a transmissão, entre outros fatores, (KIM, J.; YU, J. H.; KANG, H.; SHIN, J. H.; JEONG, H.; NOH, S. Y. , 2023). As perdas de dados decorrentes de falhas de *hardware* geralmente são causadas por problemas relacionados a unidade de disco rígido (HDD, *hard disk drive*) ou unidade de estado sólido (SSD, *solid state drive*) de um servidor ou sistema de armazenamento de dados. Com o intuito de reduzir danos e evitar possíveis perdas de dados, as empresas buscam investir seus recursos em métodos que ofereçam proteção, prevenção e recuperação de dados. Um desses métodos é conhecido como RAID.

O RAID (*Redundant Array of Independent Disks*) é um método de proteção de dados aplicado em sistemas de armazenamento distribuído. O RAID consiste em criar conjuntos de discos para o armazenamento de dados, visando otimizar a taxa de transferência de leitura e gravação, bem como fornecer alta disponibilidade aos sistemas de armazenamento de dados, (MAZIERO, C. A., 2019). O RAID é dividido em diferentes tipos de conjuntos de discos, conhecido com níveis de RAID. Existem vários níveis de RAID, tais como RAID 0, RAID 1, RAID 2, RAID 3, RAID 4, RAID 5, RAID 6, RAID 50 e RAID 60. Cada um destes níveis apresenta características distintas, permitindo a escolha da configuração mais adequada com base nos requisitos de desempenho, redundância e capacidade de armazenamento para um sistema de armazenamento distribuído, (CHEN, Y. , 2022).

O RAID é criado com base em três técnicas: distribuição de dados, espelhamento de dados e paridade. A paridade é uma técnica essencial em sistemas de armazenamento distribuídos, devido à sua capacidade de oferecer redundância e confiabilidade. Essa técnica é aplicada em sistemas RAID para garantir a integridade dos dados e capacidade de recuperar dados em caso de falhas nos discos. A paridade é implementada nos sistemas de armazenamento de dados por meio de códigos de paridade, proporcionando assim a capacidade de recuperação de dados,(LIMA, A. C., 2021).

O código de paridade é uma técnica de correção de erro aplicada em sistema de armazenamento de dados. Classificados como códigos de bloco, os códigos de paridade são baseados na adição de bits de paridade à mensagem que será transmitida ou armazenada. Eles operam em blocos de dados e utilizam operações XOR para determinar a paridade de um sistema,(RYAN, W. E.; LIN, S., 2009).

Neste trabalho iremos analisar a aplicação de códigos de bloco na construção de níveis de RAID utilizado em sistemas de armazenamento distribuído (DSS, *Distributed Storage System*). Entre os códigos de bloco discutidos, estão o código de Hamming, usado na codificação do RAID 2. Também analisaremos os códigos matriciais e de paridade, presentes no desenvolvimento do RAID 3, 4 e 5. Por fim, examinaremos os códigos de paridade linha-diagonal (RDP, *row-diagonal parity*) e EVENODD, códigos baseados na operação de XOR, utilizados para fornecer paridade dupla. Esses códigos são usados na implementação do RAID 6. Neste trabalho, discutiremos o comportamento e as principais características de cada nível de RAID avaliado. Além disso, exploraremos a diferença de desempenho entre o RAID 5 e 6, a fim de avaliar o desempenho desses níveis de RAID em ambiente de armazenamento de dados configurado com 5 discos e capacidade total lógica de 5 *gigabytes* (GB). Para isso, iremos simular um servidor de arquivos por meio de uma máquina virtual desenvolvida no *software* VirtualBox, uma ferramenta de virtualização usada para simulação de ambientes virtuais para testes.

O presente trabalho de conclusão de curso está estruturado da seguinte maneira: no Capítulo 2 abordaremos os principais conceitos referente aos códigos de blocos, destacando os códigos de Hamming e códigos matriciais de paridade, ambos aplicados nos principais sistemas de RAID existentes. No Capítulo 3, detalharemos os princípios fundamentais dos sistemas de armazenamento distribuídos, abordando especificamente a estrutura de discos HDD e SSD. No Capítulo 4, serão apresentadas as características dos principais níveis de RAID, explorando a relação entre esses níveis de RAID e os códigos correspondentes, utilizando as operações XOR. No Capítulo 5, realizaremos uma análise de desempenho em relação a um servidor de arquivos composto por dois conjuntos de discos, configurados em RAID 5 e 6, no qual serão considerados métricas como velocidade de gravação e leitura, operações de entrada/saída por segundo (IOPS, *input/output operations per second*) e latência. Por fim, no Capítulo 6, apresentaremos as conclusões finais deste trabalho.

## 2 CÓDIGO DE BLOCO

Um código de bloco é uma técnica de codificação que divide os dados em blocos e introduz bits de paridade, também conhecidos como bits de redundância, com o propósito de detectar e/ou corrigir erros de transmissão. Esses códigos de bloco constituem uma classe de códigos de correção de erros fundamentais na teoria da codificação, utilizados para garantir a integridade dos dados em sistemas de comunicação, armazenamento e processamento de informações (ARAUJO, A. C. M.; TEIXEIRA, J. F.; SANTOS, G. T., 2020).

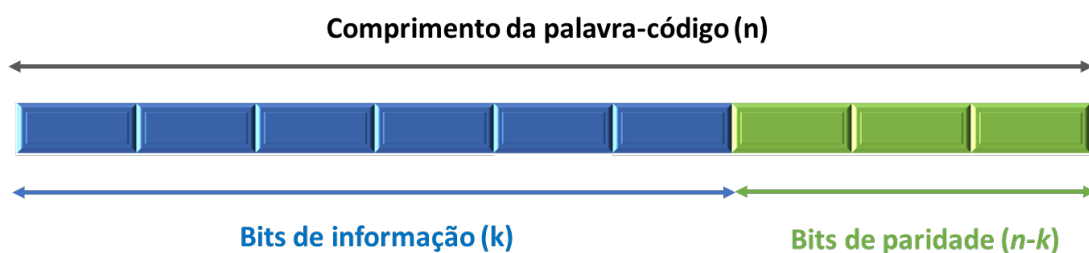
Em um código de blocos, os bits ou símbolos são divididos em blocos compostos por  $k$  bits ou símbolos, a partir dos quais são geradas palavras-código com  $n$  bits ou símbolos. A notação que caracteriza um código de bloco é especificada como  $(n, k)$ , onde  $k$  representa o número de bits que é codificado em um bloco de  $n$  bits. Assim, a quantidade de bits de paridade inseridos no processo de codificação é igual a  $(n - k)$  (CALDEIRA, J.; MATOS, R. A. O., 2023).

Neste capítulo, apresentaremos os conceitos relacionados aos códigos de blocos. Na Seção 2.1, exploraremos os princípios fundamentais dos códigos de blocos lineares, com ênfase especial nos códigos de Hamming. Na Seção 2.2, abordaremos os conceitos sobre paridade por meio dos códigos matriciais. É importante destacar que esses códigos desempenham um papel fundamental na formulação das técnicas de RAID que serão discutidas no Capítulo 4. Para a elaboração deste capítulo, as principais referências utilizadas foram [ (RYAN, W. E.; LIN, S., 2009), (SAINI, M. L.; SHARMA, V. K.; KUMAR, A., 2023), (ARAUJO, A. C. M.; TEIXEIRA, J. F.; SANTOS, G. T., 2020), (CALDEIRA, J.; MATOS, R. A. O., 2023), (XIANG, L.; XU, Y.; LUI, J.C.S.; CHANG, Q., 2021), (HANXU, H.; YUNGHSIANG, H. S.; KENNETH, S. W.; HUI, L., 2018), (PERTIN, D.; KEMPEN, A.; PARREIN, B.; NORMAND, N., 2015) e (RIBEIRO, M. P., 2021)].

### 2.1 CÓDIGO DE BLOCOS LINEARES

Um código de bloco é caracterizado pela adição de bits de paridade à mensagem que será transmitida ou armazenada, visando à detecção e correção de potenciais erros. A Figura 2 mostra a estrutura de um código de bloco em que uma palavra-código, de comprimento  $n$  bits, é formada por  $k$  bits de informação e pela adição de  $(n - k)$  bits de paridade.

Figura 2 – Estrutura básica de código de bloco com redundância.



fonte: Produção do próprio autor.

Os códigos de bloco podem ser categorizados em duas principais classes:

- Códigos de bloco lineares: são um tipo de códigos corretores de erros baseados em uma estrutura linear, na qual a combinação linear dos bits de mensagem e os bits de paridade segue um padrão linear e sistemático, facilitando a detecção e correção de erros.
- Códigos de bloco não lineares: são um tipo de códigos caracterizado pela ausência de uma estrutura linear em sua codificação. Nesse tipo de código, a relação entre os bits de mensagem e os bits de paridade não segue um padrão linear ou sistemático, tornando a codificação e a decodificação mais complexas em comparação com os códigos de bloco lineares.

Nos códigos de bloco lineares, as operações matemáticas realizadas nos bits de informação e nos bits de paridade são operações lineares. Essas operações matemáticas baseiam-se em adições e multiplicações em um corpo finito. Um corpo é uma estrutura que compreende um conjunto de elementos onde operações de adição e multiplicação podem ser realizadas. Quando esse conjunto de elementos é restrito a um número finito, ele é denominado como um corpo finito.

**Definição 1.** Um corpo  $F$  com uma quantidade finita de elementos  $q$  é chamado de corpo finito e será denotado por  $F_q$ .

**Definição 2.** Seja  $F_q$  um corpo finito com  $q$  elementos. Então,  $\mathcal{C}$  é um código de bloco linear, de comprimento  $n$  e dimensão  $k$  sobre  $F_q$ , se  $\mathcal{C}$  for um subespaço vetorial com dimensão  $k$  de  $F_q^n$ .

Como vimos, um código de bloco linear  $\mathcal{C}$  é caracterizado pelos parâmetros  $(n, k)$ . Cada valor  $c \in \mathcal{C}$  é chamado de palavra-código do código de bloco de linear  $\mathcal{C}$ . A quantidade de palavras-código  $c$  em um código  $\mathcal{C}$  sobre  $F_q$  será  $q^k$ . Isso significa que, para cada combinação única de  $k$  bits de informação, existem  $q^k$  palavras-código distintas em  $\mathcal{C}$ . Temos que:

- Para  $q = 2$ : o código é denominado binário;
- Para  $q = 3$ : o código é denominado como código ternário;
- Para  $q = N$ : o código considerado um código  $q$ -ário, onde representa  $N$  inteiro positivo qualquer.

**Exemplo 1.** Seja  $\mathcal{C} = \{000, 001, 010, 011, 100, 101, 110, 111\}$  um código de bloco linear com parâmetros  $(3, 2)$  sobre o corpo finito  $F_2^3$ . A quantidade de palavras-código deste código é igual a  $q^k = 2^3 = 8$ .

**Definição 3.** A taxa de codificação  $R_c$  de um código de bloco linear é dado pela razão entre o número de bits de informação  $k$  e o número de bits da palavra-código  $n$ , ou seja,

$$R_c = \frac{k}{n}. \quad (1)$$

**Exemplo 2.** Seja  $\mathcal{C}$  um código de bloco linear dado pelas palavras-código  $\mathcal{C} = \{000, 0001, 0010, 0011\}$  sobre  $F_2$ . A taxa de codificação será:

$$R_c = \frac{k}{n} = \frac{2}{4} = \frac{1}{2} = 0,5.$$

**Definição 4.** Sejam dois vetores,  $u = (u_1, u_2, \dots, u_n)$  e  $v = (v_1, v_2, \dots, v_n) \in F_q^n$ , a distância de Hamming, denotada por  $d_h(u, v)$ , é calculada com base nas coordenadas em que esses vetores diferem um do outro. A distância de Hamming entre esses vetores é o número de posições nas quais os elementos correspondentes de  $u$  e  $v$  têm valores diferentes.

**Propriedade 1.** Para os vetores  $u, v$  e  $w \in F_q^n$ , a distância de Hamming possui seguintes propriedades:

1.  $d_h(u, v) \geq 0 \forall u, v \in F_q^n$ ,
2.  $d_h(u, v) = 0 \Leftrightarrow u = v$ ,
3.  $d_h(u, v) = d_h(v, u)$ ,
4.  $d_h(u, v) \leq d_h(u, w) + d_h(w, v)$ .

**Exemplo 3.** Para os vetores  $u = (001)$ ;  $v = (010)$ ;  $w = (110)$  em  $F_2^3$ , a distância de Hamming entre os vetores será:

$$d_h(u, v) = d_h(001, 010) = 2,$$

$$d_h(u, w) = d_h(001, 110) = 3,$$

$$d_h(w, v) = d_h(110, 010) = 1.$$

**Definição 5.** Seja  $C$  um código de bloco linear. A menor distância de Hamming entre todos os pares possíveis em um conjunto de palavras é chamada de distância mínima de Hamming. A distância mínima de Hamming, denotada por  $d(C)$ , é definida como:

$$d(C) = \min\{d_h(u, v)\}, \quad (2)$$

sendo  $u, v \in C$  e  $u \neq v$ .

**Exemplo 4.** Para um código de bloco linear  $C = \{000, 110, 100\}$  sobre o corpo finito  $F_2^3$ , a distância de Hamming será:

$$d_h(000, 110) = 2,$$

$$d_h(000, 100) = 1,$$

$$d_h(110, 100) = 1.$$

Logo, a distância mínima de Hamming é  $d(C) = \min\{2, 1\} = 1$ .

**Definição 6.** O peso de Hamming  $w_H(u)$  de um vetor  $u = (u_1, u_2, \dots, u_n)$  em  $F_q^n$ , é definido como a quantidade de componentes não-nulos contido neste vetor. E o peso mínimo de Hamming  $w(C)$  de um código  $C$  é aquele com o menor peso das palavras-código.

**Teorema 1.** Seja  $C$  um código de bloco linear e  $u, v \in C$ . Nesse contexto, podemos afirmar que:

1.  $d_H(u, v) = w_H(u - v)$ ;
2.  $d(C) = w(C)$ .

**Exemplo 5.** Considerando os vetores  $u = (000)$ ,  $v = (110)$  e  $w = (111)$  sobre o corpo finito  $F_2^3$ , a distância de Hamming entre os vetores será:

$$d_h(000, 110) = 2,$$

$$d_h(000, 111) = 3,$$

$$d_h(110, 111) = 1.$$

Logo, a distância mínima de Hamming é  $d(C) = 1$ . Ao calcular o peso de Hamming, obtemos:

$$d_H(u, v) = w_H(u - v) \rightarrow w_H(110) = 2,$$

$$d_H(u, w) = w_H(u - v) \rightarrow w_H(111) = 3,$$

$$d_H(v, w) = w_H(v - v) \rightarrow w_H(001) = 1.$$

Portanto, o peso mínimo de Hamming é igual a  $w(C) = 1$ .

Considerando um código de bloco linear  $C$  com parâmetros  $(n, k)$  e distância mínima de Hamming  $d$ , podemos representar este código pelos parâmetros  $(n, k, d)$ .

**Teorema 2.** Um código de bloco linear  $C$  com parâmetros  $(n, k, d)$ , possui a capacidade de detecção de até  $t_d = (d - 1)$  erros e a capacidade de correção de até  $t_c = \lfloor \frac{d-1}{2} \rfloor$  erros.

**Exemplo 6.** Seja  $C$  um código de bloco linear com parâmetros  $(3, 2, 3)$ . A capacidade desse código de detectar e corrigir erros será:

$$t_d = (d - 1) = (3 - 1) = 2;$$

$$t_c = \frac{d - 1}{2} = \frac{3 - 1}{2} = 1.$$

Seja  $\mathcal{C}$  um código de bloco linear com parâmetros  $(n, k, d)$ . Como  $\mathcal{C} \subset F_q^n$ , podemos identificar  $k$  vetores linearmente independentes que geram  $\mathcal{C}$ . Esses  $k$  vetores podem ser organizados como as linhas de uma matriz  $k \times n$ , conhecida como a matriz geradora para código  $\mathcal{C}$ .

**Definição 7.** A matriz geradora  $G$  de um código de bloco linear  $\mathcal{C}$  é definida a partir dos vetores  $g_i = g_{i,1}, \dots, g_{i,n}$ , onde  $i = 1, \dots, k$ , ou seja,

$$G = \begin{pmatrix} g_0 \\ g_1 \\ \vdots \\ g_{k-1} \end{pmatrix} = \begin{pmatrix} g_{0,0} & g_{0,1} & \cdots & g_{0,n-1} \\ g_{1,0} & g_{1,1} & \cdots & g_{1,n-1} \\ \vdots & \vdots & \ddots & \vdots \\ g_{k-1,0} & g_{k-1,1} & \cdots & g_{k-1,n-1} \end{pmatrix}.$$

**Exemplo 7.** Seja  $\mathcal{C} = \{0000, 0001, 1001, 1101\}$  um código de bloco. A matriz geradora  $G$  pode ser representada da seguinte forma:

$$G = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 \end{pmatrix},$$

onde a matriz  $G$  contém os vetores que geram o código de bloco  $\mathcal{C}$ . Cada linha da matriz representa um vetor do código de bloco linear  $\mathcal{C}$ .

**Definição 8.** Seja a mensagem a ser codificada, representada por  $u = (u_0, u_1, \dots, u_{k-1})$ . A palavra-código correspondente é obtida da seguinte forma:

$$c = u \cdot G = (u_0, u_1, \dots, u_{k-1}) \cdot \begin{pmatrix} g_0 \\ g_1 \\ \vdots \\ g_{k-1} \end{pmatrix}, \quad (3)$$

logo, temos que:

$$c = u_0 g_0 + u_1 g_1 + \cdots + u_{k-1} g_{k-1}. \quad (4)$$

**Exemplo 8.** Seja  $\mathcal{C}$  um código de bloco linear com parâmetros  $(7, 4)$  sobre  $F_q^n$  e a mensagem  $u = (1000)$ . A matriz geradora  $G$  será:

$$G = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}.$$

Para obter a palavra-código, calculamos o produto entre a mensagem e a matriz geradora  $G$ , como demonstrado a seguir:

$$c = (1000) \cdot \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} = (1000001),$$

**Definição 9.** Uma matriz geradora  $G$  de um código de bloco  $\mathcal{C}$  na forma sistemática é projetada para simplificar a transmissão e recuperação de informações. Ela é definida da seguinte maneira:

$$G = (I_k | P), \quad (5)$$

onde  $I_k$  é a matriz identidade de ordem  $k$  e  $P$  é uma matriz de ordem  $k \times (n - k)$ .

**Definição 10.** Seja  $\mathcal{C}$  um código de parâmetros  $(n, k)$ . Se  $G = (I_k | P)$  é a matriz geradora de  $\mathcal{C}$  na forma sistemática, então a matriz controle de paridade de  $\mathcal{C}$  é:

$$H = (-P^T | I_{n-k}), \quad (6)$$

em que  $P^T$  é a matriz transposta de  $P$  de ordem  $(n - k) \times k$ .

Esta matriz controle de paridade  $H$  é utilizada para verificar se uma mensagem pertence ou não ao código. Para isso, a operação de multiplicação entre as matrizes  $G$  e  $H^T$ , deve ser igual a zero, ou seja,

$$G \cdot H^T = (I_k | P) \cdot (-P^T | I_{n-k}) = 0, \quad (7)$$

se  $v \in \mathcal{C}$ , então,

$$v \cdot H^t = 0 \quad (8)$$

**Exemplo 9.** Seja  $\mathcal{C}$  um código bloco linear com parâmetros  $(5, 3)$  e a matriz geradora  $G$  dada por:

$$G = \begin{pmatrix} 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix}.$$

A partir da matriz geradora na forma sistemática, a matriz  $H$  será dada por:

$$H = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 \end{pmatrix}$$

Para verificar se a mensagem  $v_1 = (01100)$  é uma palavra código de  $\mathcal{C}$ . Multiplicando  $v_1$  por  $H^T$ , ou seja,

$$v_1 \cdot H^t = (01100) \cdot \begin{pmatrix} 1 & 1 \\ 0 & 1 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{pmatrix} = (01),$$

logo,  $v_1 \notin \mathcal{C}$ .

Para verificar se a mensagem  $v_2 = (00100)$  é uma palavra código de  $\mathcal{C}$ . Multiplicando  $v_2$  por  $H^T$ ,

$$v_2 \cdot H^T = (00100) \cdot \begin{pmatrix} 1 & 1 \\ 0 & 1 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{pmatrix} = (00),$$

Portanto, a palavra-código  $v_2 \in \mathcal{C}$ .

### 2.1.1 Código de Hamming

O código de Hamming é um exemplo de código de bloco linear muito utilizado em diversas aplicações. Os parâmetros  $(n, k, 3)$  do código de Hamming são determinados por:

- Comprimento da palavra-código:  $n = 2^m - 1$ ;
- Número de bits de informação:  $k = 2^m - m - 1$ ;
- Distância mínima de Hamming:  $d = 3$ .

Os códigos de Hamming são códigos perfeitos, isto é, alcançam a taxa mais alta para os códigos com o seu comprimento de bloco e uma distância mínima de 3. Um código de Hamming muito utilizado é o  $(7, 4)$  que iremos apresentar no exemplo a seguir.

**Exemplo 10.** O código de Hamming  $(7, 4)$  codifica 4 bits em 7 bits através da adição de 3 bits de paridade. Considerando um codificador sistemático, se  $u = (u_1, u_2, u_3, u_4)$  é uma mensagem a ser codificada por este código então uma palavra-código será dada por

$$c = (u_1, u_2, u_3, u_4, m_1, m_2, m_3), \quad (9)$$

em que  $m_1, m_2, m_3$  serão os bits de paridade, que podem ser calculados de acordo com as seguintes relações:

- $m_1 = u_1 + u_2 + u_3$ ;
- $m_2 = u_1 + u_2 + u_4$  e
- $m_3 = u_2 + u_3 + u_4$ .

Assim, para determinar a palavra-código correspondente à mensagem  $u = (0011)$  usando o código de Hamming  $(7, 4)$ , calculamos

$$m_1 = 0 + 0 + 1 = 1; \quad m_2 = 0 + 0 + 1 = 1 \quad e \quad m_3 = 0 + 1 + 1 = 0. \quad (10)$$

Portanto, a palavra-código é  $n = (0011110)$ . Uma matriz geradora  $G$  para o código de Hamming  $(7, 4)$  acima apresentado é:

$$G = \left( I_k | P \right) = \begin{pmatrix} 1 & 0 & 0 & 0 & \vdots & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & \vdots & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & \vdots & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & \vdots & 0 & 1 & 1 \end{pmatrix}.$$

## 2.2 CÓDIGOS MATRICIAIS DE PARIDADE

Os códigos matriciais são um tipo de códigos de correção de erros que usam matrizes para codificar e decodificar informações. Devido à sua simplicidade e versatilidade, os códigos matriciais são capazes de lidar com a correção de erros aleatórios, bem como a correção de erros em rajadas unidimensionais e até mesmo a correção de erros em duas dimensões.

Um código de paridade é um tipo de código de detecção e correção de erros baseado na inclusão de um bit adicional, conhecido como bit de paridade, o qual pode ser alocado no início ou no final de uma sequência de bits que está sendo transmitida ou armazenada. O bit de paridade é utilizado para a verificação da integridade dos dados e na detecção de erros, contribuindo para a confiabilidade de sistemas de comunicação e armazenamento de dados. Assim, a principal função do bit de paridade é identificar possíveis erros ou até mesmo corrigir o apagamento simples em palavras transmitidas ou armazenadas.

O bit de paridade é calculado com base nos dados armazenados ou transmitidos e é representado pelos valores 0 ou 1, por meio de operações de soma utilizando a operação ou-exclusivo (XOR, *exclusive or*), denotada por  $\oplus$ .

Esse método de paridade é usado para detecção de erros, sendo classificado em dois tipos principais: paridade par e paridade ímpar.

- **Paridade par:**

- Se a quantidade de bits "1" na sequência for par  $\rightarrow$  bit de paridade 0.
- Se a quantidade de bits "1" na sequência for ímpar  $\rightarrow$  bit de paridade 1.

- **Paridade ímpar:**

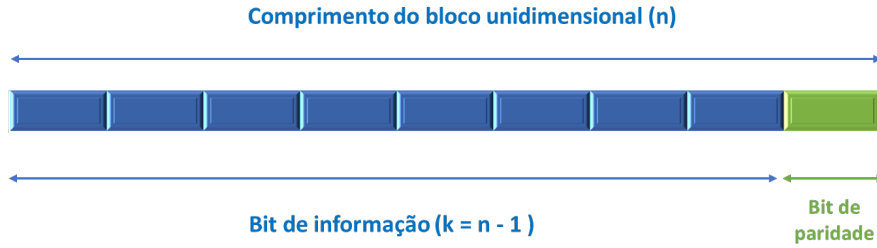
- Se a quantidade de bits "1" na sequência for par  $\rightarrow$  bit de paridade 1.
- Se a quantidade de bits "1" na sequência for ímpar  $\rightarrow$  bit de paridade 0.

Nas subseções a seguir iremos apresentar códigos matriciais unidimensionais e bidimensionais com cálculos de paridade nas linhas, colunas ou diagonais.

### 2.2.1 Código Matricial de Paridade Unidimensional

Um código matricial de paridade unidimensional é um método de detecção de erros, no qual um bit de paridade é acrescentado em um único bloco de informação. A Figura 3 apresenta um código de erro com um único bit de paridade.

Figura 3 – Código de paridade unidimensional.



fonte: Produção do próprio autor.

**Definição 11.** Um código matricial de paridade unidimensional, contendo um único bit de paridade, pode ser definido como um código de bloco linear  $\mathcal{C}$ , com parâmetros  $(n, k, d) = (n, n - 1, 2) = (k + 1, k, 2)$ , onde  $n$  representa o comprimento total do bloco de código,  $k$  denota a quantidade de bits de informação e o valor 2 é a distância mínima de Hamming.

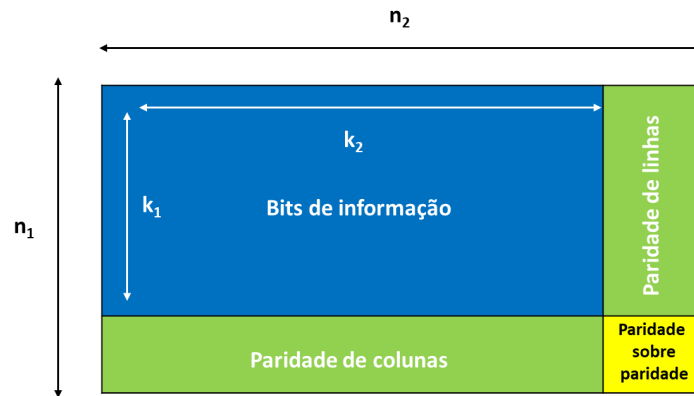
Seja  $u = (u_0, u_1, \dots, u_{n-2})$  a mensagem a ser codificada. Para criar uma palavra-código, um bit de paridade, denotado como  $m$ , é adicionado à mensagem, resultando em  $c = (u = u_0, u_1, \dots, u_{n-2}, m)$ . O bit de paridade  $m$  é calculado por:

$$m = (u_0 \oplus u_1 \oplus \dots \oplus u_{n-2}). \quad (11)$$

### 2.2.2 Código Matricial de Paridade Bidimensional

Diferentemente dos códigos unidimensionais, nos quais os dados são organizados em uma única sequência linear, os códigos bidimensionais dispõem os dados em formato matricial. Um código matricial de paridade bidimensional é um tipo de código de correção de erros composto por bits de dados e paridade que são distribuídos nas linhas e colunas de uma matriz.

Figura 4 – Código matricial de paridade bidimensional.



fonte: Produção do próprio autor.

**Definição 12.** Um código matricial de paridade bidimensional é caracterizado por apresentar paridades nas linhas e nas colunas. Esse tipo de código é definido pelos seguintes parâmetros:

$$[n_1 n_2, (n_1 - 1)(n_2 - 1), 4] = [(k_1 + 1)(k_2 + 1), k_1 k_2, 4], \quad (12)$$

onde 4 é a distância mínima de Hamming.

O bit de paridade das linhas é representado por  $m_i$ , enquanto o bit de paridade das colunas é representado por  $m_j$ . O cálculo desses bits de paridade,  $m_i$  e  $m_j$ , pode ser realizado por meio das seguintes expressões:

$$m_i = (u_{i,0} \oplus u_{i,1} \oplus \cdots \oplus u_{i,n-2}), \quad (13)$$

$$m_j = (u_{j,0} \oplus u_{j,1} \oplus \cdots \oplus u_{j,n-2}). \quad (14)$$

Os códigos de paridade bidimensionais também podem ser representados por meio da notação  $A(n_1, n_2)$ , onde  $n_1$  e  $n_2$  denotam as dimensões da matriz de dados. Essa notação indica o tamanho da matriz na qual os dados são organizados, possibilitando a adição de bits de paridade nas linhas e nas colunas para fins de detecção e correção de erros.

**Exemplo 11.** Seja  $u = (1011)$  a mensagem a ser codificada. Para criar uma matriz  $A(3, 3)$ , é necessário preencher a matriz com os bits de informações da mensagem  $u = (1011)$ , onde a primeira linha da matriz devem ser adicionados os dois primeiros bits da mensagem  $u$  e na segunda linha da matriz são inseridos os outros dois bits da mensagem. Para calcular a paridade entre as linhas da matriz, deve-se utilizar a Equação 13. Assim, temos que:

$$m_i = 1 \oplus 0 = 1,$$

$$m_i = 1 \oplus 1 = 0.$$

A paridade entre as colunas da matriz, pode ser calculada utilizando a Equação 14.

$$m_j = 1 \oplus 1 = 0,$$

$$m_j = 0 \oplus 1 = 1.$$

Para calcular a paridade sobre a paridade, considerando a paridade das linhas ou paridade das colunas.

$$m_i = 0 \oplus 1 = 1,$$

$$m_j = 1 \oplus 0 = 1.$$

Por fim, tem-se um código matricial de paridade diagonal  $A(3, 3)$ , como mostrado na Figura 5.

Figura 5 – Código matricial de paridade diagonal  $A(3, 3)$ .

|   |   |   |
|---|---|---|
| 1 | 0 | 1 |
| 1 | 1 | 0 |
| 0 | 1 | 1 |

fonte: Produção do próprio autor.

### 2.2.3 Código Matricial de Paridade Diagonal

Um código matricial de paridade diagonal é um tipo de código de correção de erros, no qual os bits de paridade são incluídos nas linhas e colunas. Um código matricial de paridade diagonal é representado como  $A(n, n)$ . Esse tipo de código matricial de paridade diagonal pode ser obtido a partir de um código matricial bidimensional nas colunas, representado como  $A(n, n - 1)$ , com a inclusão de bits de paridade na última coluna da matriz. Esses bits de paridade são calculados através da operação de XOR. Em um arranjo de disco, a paridade é calculada por:

$$P = D_1 \oplus D_2 \oplus D_3 \oplus \dots \oplus D_{n-1}. \quad (15)$$

**Exemplo 12.** Considere  $u = (ABCDEFGH IJKLMN OP)$  uma mensagem a ser utilizada em um código matricial com  $A(4, 3)$ . Para transformar essa mensagem em um código matricial com paridades nas diagonais, conforme Figura 6, é necessário realizar a operação do XOR entre os elementos da matriz. Desse modo, as paridades diagonais ( $D, G, J$  e  $M$ ) são obtidas da seguinte forma:

$$\begin{aligned} D &= (A \oplus B \oplus C), \\ G &= (E \oplus F \oplus H), \\ J &= (I \oplus K \oplus L), \\ M &= (N \oplus O \oplus P). \end{aligned}$$

Figura 6 – Exemplo de um código matricial com paridade diagonal  $A(4, 4)$ .

|   |   |   |   |
|---|---|---|---|
| A | B | C | D |
| E | F | G | H |
| I | J | K | L |
| M | N | O | P |

fonte: Produção do próprio autor.

### 2.2.4 Código RDP

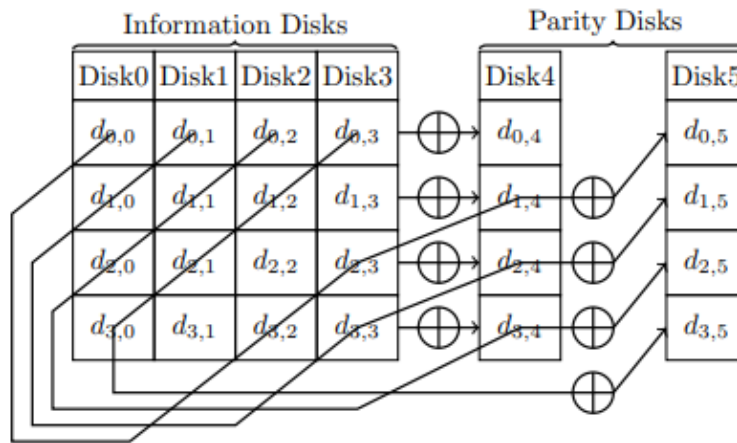
O código de paridade linha-diagonal (RDP, *row-diagonal parity*) refere-se a um código de paridade dupla aplicado em sistemas de armazenamento de dados, como na arquitetura de RAID. Criado em 2004, o código RDP visa aumentar a confiabilidade dos dados, proporcionando redundância de dados e tolerância a falhas em um conjunto de discos, (HANXU, H.; YUNGHSIANG, H. S.; KENNETH, S.

W.; HUI, L., 2018). Também conhecido como RAID de paridade dupla (RAID-DP, *Dual Parity*, o código RDP é considerado um dos principais códigos utilizado em RAID 6 devido sua otimização em eficiência computacional e entrada/saída (E/S), (XIANG, L.; XU, Y.; LUI, J.C.S.; CHANG, Q., 2021)

O código RDP é uma categoria de código matricial, no qual as primeiras colunas da matriz representam os bits de informação, enquanto as duas últimas são designadas para armazenar a paridade. Essas paridades são identificadas como paridade  $P$  e paridade  $Q$ . Por meio dessas paridades, o sistema assegura a recuperação completa das informações mesmo diante de até duas falhas de disco simultâneas.

A codificação de um código RDP é baseada no uso de uma matriz bidimensional com dimensões  $(C_p - 1) \times (C_p + 1)$ , onde o parâmetro de controle  $C_p$  é um número primo maior que 2. A Figura 7 ilustra a arquitetura genérica do código RDP.

Figura 7 – Código RDP para  $C_p = 5$ .



fonte: (XIANG, L.; XU, Y.; LUI, J.C.S.; CHANG, Q., 2021).

**Definição 13.** Um código RDP é um código corretor de erros com parâmetros  $(C_p, n, b)$ , em que  $C_p$  é o controle de paridade, utilizado para determinar a dimensão da matriz,  $n$  representa a quantidade de colunas e  $b$  identifica a quantidade de linhas da matriz. A quantidade de linhas e colunas pode ser calculada utilizando as seguintes expressões:

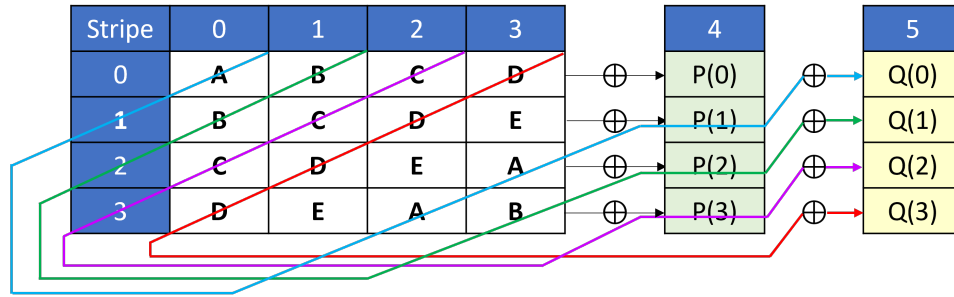
$$b = (C_p - 1); \quad (16)$$

$$n = (C_p + 1). \quad (17)$$

Nos códigos RDP, o cálculo da paridade é efetuado exclusivamente por meio de operações XOR. Essas operações XOR permitem que o código RDP efetue a combinação e manipulação dos bits de informação para gerar os bits de paridade.

**Exemplo 13.** Considere um código RDP com controle de paridade  $C_p = 5$ . Suponha que a mensagem  $u = (ABCDBCDECADEAB)$  seja armazenada nesta matriz, conforme mostrado na Figura 8.

Figura 8 – Demonstração do cálculo da paridade  $P$  e  $Q$  via código RDP.



O calcular a paridade  $P$  é aplicar a operação XOR entre as linhas da matriz, similar ao processo realizado no Exemplo 12. Desse modo, a paridade  $P(0)$ ,  $P(1)$ ,  $P(2)$  e  $P(3)$  será:

$$P(0) = A \oplus B \oplus C \oplus D,$$

$$P(1) = B \oplus C \oplus D \oplus E,$$

$$P(2) = C \oplus D \oplus E \oplus A,$$

$$P(3) = D \oplus E \oplus A \oplus B.$$

Para calcular a paridade  $Q$  é aplicado a técnica de paridade diagonal utilizando a operação XOR entre as colunas dos dados e a paridade  $P$ . Para o cálculo de cada bloco da paridade  $Q$ , linhas coloridas foram traçadas, como exemplificado na Figura 8.

- Para calcular  $Q(0)$ , considere a linha azul. Assim,  $Q(0)$  é dado por:

$$Q(0) = A \oplus A \oplus A \oplus P(1).$$

- Para determinar a paridade  $Q(1)$ , deve seguir a linha verde. Dessa forma  $Q(1)$  será:

$$Q(1) = B \oplus B \oplus B \oplus P(2).$$

- A linha roxa determinar o cálculo da paridade  $Q(2)$ :

$$Q(2) = C \oplus C \oplus C \oplus P(3).$$

- A paridade  $Q(3)$  é representada pela diagonal principal, destacada pela linha em vermelho. Desse modo, a paridade do  $Q(3)$  será:

$$Q(3) = D \oplus D \oplus D \oplus D.$$

### 2.2.5 Código EVENODD

O código EVENODD é um tipo de código matricial desenvolvido para aplicações relacionadas aos setores de sistemas de armazenamento. O código de EVENODD, surgiu em 1995 para atender à crescente demanda do mercado por sistemas de armazenamento com maior taxa de confiabilidade. Esse código proporciona uma camada extra de redundância e confiabilidade ao permitir a tolerância de até duas falhas de disco (XIANG, L.; XU, Y.; LUI, J.C.S.; CHANG, Q., 2021).

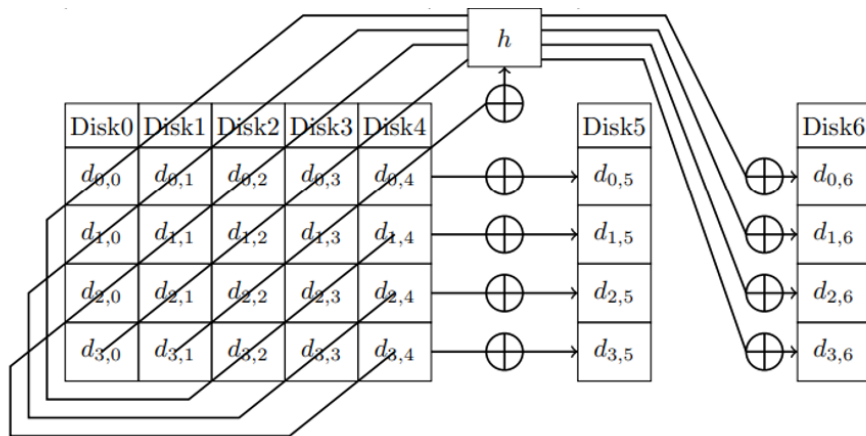
Devido à sua capacidade de lidar com até duas falhas, o código EVENODD é uma técnica eficaz na correção de erros, sendo amplamente utilizado na implementação do RAID 6, conforme será demonstrado em detalhes na Subseção 4.1.7.

O código EVENODD, similar ao código RDP, utiliza a operação XOR para calcular a paridade  $P$  de um sistema de armazenamento de dados distribuído. No entanto, ao contrário do código RDP, o código EVENODD utiliza um fator de paridade  $h$  dado por:

$$h = \bigoplus_{j=1}^{C_p-1} d_{p-1-j,j}. \quad (18)$$

A Figura 9 apresenta a arquitetura geral de um código EVENODD, considerando controle de paridade igual a  $C_p = 5$ .

Figura 9 – Arquitetura geral do código EVENODD.



fonte: (XIANG, L.; XU, Y.; LUI, J.C.S.; CHANG, Q., 2021).

**Definição 14.** Um código EVENODD possui  $(C_p + 2)$  colunas e  $(C_p - 1)$  linhas. Para determinar a paridade  $Q$  é necessário um valor intermediário  $h$ .

**Exemplo 14.** Seja  $u = (ABCDEBCDEACDEABDEABC)$  uma mensagem a ser codificada utilizando o método do código EVENODD com  $C_p = 5$ . As paridades geradas por esses símbolos de informação podem ser calculadas de acordo com os seguintes passos:

**Passo 1:** Distribuir a mensagem  $u$  entre os blocos dos discos dedicados para símbolos de informação, conforme ilustrado na Figura 10.

Figura 10 – Distribuição mensagem  $u$  no arranjo de disco para matriz com  $C_p = 5$ .

| Stripe | 0 | 1 | 2 | 3 | 4 | 5    | 6    |
|--------|---|---|---|---|---|------|------|
| 0      | A | B | C | D | E | P(0) | Q(0) |
| 1      | B | C | D | E | A | P(1) | Q(1) |
| 2      | C | D | E | A | B | P(2) | Q(2) |
| 3      | D | E | A | B | C | P(3) | Q(3) |

fonte: Produção do próprio autor.

**Passo 2:** Para calcular a paridade  $P$ , representada pelos bloco  $P(0)$ ,  $P(1)$ ,  $P(2)$  e  $P(3)$ , é necessário aplicar a técnica de paridade bidimensional por linha. Desse modo, os blocos da paridade  $P$  resulta em:

$$P(0) = A \oplus B \oplus C \oplus D \oplus E,$$

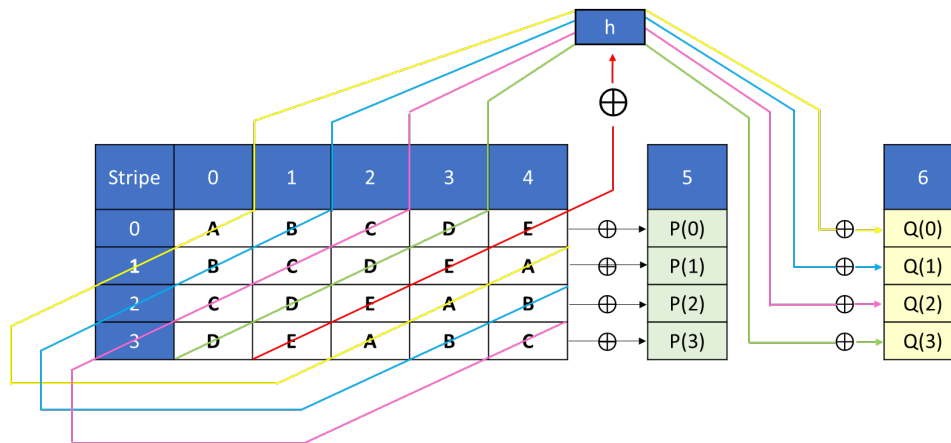
$$P(1) = B \oplus C \oplus D \oplus E \oplus A,$$

$$P(2) = C \oplus D \oplus E \oplus A \oplus B,$$

$$P(3) = D \oplus E \oplus A \oplus B \oplus C.$$

**Passo 3:** Para calcular a paridade  $Q$ , representada pelos bloco  $Q(0)$ ,  $Q(1)$ ,  $Q(2)$  e  $Q(3)$ , é necessário aplicar a técnica de paridade diagonal e acrescentar o fator de paridade  $h$ , conforme apresenta a Figura 11.

Figura 11 – Código EVENODD para  $C_p = 5$ .



fonte: Produção do próprio autor.

O fator de paridade  $h$ , representado pela linha vermelha, pode ser calculado da seguinte forma:

$$h = E \oplus E \oplus E \oplus E.$$

Para o cálculo de cada bloco da paridade  $Q$  deve seguir as linhas coloridas como exemplificado na Figura 10. Desse modo, obtemos:

- A linha amarela representa a  $Q(0)$ . Assim,  $Q(0)$  é dado por:

$$Q(0) = A \oplus A \oplus A \oplus A \oplus h.$$

- A linha azul identifica a paridade  $Q(1)$ . Dessa forma  $Q(1)$  resulta em:

$$Q(1) = B \oplus B \oplus B \oplus B \oplus h.$$

- A linha roxa determinar a paridade  $Q(2)$ , ou seja,

$$Q(2) = C \oplus C \oplus C \oplus C \oplus h.$$

- O cálculo para paridade  $Q(3)$  é representado pela linha verde. Desse modo, a paridade do  $Q(3)$  será:

$$Q(3) = D \oplus D \oplus D \oplus D \oplus h.$$

### 3 SISTEMAS DE ARMAZENAMENTO DISTRIBUÍDO

Os sistemas de armazenamento distribuídos (DSS, *distributed storage systems*) surgiram em resposta aos avanços tecnológicos que apresentaram à necessidade de armazenar e processar um volume cada vez maior de dados de maneira confiável, disponível, escalável e com tolerância a falhas, (KIM, J.; YU, J. H.; KANG, H.; SHIN, J. H.; JEONG, H.; NOH, S. Y. , 2023).

Um sistema de armazenamento distribuído refere-se a uma infraestrutura de armazenamento em que os dados são distribuídos entre vários dispositivos interconectados, ao invés de estarem concentrados exclusivamente em um único dispositivo central, ( SHVIDKIY, A. A.; SVELIEVA, A. A.; ZARUBIN, A. A., 2021). Desse modo, mesmo que os dados estejam distribuídos fisicamente em dispositivos distintos, os usuários finais têm uma percepção unificada do sistema como uma única solução lógica.

Os sistemas de armazenamento distribuídos visam aprimorar o desempenho, oferecer redundância ao sistema, promover a escalabilidade e proporcionar maior disponibilidade dos dados. Por meio desses sistemas, os dados podem ser replicados ou fragmentados em diversos canais, incluindo: sistemas de computadores, plataformas em nuvem e unidades de armazenamento individuais, (TANENBAUM, A. S.; STEVEN, M. V., 2007).

#### 3.1 PRINCÍPIOS DOS SISTEMAS DE ARMAZENAMENTO DISTRIBUÍDO

Esta seção aborda os princípios fundamentais relacionados aos sistemas de armazenamento distribuídos. Esses princípios são essenciais para compreender as discussões apresentadas ao longo deste trabalho, visando facilitar a compreensão de como os dados são organizados, compartilhados e gerenciados em um ambiente distribuído.

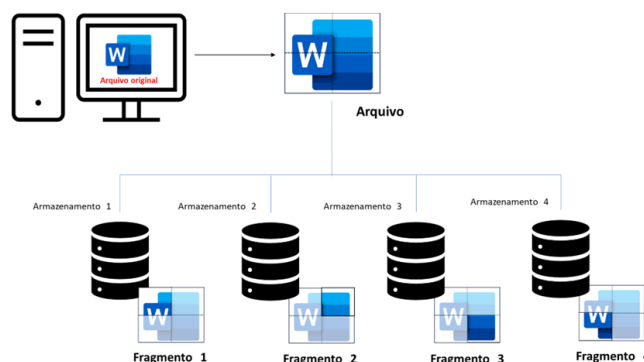
##### 3.1.1 Distribuição de Dados

A distribuição de dados é uma técnica que envolve a divisão dos dados em partes menores, chamadas de fragmentos. Esses fragmentos de dados são então alocados em vários dispositivos individuais dentro de uma rede. Normalmente, esses dispositivos são representados por servidores, computadores, dispositivos de armazenamento e unidades de armazenamento individuais, ( SHVIDKIY, A. A.; SVELIEVA, A. A.; ZARUBIN, A. A., 2021).

A distribuição de dados permite a otimização dos recursos disponíveis, pois os dados podem ser distribuídos de forma eficiente entre os dispositivos da rede. O principal objetivo da distribuição de dados é aprimorar o desempenho e a capacidade de armazenamento. A distribuição de dados permite que os dados sejam acessados com mais eficiência, visto que vários dispositivos estão trabalhando em conjunto na leitura e gravação de dados. Essa técnica resulta em tempos de acesso mais rápidos e na utilização mais eficiente dos recursos disponíveis, (MAZIERO, C. A., 2019).

A Figura 12 ilustra um exemplo de distribuição de dados, onde um arquivo de texto, no formato *Word*, é fragmentado e armazenado em quatro dispositivos de armazenamento individuais, onde cada dispositivo pode ser dividido em blocos.

Figura 12 – Exemplo distribuição de um arquivo.



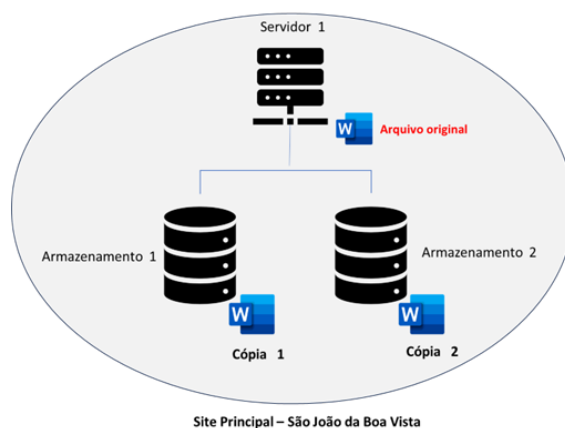
fonte: Produção do próprio autor.

### 3.1.2 Espelhamento de Dados

O espelhamento, também conhecido como replicação, é uma técnica utilizada para criar uma cópia de dados em outro local. Nesse método, o sistema de armazenamento de dados é projetado para replicar as informações gravadas em um dispositivo para outro dispositivo, (THOMASIAN, A., 2018). Um dos principais objetivos da utilização do espelhamento de dados é aumentar a disponibilidade do sistema, pois as cópias dos dados podem ser usadas para procedimentos de recuperação em caso de falhas de *hardware*.

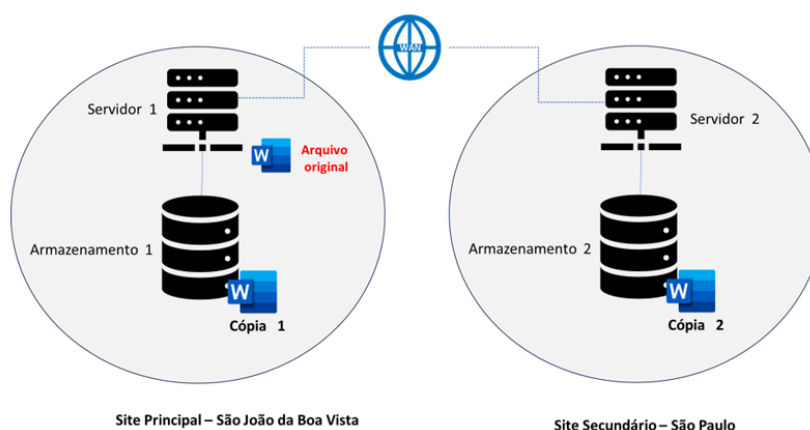
Segundo (SOMASUNDARAM G.; SHRIVASTAVA, A.; FERNANDES, A. P. , 2011), o espelhamento de dados pode ser dividido em duas categorias distintas, sendo elas: espelhamento local e espelhamento remoto. No contexto do espelhamento local, as cópias replicadas dos dados são gerenciadas e mantidas dentro do mesmo ambiente físico. E o espelhamento remoto envolve a criação de cópias de dados em locais geograficamente separados. Muitas vezes, essas réplicas são armazenadas em um centro de dados (*Data Center*) secundário, visando proporcionar maior disponibilidade e a resiliência dos dados. As Figura 13 e Figura 14, apresentam um exemplo de espelhamento local e remoto.

Figura 13 – Exemplo espelhamento local.



fonte: Produção do próprio autor.

Figura 14 – Exemplo espelhamento remoto.



fonte: Produção do próprio autor.

### 3.1.3 Tolerância a Falhas

Segundo (SILBERSCHATZ, A., 2015), um sistema com tolerância a falhas é aquele que possui uma infraestrutura projetada para continuar operando sem interrupções, mesmo que alguns de seus componentes apresentem defeitos durante a execução das atividades. Em outras palavras, se ocorrer uma falha, como a de um disco em um servidor, o sistema é capaz de continuar funcionando, assegurando a disponibilidade dos serviços e evitando impactos negativos nos usuários e nas operações da empresa.

Os sistemas programados para lidar com falhas podem ser classificados como sistemas redundantes. Os sistemas redundantes são aqueles projetados com dois ou mais componentes essenciais em sua estrutura. Em sistema de armazenamento com redundância, os discos devem ser configurados em um arranjo RAID, conceito abordado no Capítulo 4.

O conceito de redundância também pode ser aplicado em ambientes mais amplos, como infraestruturas de *Data Center*, visando diminuir os impactos de possíveis falhas individuais, proporcionando uma infraestrutura resiliente capaz de manter ativo mesmo diante de falhas.

### 3.1.4 Disponibilidade, Escalabilidade e Confiabilidade

Quando uma empresa perde seus dados ou tem falhas de *hardware* em seu ambiente, ela pode ter uma série de prejuízos graves, como por exemplo: interrupção vendas, pausas na produção e comprometimento de informações confidenciais. Esse é um dos motivos que os ambientes de processamento e armazenamentos de dados devem ser projetados para tentar minimizar o dano causado por falhas de equipamentos.

Segundo (SOMASUNDARAM G.; SHRIVASTAVA, A.; FERNANDES, A. P., 2011), a disponibilidade de dados refere-se à capacidade da infraestrutura de operar de forma contínua, sem interrupções prolongadas. A disponibilidade assegura que uma empresa seja capaz de acessar suas informações sempre que necessário, sem enfrentar interrupções inesperadas.

A alta disponibilidade (HA, *High Availability*) é uma característica altamente desejável em sistemas de armazenamento distribuídos. Essa característica garante que os usuários possam acessar seus dados conforme necessário, mesmo em situações de falhas ou interrupções temporárias. *hardware*, (SILBERSCHATZ, A., 2015). A alta disponibilidade está frequentemente ligada a sistemas projetados para tolerar falhas e evitar possíveis inatividades, (LIMA, A. C., 2021). Para implementar um sistema com alta disponibilidade pode demandar um alto investimento.

À medida que as empresas crescem, aumenta a necessidade de instalação de servidores adicionais, a incorporação de novos aplicativos e aumentar a capacidade de armazenamento. Nesse contexto, a solução de armazenamento deve ser escalável para atender à demanda de crescimento da empresa, (SOMASUNDARAM G.; SHRIVASTAVA, A.; FERNANDES, A. P. , 2011). A escalabilidade refere-se à habilidade de um sistema se adaptar ao aumento na demanda de serviços, ou seja, a capacidade de adicionar ou reduzir recursos, tanto em servidores quanto em dispositivos de armazenamento, de acordo com as demandas variáveis, garantindo que o sistema funcione adequadamente e esteja disponível quando necessário, (SILBERSCHATZ, A., 2015).

A confiabilidade refere-se à capacidade de um sistema ou componente de desempenhar suas funções sem interrupções. Em ambientes de armazenamento de dados, a confiabilidade está relacionada à capacidade do sistema em manter os dados acessíveis, mesmo diante de possíveis falhas, (SOMASUNDARAM G.; SHRIVASTAVA, A.; FERNANDES, A. P. , 2011). Neste contexto, um sistema de armazenamento confiável é aquele projetado para minimizar períodos de inatividade, assegurando acesso contínuo aos dados, mesmo diante de potenciais falhas de *hardware*.

### 3.2 IMPLANTAÇÕES DE SISTEMAS DE ARMAZENAMENTO DISTRIBUÍDOS

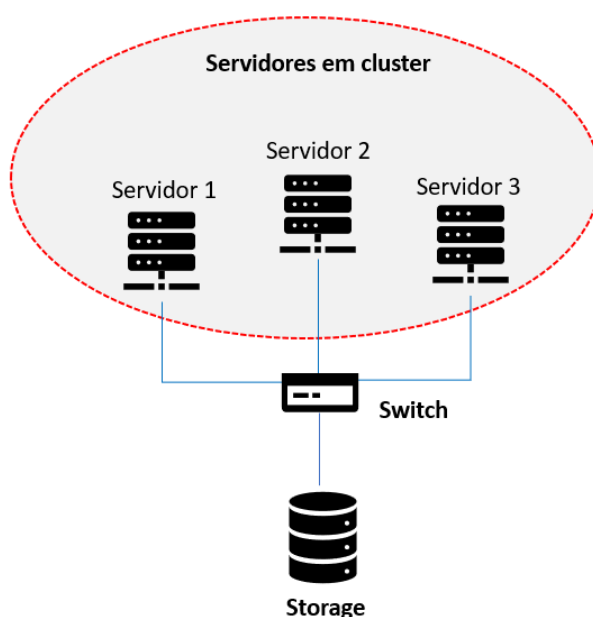
No mercado atual, as implantações de sistemas de armazenamento distribuído são aplicadas em inúmeros cenários. As implementações podem variar de acordo com as necessidades de cada ambiente, incorporando requisitos específicos de desempenho, disponibilidade, custo e escalabilidade. Algumas das principais implantações de sistemas de armazenamento distribuído incluem:

- Sistemas de armazenamento distribuído em *cluster*;
- Sistemas de armazenamento distribuído em nuvem (*cloud*);
- Sistemas de armazenamento distribuído para arquiteturas com virtualização;
- Sistemas de armazenamento distribuído em unidades de armazenamento de dados.

Os sistemas de armazenamento distribuído baseados em redes de computadores consistem em diversos servidores individuais, configurados em conjunto (*cluster*), ou seja, um grupo de computadores ou servidores interligados em uma rede, com o objetivo de proporcionar maior capacidade de processamento, aumentar o armazenamento e garantir alta disponibilidade em comparação com um único servidor isolado, (TANENBAUM, A. S.; STEVEN, M. V., 2007).

Em um centro de dados *Data Center* tradicional, os servidores configurados em *cluster* estão interconectados por meio de uma rede de comunicação. Essa rede pode ser formada por equipamentos de armazenamento (*storage*) e *switch*. A Figura 15 apresenta um exemplo de arquitetura tradicional.

Figura 15 – Exemplo arquitetura tradicional de um *cluster*.

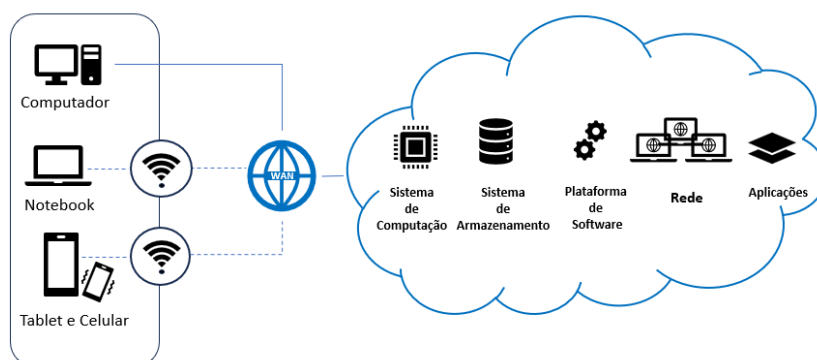


fonte: Produção do próprio autor.

Outra aplicação dos sistemas de armazenamento distribuídos é a computação em nuvem ( *cloud computing*). Segundo (SILBERSCHATZ, A., 2015), a computação em nuvem refere-se ao fornecimento de serviços de computação por meio da *internet*. A computação em nuvem pode ser definida como um sistema de computação sob demanda com acesso onipresente à rede. Isso se deve às suas características, que substituem a necessidade de empresas ou indivíduos adquirirem equipamentos físicos. Em vez disso, esses consumidores podem provisionar e otimizar recursos conforme sua demanda, proporcionando escalabilidade e eliminando a necessidade de gerenciar a infraestrutura física.

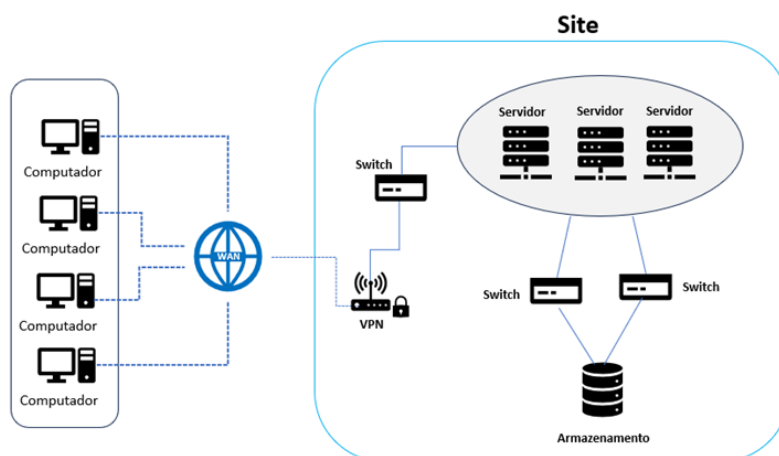
A infraestrutura de computação em nuvem reúne componentes de *hardware* e *software*, os quais são acessados pelos usuários por meio da *internet*, conforme Figura 16. A computação em nuvem oferece uma infraestrutura similar à computação local (*on-premise*), ilustrada na Figura 17.

Figura 16 – Infraestrutura de *cloud*.



fonte: Produção do próprio autor.

Figura 17 – Infraestrutura de Data Center tradicional.



fonte: Produção do próprio autor.

Em ambientes com virtualização, a implantação de sistemas de armazenamento distribuído proporciona maior escalabilidade e disponibilidade para as infraestruturas virtualizadas. Neste tipo de ambiente, os sistemas de armazenamento distribuído visam oferecer recursos de armazenamento compartilhado que podem ser alocados de forma flexível para máquinas virtuais. Uma máquina virtual (VM, *virtual machine*) pode ser descrita como um computador virtual independente, alocado dentro de um servidor físico. As máquinas virtuais compartilham os recursos de um único servidor físico, o que resulta em uma utilização mais eficaz dos recursos computacionais como por exemplo: a unidade central de processamento (CPU, *Central Processing Unit*), memória e armazenamento. O principal objetivo da virtualização é fornecer uma versão virtual de tecnologias essenciais em computação. As principais categorias de virtualização são:

- Virtualização de *hardware*: é uma técnica pela qual uma camada de *software*, conhecida como *hypervisor*, cria máquinas virtuais. Esse modelo de virtualização possibilita a execução simultânea de vários sistemas operacionais em um mesmo servidor físico.
- Virtualização de armazenamento: A virtualização de armazenamento, também conhecida como Armazenamento Definido por *Software* (SDS, *Software Defined Storage*), consiste em uma camada de *software* entre os dispositivos de armazenamento físico e os sistemas que acessam esses dispositivos. A camada de *software*, permite que múltiplos dispositivos de armazenamento sejam agrupados e gerenciados como um único recurso de armazenamento virtual, ocasionando maior flexibilidade, escalabilidade e eficiência no gerenciamento de dados.
- Virtualização de redes: é uma técnica que visa criar uma infraestrutura virtual de rede com base na infraestrutura física existente. Ela possibilita a flexibilização da entrega de funções de rede, permitindo a configuração e gestão eficiente de redes virtuais. A virtualização de rede simplifica a configuração, gestão e escalabilidade da infraestrutura de rede.

Por fim, os sistemas de armazenamento distribuído em unidades de armazenamento de dados referem-se a arquiteturas de armazenamento em que os dados são distribuídos entre várias unidades de

armazenamento individuais, para fornecer disponibilidade ao sistema. A aplicação desse conceito é incorporada na formulação do RAID.

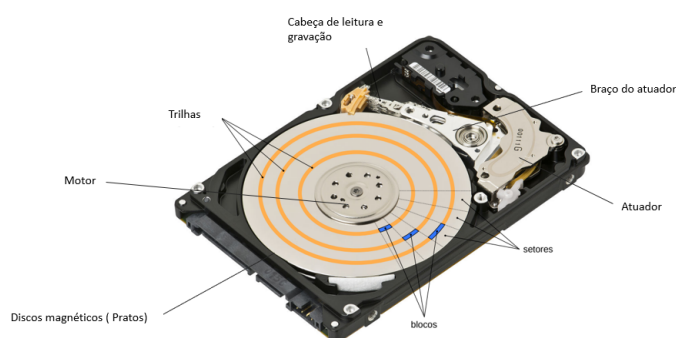
### 3.3 DISPOSITIVOS DE ARMAZENAMENTO DE DADOS

Este capítulo tem como objetivo fornecer uma visão detalhada das principais características e do funcionamento dos dispositivos de armazenamento mais utilizados em sistemas computacionais: a unidade de disco rígido (HDD) e a unidade de estado sólido (SSD).

Unidade de disco rígido (HDD, *Hard Disk Drive*), também conhecido disco rígido (HD, *Hard Disk*), é um componente de armazenamento de dados usado para armazenar informações de forma não volátil. Dessa forma, o disco permite que as informações gravadas não sejam apagadas após o desligamento do dispositivo, (ANCIOTO, A. S. R., 2018).

Um disco rígido é formado por vários componentes móveis que trabalham em conjunto para armazenar e buscar dados. A Figura 18 apresenta a estrutura física interna de um HDD, exibindo os principais componentes físicos de um disco rígido.

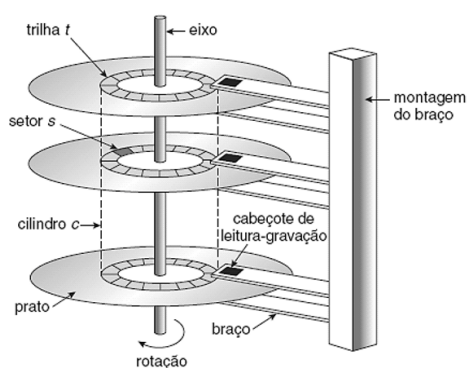
Figura 18 – Principais componente internos de um disco rígido.



fonte: Adaptado de (MAZIERO, C. A., 2019).

Os dados são gravados nos pratos, também chamados de discos magnéticos. Os discos magnéticos possuem um formato semelhante ao de um disco compacto (CD, *compact disc*). Segundo (TANENBAUM, A. S.; Woodhul, A. S., 2009), um disco rígido pode ser composto por um ou mais pratos, onde cada superfície de um prato possui um ímã e braço dedicado, conforme ilustra a Figura 19.

Figura 19 – Esboço da estrutura física de um disco rígido.



fonte: (SILBERSCHATZ, A., 2015)

Os discos magnéticos possuem uma superfície dividida em vários círculos, chamados de trilhas. Segundo (OLIVEIRA, G. Vinícius; Correa, Ricardo., 2021), em média existem de 5.000 a 30.000 trilhas em cada superfície do prato. Além disso, essas trilhas são subdivididas em pequenas seções denominadas setores. Uma única trilha pode abrigar de 100 a 500 setores. Os setores são um dos principais elementos de um disco rígido, pois são os locais onde as informações são gravadas e lidas. Geralmente, cada setor possui um tamanho fixo de dados, variando entre 512 *bytes* ou 4.096 *bytes*, dependendo do padrão de formatação do disco, (MAZIERO, C. A., 2019).

Esses discos giram rapidamente em torno de um eixo central, que é conhecido como o motor elétrico, responsável por girar os pratos a alta velocidade. Outro componente importante é a cabeça de leitura e gravação que são em pequenos ímãs eletromagnéticos posicionados extremamente próximos da superfície dos pratos, (DELGADO, J.; RIBEIRO, C., 2017). Essas cabeças movem-se sobre os pratos para efetuar a leitura e gravação de dados. Elas são fixadas em um braço do atuador móvel que as conecta ao atuador. Esse componente é utilizado para mover as cabeças para a posição correta sobre os discos magnéticos. A placa lógica ou placa controladora é responsável por controlar as operações, gerenciar a leitura e a gravação de dados e coordenar a comunicação entre o disco rígido e a placa-mãe do computador.

O funcionamento de um disco rígido é um processo coordenado, onde cada componente trabalha em conjunto para garantir o armazenamento e o acesso eficientes de dados. O funcionamento de disco rígido pode ser dividido em duas etapas principais:

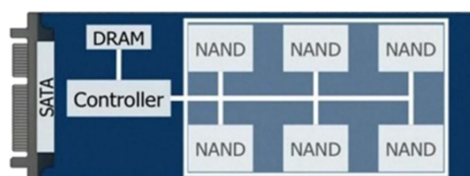
1. O constante giro dos discos magnéticos: O funcionamento de um disco rígido requer a constante rotação dos discos magnéticos a elevadas velocidades. A velocidade de rotação dos discos varia geralmente entre 5400 e 15000 rotações por minuto (RPM, *revolutions per minute*). Essa rotação é impulsionada por um motor elétrico localizado no eixo central dos discos magnéticos, (OLIVEIRA, G. Vinícius; Correa, Ricardo., 2021).
2. Processo de leitura e gravação dos dados: Quando é solicitada a leitura ou gravação de informações no disco rígido, o controlador da placa lógica envia comandos ao atuador. O atuador, por sua vez, move os braços de forma ágil para posicionar as cabeças de leitura e gravação sobre a trilha e o setor desejados. Nesse processo, as cabeças de leitura e gravação realizam a leitura e/ou gravação de dados nos discos magnéticos que compõem o disco rígido.

Unidade de estado sólido (SSD, *Solid State Drive*), é um dispositivo de armazenamento de dados baseado em memória rápida (*flash*). A memória *flash* é caracterizada por ser um dispositivo de armazenamento não volátil (STAIR, R. M.; REYNOLDS, G. W.; Brayant, R., 2021). Um disco SSD é formado por circuitos integrados baseado em um material semicondutor. Os discos SSDs são amplamente utilizados devido à sua velocidade, confiabilidade e capacidade de preservar dados por longos períodos sem a necessidade de alimentação contínua. Além disso, eles proporcionam uma melhoria expressiva no desempenho de sistemas de armazenamento de dados, tornando-os uma escolha popular em cenários que demandam alta performance e segurança na preservação de informações, (ANCIOTO, A. S. R., 2018).

A estrutura de um disco SSD é composta por vários componentes eletrônicos e de armazenamento que trabalham juntos para armazenar e acessar dados. Os principais componentes que compõem um SSD são: memória NÃO-E (NAND, *NOT-AND*), controlador, memória de acesso aleatório dinâmico (DRAM, *Dynamic Random Access Memory*) e interface.

A Figura 20 apresenta a estrutura interna de um disco SSD, destacando os principais componentes que compõem esse dispositivo de armazenamento de dados.

Figura 20 – Estrutura interna de um disco SSD.



fonte: (MUNHOS, F. L. O., 2018)

A memória NAND é uma memória rápida (*flash*) composta por células de memória que conservam os dados de forma não volátil. Existem diferentes tipos de células de memória NAND comuns, incluindo:

- Células de nível único (SLC, *Single level cell*): esse tipo de célula armazena apenas 1 bit de informação por célula. Devido à sua capacidade de armazenar bit 0 (zero) ou 1 (um), essa memória é recomendada para aplicações que exigem um desempenho superior em termos de velocidade de gravação e recuperação de dados. As células SLC são conhecidas por sua velocidade, resistência e confiabilidade. Uma desvantagem associada à tecnologia SLC é o seu custo elevado (KONDO, E., 2017).
- Células de nível duplo (MLC, *Multi level cell*): são usadas para armazenar dois bits por célula ou mais bits por célula, tornando essa opção mais econômica em relação à memória do tipo SLC. No entanto, as células com tecnologia MLC oferecem menor durabilidade e desempenho em relação às SLC (KONDO, E., 2017).
- Célula de nível triplo (TLC, *Triple Level Cell*): esse tipo de célula, que por sua vez, armazena três de informação por célula. Essa característica é fundamental para reduzir os custos e aumentar a capacidade dos dispositivos SSD. As células TLC são amplamente utilizadas devido à sua relação custo benefício (CARVALHO, A. V. F., 2014).
- Célula de nível quádruplo (QLC, *Quad level cell*): a tecnologia mais recente no mundo dos SSDs é a QLC, que armazena quatro bits de dados por célula. Um dispositivo com tecnologia QLC proporciona uma alta densidade e grande capacidade de memória.

Outro componente essencial para o funcionamento de um disco SSD é o controlador, componente responsável por gerenciar a leitura e gravação de dados na memória *flash* distribui os dados que são armazenados ou recuperados das memórias *flash* NAND. Os discos SSDs são formados por memória de acesso aleatório dinâmico (DRAM, *Dynamic Random-Access Memory*). A DRAM é

usada para armazenar informações temporárias e auxiliar na gestão de dados em uso, que podem ser posteriormente transferidos para as células de memória NAND.

Os SSDs e os HDDs possuem diferenças distintas em termos de características físicas e funcionais. Essas diferenças incluem: desempenho, durabilidade, consumo de energia, custo por *gigabyte* (GB) e variedade capacidade disponível. A Figura 21 apresenta uma breve comparação entre a estrutura física de um disco rígido e um disco de estado sólido.

Figura 21 – Comparação entre a estrutura física de um disco HD e disco SSD



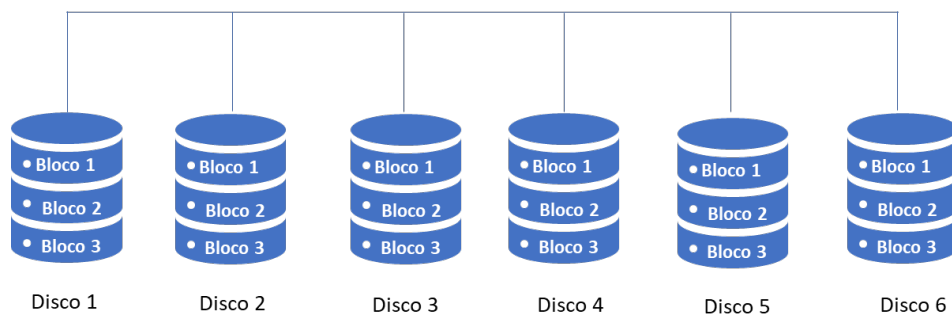
fonte: (PIONTKOSKI, Gilvano, 2017)

Os discos SSDs são dispositivos que apresentam a melhor taxa de leitura e gravação de dados. Isso resulta em tempos de inicialização mais rápidos, carregamento de aplicativos mais ágil e melhor desempenho geral do sistema, ou seja, os discos SSD oferecem melhor desempenho em relação aos discos HDD, (PIONTKOSKI, Gilvano, 2017). Embora os discos SSDs sejam uma opção extremamente atraente no mercado, muitas empresas continuam a adquirir discos HDD tradicionais devido ao fator de custo, que ainda é significativamente mais baixo em comparação com os SSDs (CARVALHO, A. V. F., 2014). Além disso, os discos HDD oferecem uma maior variedade de capacidades de armazenamento. É possível encontrar no mercado discos HDD com capacidades que variam de 128 *gigabytes* (GB) a 24 *terabytes* (TB).

## 4 RAID

Arranjo Redundante de Discos Independentes (RAID, *Redundant Array of Independent Disks*) é um método que combina duas ou mais unidades de discos individuais para formar um conjunto de discos, visando aprimorar as operações de gravação e leitura, bem como oferecer maior confiabilidade ao sistema de armazenamento distribuído, (LIMA, A. C., 2021). Em sistema RAID, os dados são distribuídos e armazenados em blocos nos discos individuais, conforme exemplificado na Figura 22.

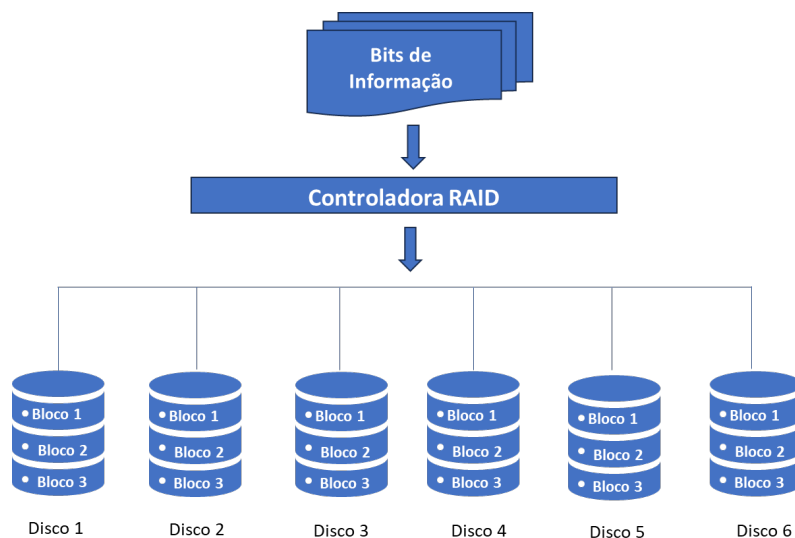
Figura 22 – Sistema de RAID com discos individuais fragmentados em bloco.



fonte: Produção do próprio autor.

Quando discos HDD ou SSD são configurados em RAID, o sistema operacional os enxerga como uma única unidade lógica. A controladora RAID é responsável por gerenciar o envio e a busca de dados, que podem ser fragmentados ou replicados, para os discos do arranjo. A Figura 23 apresenta um esboço do funcionamento de um sistema configurado em RAID.

Figura 23 – Funcionamento geral do RAID



fonte: Produção do próprio autor.

Conforme discutido na Seção 3, a disponibilidade de recursos e a confiabilidade são métricas fundamentais nos sistemas de armazenamento distribuído. Nesse contexto, o RAID é essencial para o funcionamento eficaz desses sistemas, visto que ele proporciona alta disponibilidade mesmo que

ocorra uma falha de disco. O RAID possui a capacidade de recuperar os dados originais em caso de falha de disco, garantindo a confiabilidade dos dados. Além disso, o RAID pode ser implementado para otimizar o desempenho do sistema, especialmente devido a distribuição de dados entre várias unidades de disco HDD ou SSD. Durante operações de gravação ou leitura de dados, o sistema pode acessar diversas unidades de armazenamento simultaneamente, buscando ou gravando informações. Esse método contribui significativamente para aumentar a taxa de transferência de dados, resultando em um notável aprimoramento de desempenho, (LEE, K.; YU, G.; HWANG, Y.; JUN, B.; SON, H.; SONG, Y. H. , 2022).

O conceito RAID surgiu na década de 1980, quando um grupo de pesquisadores formado por David A. Patterson, Garth A. Gibson e Randy H. Katz publicaram pela Universidade da Califórnia Berkeley um artigo chamado "Um caso para arranjos redundantes de baixo custo" ("A Case for Redundant Arrays of Inexpensive Disks"), esse estudo explorou o uso de discos de baixo custo configurados em conjunto para aumentar a confiabilidade e o desempenho de sistema de armazenamento. (PATTERSON, D. A.; GIBSON, G.; KATZ, R. H. , 1988). Inicialmente, o RAID era conhecido como Arranjos Redundantes de Discos Baratos. Entretanto, com o desenvolvimento dos estudos, o termo RAID evoluiu para o que conhecemos hoje como Arranjo Redundante de Discos Independentes (*Redundant Array of Independent Disks*), (SILBERSCHATZ, A., 2015).

#### 4.1 NÍVEIS DE RAID

A configuração dos discos em um sistema RAID pode ser realizada utilizando várias composições diferentes de técnicas base, conhecidas como níveis de RAID. Dessa forma, os níveis de RAID é um termo utilizado para representar as diversas configurações de disco estabelecidas em um sistema de RAID, e cada um deles oferece um conjunto específico de características, como desempenho, redundância e proteção de dados. Esses níveis de RAID são definidos com base nas seguintes técnicas fundamentais:

- Distribuição de dados: Conforme mencionado na Subseção 3.1.1, a distribuição de dados é um método que distribui os dados em vários discos, proporcionando maior desempenho ao sistema.
- Espelhamento: Como discutido na Subseção 3.1.2, o espelhamento é uma técnica de replicação de dados, fornecendo redundância e proteção de dados.
- Paridade: Essa técnica permite a reconstrução de conjuntos de dados em caso de falhas, garantindo a redundância e proteção de dados.

Nesta seção, apresentaremos os princípios dos principais níveis de RAID, incluindo o RAID 0, RAID 1, RAID 2, RAID 3, RAID 4, RAID 5 e RAID 6. Para a elaboração deste capítulo, as principais referências utilizadas foram ((SOMASUNDARAM G.; SHRIVASTAVA, A.; FERNANDES, A. P. , 2011), (MAZIERO, C. A., 2019), (STALLINGS, W., 2017), (SILBERSCHATZ, A., 2015), (TANENBAUM, A. S.; Woodhul, A. S., 2009), (XIANG, L.; XU, Y.; LUI, J.C.S.; CHANG, Q., 2021), (PERTIN, D.; KEMPEN, A.; PARREIN, B.; NORMAND, N., 2015), (HANXU, H.; YUNGHSIANG,

H. S.; KENNETH, S. W.; HUI, L., 2018), (CHEN, Y. , 2022) e (WU, R.; WU, Y.; WANG, M.; J WANG, L. , 2021)).

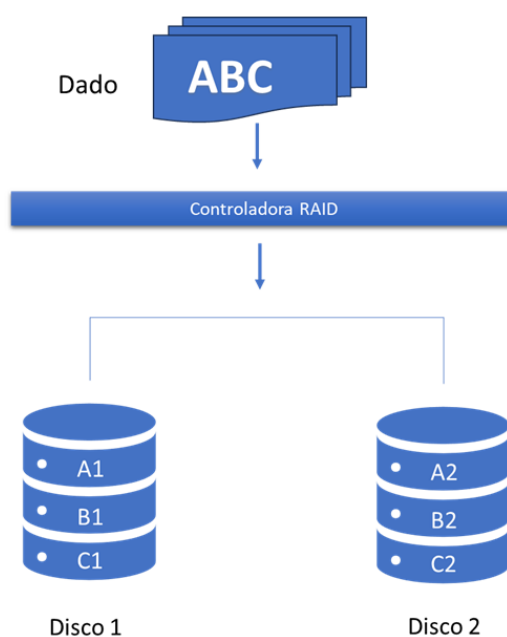
#### 4.1.1 RAID 0: Distribuição

O RAID 0 é baseado na técnica de distribuição de dados, descrita na Subseção 3.1.1. Esse tipo de configuração requer no mínimo duas unidades de disco para ser implementado em um sistema de armazenamento de dados.

Conforme mencionado na Seção 3.3, os discos rígidos são equipados com cabeças de leitura e gravação em cada disco magnético. Dessa forma, quanto mais discos o sistema possui, maior é a quantidade de cabeças de leitura/gravação. Portanto, ao aumentar o número de discos no sistema, a capacidade de leitura e gravação é ampliada, o que, por sua vez, resulta em um melhor desempenho do sistema quando configurado em RAID 0.

A Figura 24, ilustra um exemplo da configuração de discos em RAID 0, no qual um arquivo é fragmentado e gravado, de forma distribuída, em duas unidades de armazenamento independentes.

Figura 24 – RAID 0.



fonte: Produção do próprio autor.

O RAID 0 não oferece redundância ao sistema. Dessa forma, se uma unidade falhar, todos os dados armazenados no RAID 0 serão perdidos. O RAID nível 0 não é considerado um membro típico da família RAID, uma vez que ele não inclui técnicas de redundância para lidar com falhas de disco.

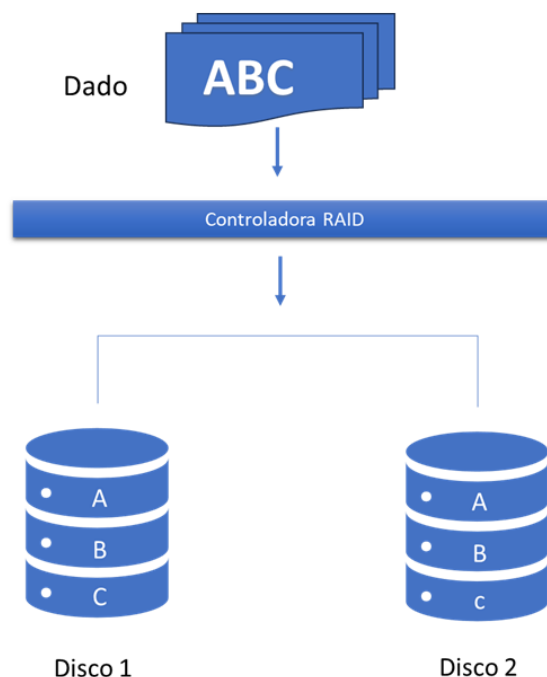
#### 4.1.2 RAID 1: Espelhamento

O RAID 1 utiliza a técnica de espelhamento de dados, discutida na Subseção 3.1.2. Em um sistema de armazenamento configurado em RAID 1, se um dos discos falhar, os dados permanecerão acessíveis

no outro disco espelhado. Essa redundância é fundamental para garantir a disponibilidade e a proteção dos dados.

A Figura 25, mostra a configuração de discos em RAID 1. Nessa configuração, a duplicação de dados ocorre de forma simultânea, resultando em uma cópia idêntica em cada uma das unidades de discos individuais.

Figura 25 – RAID 1.



fonte: Produção do próprio autor.

Embora o RAID 1 ofereça alta disponibilidade e uma sólida proteção aos dados, é importante considerar que essa técnica geralmente possui uma gravação mais lenta, tornando o sistema menos performático. Outra desvantagem do RAID 1 está relacionada com o custo mais elevado em comparação com outros níveis de RAID. Isso ocorre porque a configuração de discos em RAID 1 requer o dobro do espaço em disco físico em relação à capacidade lógica desejada. A implementação do RAID 1 é recomendada para aplicações em que a integridade dos dados e a manutenção da operação contínua são fundamentais, tais como:

- **Dados críticos:** Em servidores que armazenam dados críticos, o RAID 1 é frequentemente usado para garantir que os dados estejam sempre disponíveis, mesmo se uma unidade falhar.
- **Sistemas de monitoramento de vídeo:** Em sistemas de vigilância com monitoramento por vídeo, o RAID 1 é usado para evitar a perda de imagens críticas devido a falhas de disco.
- **Informações bancárias:** Em servidores com informações bancárias requerem que os dados sejam protegidos. O RAID 1 oferece uma solução sólida de proteção de dados, além de garantir que os dados bancários estejam sempre acessíveis.

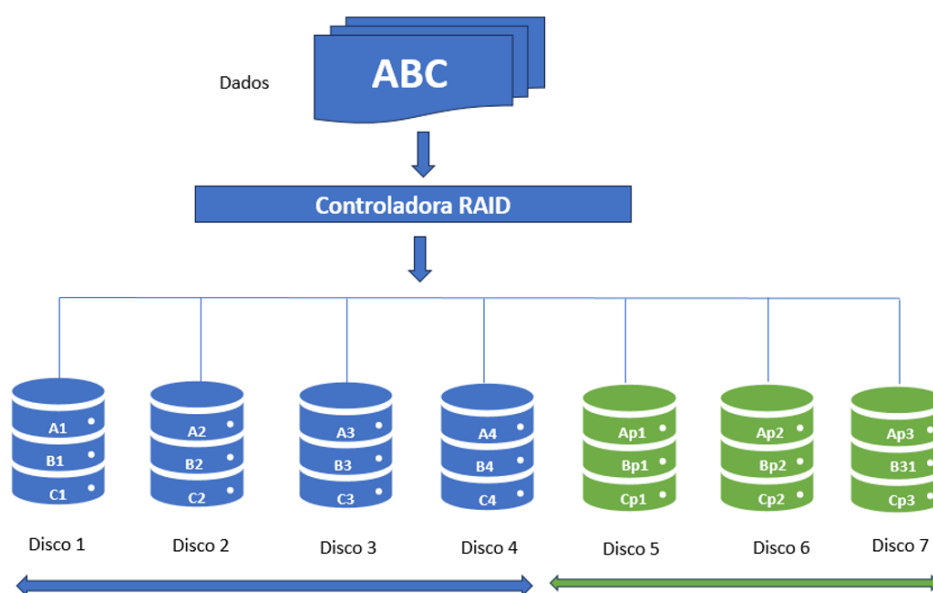
### 4.1.3 RAID 2: Distribuição e Redundância via Código de Hamming

O RAID 2 é uma técnica que fornece proteção de dados aos sistemas de armazenamento distribuído. O RAID 2 é baseado no código de Hamming. Conforme discutido na Subseção 2.1.1, o código de Hamming é um código de correção de erros que adiciona bits de paridade a um conjunto de bits de dados.

No RAID 2, a paridade é gerada por meio do código de Hamming, onde cada bit de dados é armazenado em uma unidade de disco separada, enquanto os bits de paridade são registrados em unidades adicionais, destinadas à detecção e correção de erros.

A Figura 26, apresenta um exemplo da configuração de discos em RAID 2, onde os dados são distribuídos nos três primeiros discos e os três últimos discos são dedicados para o armazenamento dos bits de paridade.

Figura 26 – RAID 2.



fonte: Produção do próprio autor.

**Exemplo 15.** Considerando um sistema de armazenamento distribuído com 7 discos configurados em RAID 2 via código de Hamming  $(7, 4)$ , a mensagem  $u = (100110001111)$  deve ser armazenada. Neste caso, os bits de informação da mensagem  $u$  são armazenados nos primeiros discos do arranjo, como ilustrado na Figura 27.

Figura 27 – Distribuição dos bits de informação da mensagem  $u$ .

| Disco 1 | Disco 2 | Disco 3 | Disco 4 | Disco 5 | Disco 6 | Disco 7 |
|---------|---------|---------|---------|---------|---------|---------|
| 100     | 110     | 001     | 111     | m1      | m2      | m4      |

Abaixo da tabela, duas setas horizontais indicam a divisão: uma seta azul para os discos 1 a 4 e uma seta verde para os discos 5 a 7.

Bits de informação Bits de paridade

fonte: Produção do próprio autor.

Para calcular os bits de paridade, o cálculo da paridade é realizado da seguinte maneira:

- Para  $m_1$  : calcular o bit de paridade a partir dos bits de informação contidos dos discos 1, disco 2 e disco 3. Assim, obtemos:

$$m_1 = k_1 \oplus k_2 \oplus k_3 = 100 \oplus 110 \oplus 001 = 001.$$

- Para  $m_2$  : calcular o bit de paridade a partir dos bits de informação contidos dos discos 1, disco 2 e disco 4. Desse modo, temos:

$$m_2 = k_1 \oplus k_2 \oplus k_4 = 100 \oplus 110 \oplus 111 = 101.$$

- Para  $m_3$  : calcular o bit de paridade a partir dos bits de informação contidos dos discos 2, disco 3 e disco 4. Portanto,

$$m_3 = k_2 \oplus k_3 \oplus k_4 = 110 \oplus 001 \oplus 111 = 000.$$

O resultado do arranjo pode ser visualizado na Figura 28.

Figura 28 – Resultado do exemplo sobre RAID 2.

| Disco 1 | Disco 2 | Disco 3 | Disco 4 | Disco 5 | Disco 6 | Disco 7 |
|---------|---------|---------|---------|---------|---------|---------|
| 100     | 110     | 001     | 111     | 001     | 101     | 000     |

fonte: Produção do próprio autor.

Portanto, temos um código com parâmetro  $(21, 12)$ . Para determinar a matriz geradora resultante desse exemplo, é necessário seguir os passos apresentados no Exemplo 10 discutido na Seção 2.

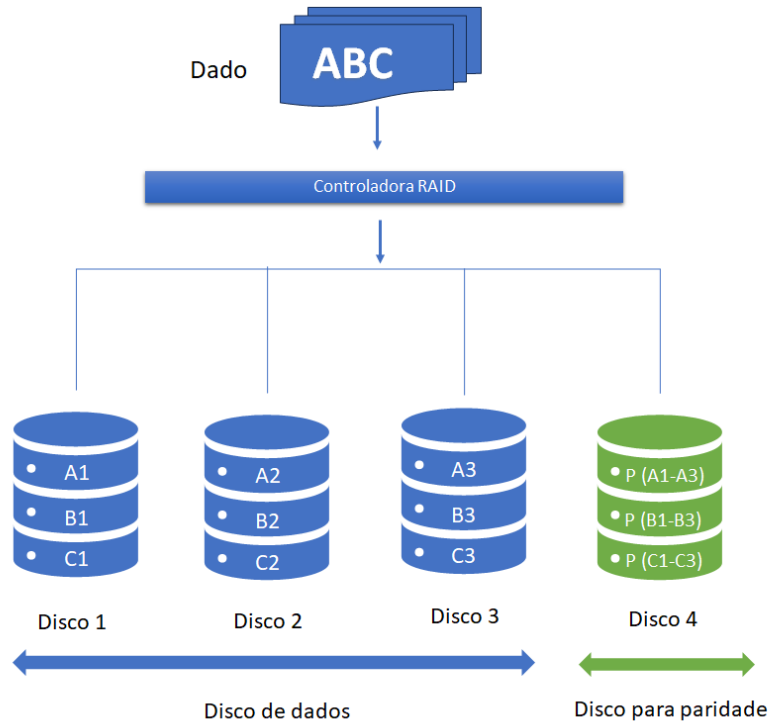
No mercado atual, a configuração de dispositivos de armazenamento utilizando RAID 2 é rara e considerada obsoleta devido ao surgimento de níveis RAID mais simples e eficientes, como o RAID 5.

#### 4.1.4 RAID 3: Distribuição e Redundância via Paridade Unidimensional

O RAID 3 é uma técnica desenvolvida com base na distribuição e na utilização de paridade para fornecer redundância ao sistema. Em sistemas configurados em RAID 3, a reconstrução de dados ocorre por meio da paridade unidimensional apresentado na Subseção 2.2.1.

No RAID 3, a paridade é gerada em nível de *bytes*, onde os dados são divididos a nível de *bytes* e são distribuídos nos blocos de cada disco. E os *bytes* de paridade são armazenados em um disco dedicado. A Figura 29, apresenta um sistema de discos armazenados em RAID 3.

Figura 29 – RAID 3.



fonte: Produção do próprio autor.

**Exemplo 16.** Considere um conjunto de discos configurados em RAID 3, nos quais os seguintes bytes de informação foram armazenados nos discos de dados:  $D_1 = (10101010)$ ,  $D_2 = (11011101)$  e  $D_3 = (01100110)$ . A paridade  $P$  contida no disco  $D_4$ , pode ser calculada utilizando a Equação 14. Desse modo, temos que:

$$D_4 = D_1 \oplus D_2 \oplus D_3 = 10101010 \oplus 11011101 \oplus 01100110,$$

logo, paridade gerada que será armazenada no disco 4 será:

$$D_4 = (00010001).$$

**Exemplo 17.** Com base no exemplo anterior, suponha que o disco  $D_1$  falhe. Nesse caso, os dados podem ser recuperados a partir da paridade contida no disco  $D_4$ . Para efetuar a recuperação, é essencial substituir o disco defeituoso pelo novo disco. Neste caso, o novo disco é representado por  $D_5$ . Após acrescentar novo disco no sistema, a paridade do disco  $D_4$  é usada para reconstruir os dados referentes ao disco  $D_1$ . Os dados a serem restaurados no  $D_5$  é calculada pela seguinte expressão:

$$D_5 = D_4 \oplus D_2 \oplus D_3.$$

Substituindo os bytes de informação e bytes de paridade, obtemos:

$$D_5 = 00010001 \oplus 11011101 \oplus 01100110 = 10101010.$$

A recuperação dos dados armazenados no disco  $D_1$  é realizada através da aplicação da operação

*XOR baseado nas técnicas discutidas no código matricial com paridade unidimensional. Desta forma, o novo disco  $D_5$  terá os seguintes dados:*

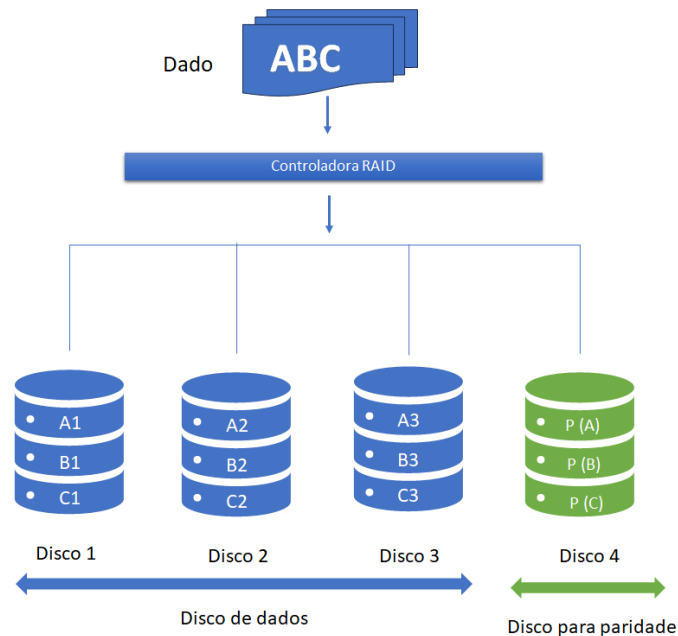
$$D_5 = (10101010).$$

#### 4.1.5 RAID 4: Distribuição e Redundância via Paridade Unidimensional

O RAID 4 é uma técnica baseada na distribuição de dados e paridade unidimensional. Sua finalidade é oferecer redundância de dados e melhorar o desempenho do sistema de armazenamento. Similar ao RAID 3, o RAID 4 utiliza um único disco dedicado à paridade. A principal diferença entre o RAID 4 e o RAID 3 está relacionada à distribuição dos dados no sistema de armazenamento. No RAID 4, os dados são distribuídos em blocos de bits, enquanto no RAID 3, os dados são distribuídos em *bytes*.

A Figura 30, mostra um sistema com 4 discos configurados em RAID 4.

Figura 30 – RAID 4.



fonte: Produção do próprio autor.

**Exemplo 18.** Considere um sistema de armazenamento distribuído com um arranjo de 4 discos configurados em RAID 4, no qual os bits de informação de uma mensagem  $u$  são distribuídos em bloco entre os disco, conforme a Figura 31.

Figura 31 – Distribuição de dados no arranjo de disco em RAID 4.

| Disco 1    | Disco 2    | Disco 3    | Disco 4 |
|------------|------------|------------|---------|
| 1010101000 | 1101110100 | 0001100110 | P1      |
| 1101110111 | 1010101011 | 0001100111 | P2      |
| 0001100111 | 1101110100 | 1010101000 | P3      |

fonte: Produção do próprio autor.

A integridade dos dados armazenados é garantida através do cálculo da paridade bidimensional em bloco, obtida por meio da Equação 13. Dessa forma, obtemos:

$$P_1 = 1010101000 \oplus 1101110100 \oplus 0001100110 = 10110111010,$$

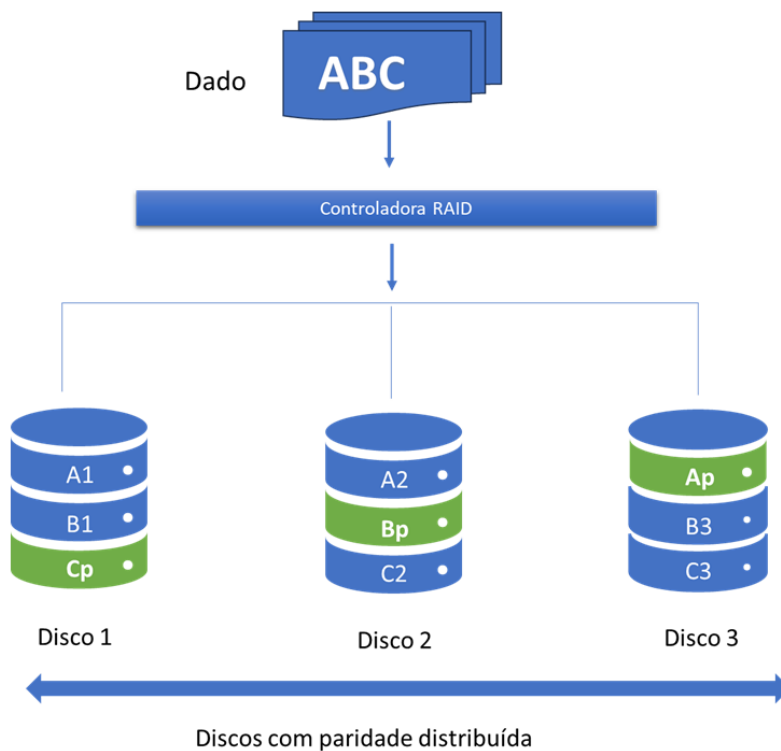
$$P_2 = 1101110111 \oplus 1010101011 \oplus 0001100111 = 0110111011,$$

$$P_3 = 0001100111 \oplus 1101110100 \oplus 1010101000 = 0110111011.$$

#### 4.1.6 RAID 5: Distribuição e Paridade

O RAID 5 é uma técnica de armazenamento desenvolvida com base na distribuição de dados e paridade simples distribuída pelo arranjo de discos formado por  $N + 1$ , onde  $N$  é o número de discos. Nos sistemas de armazenamento RAID 5, a paridade é distribuída em todos os discos de um arranjo, superando assim as desvantagens encontradas nos RAID 3 e 4, que utilizam discos dedicados exclusivamente para a paridade. Com isso o RAID 5 proporciona melhores taxas de transferências e proteção de dados para os sistemas de armazenamento de dados. A Figura 32, mostra a configuração de discos em RAID 5.

Figura 32 – RAID 5.



fontes: Produção do próprio autor.

Em sistemas com RAID 5, a controladora de RAID distribui os dados pelo conjunto de disco, proporcionando maior desempenho na leitura de dados, uma vez que, essa técnica permite que várias operações de leitura sejam executadas simultaneamente, aumentando a taxa de transferência de dados.

No entanto, o RAID 5 oferece uma taxa de transferência de gravação mediano devido ao cálculo da paridade.

Para o cálculo da paridade, o RAID 5 utiliza os conceitos visto nos códigos matriciais e de paridade diagonal, discutidas na Subseção 2.2.3.

**Exemplo 19.** Consider um arranjo com 4 discos configurado em RAID 5 e a mensagem  $u = (100000100011100000100011100100100000110000100000111111100101100011100000)$  a ser armazenada neste disco. Para calcular a paridade deste arranjo deve-se seguir os seguintes passos:

**Passo 1:** Distribuir os dados pelo arranjo de disco, conforme a Figura 33.

Figura 33 – Distribuição dos bits de informação no arranjo de disco configurado em RAID 5.

| D1     | D2     | D3     | D4     |
|--------|--------|--------|--------|
| 100000 | 100011 | 100000 | P1     |
| 100011 | 100100 | P2     | 100000 |
| 110000 | P3     | 100000 | 111111 |
| P4     | 100101 | 100011 | 100000 |

fonte: Produção do próprio autor.

**Passo 2:** Calcular as paridade diagonais de cada bloco dos discos. Para isso é necessário usar as Equações 15. Desse modo, obtemos:

$$P_1 = 100000 \oplus 100011 \oplus 100000 = 100011,$$

$$P_2 = 100011 \oplus 100100 \oplus 100000 = 100111,$$

$$P_3 = 110000 \oplus 100000 \oplus 111111 = 101111,$$

$$P_4 = 100101 \oplus 100011 \oplus 100000 = 100110.$$

**Passo 3:** Adicionar os bit de paridade calculados no arranjo de disco, conforme a Figura 34.

Figura 34 – Distribuição dos bits de informação e paridade no arranjo configurado em RAID 5.

| D1      | D2      | D3     | D4     |
|---------|---------|--------|--------|
| 100000  | 100011  | 100000 | 100011 |
| 100011  | 100100  | 100111 | 100000 |
| 110000  | 101 111 | 100000 | 111111 |
| 101 110 | 100101  | 100011 | 100000 |

fonte: Produção do próprio autor.

Nota-se que as paridade determinadas por meio das operações XOR são distribuídas em todos os discos do arranjo.

**Exemplo 20.** Considere o arranjo de disco mencionado no exemplo anterior. Suponha que ocorra uma falha no disco  $D_4$ , conforme a Figura 35.

Figura 35 – Arranjo de discos configurados em RAID 5 com a presença de um disco danificado.

| Disco 1 | Disco 2 | Disco 3 | Disco 4 |
|---------|---------|---------|---------|
| 100000  | 100011  | 100000  | 100011  |
| 100011  | 100100  | 100111  | 100000  |
| 110000  | 101111  | 100000  | 110000  |
| 100110  | 100101  | 100011  | 100000  |

fonte: Produção do próprio autor.

Devido à paridade implementada no arranjo RAID 5, é possível recuperar os dados armazenados no disco  $D_4$  por meio da operação XOR. Para isso, deve-se seguir as etapas descrita abaixo.

**Etapa 1:** Substituir o disco defeituoso por um novo disco, conforme ilustrado na Figura 36.

Figura 36 – Adição do disco novo ao arranjo.

| Disco 1 | Disco 2 | Disco 3 | Disco 5 (novo) |
|---------|---------|---------|----------------|
| 100000  | 100011  | 100000  | X1             |
| 100011  | 100100  | 100111  | X2             |
| 110000  | 101111  | 100000  | X3             |
| 100110  | 100101  | 100011  | X4             |

fonte: Produção do próprio autor.

**Etapa 2:** A recuperação de dados é realizada a partir dos dados dos outros discos por meio da operação de XOR. Assim, temos:

$$X1 = 100000 \oplus 100011 \oplus 100000 = 100011, \quad (1)$$

$$X2 = 100011 \oplus 100100 \oplus 100111 = 100000, \quad (2)$$

$$X3 = 110000 \oplus 101111 \oplus 100000 = 111111, \quad (3)$$

$$X4 = 100101 \oplus 100011 \oplus 100011 = 100000. \quad (4)$$

Portanto, os dados foram recuperados, como ilustrado na Figura 37.

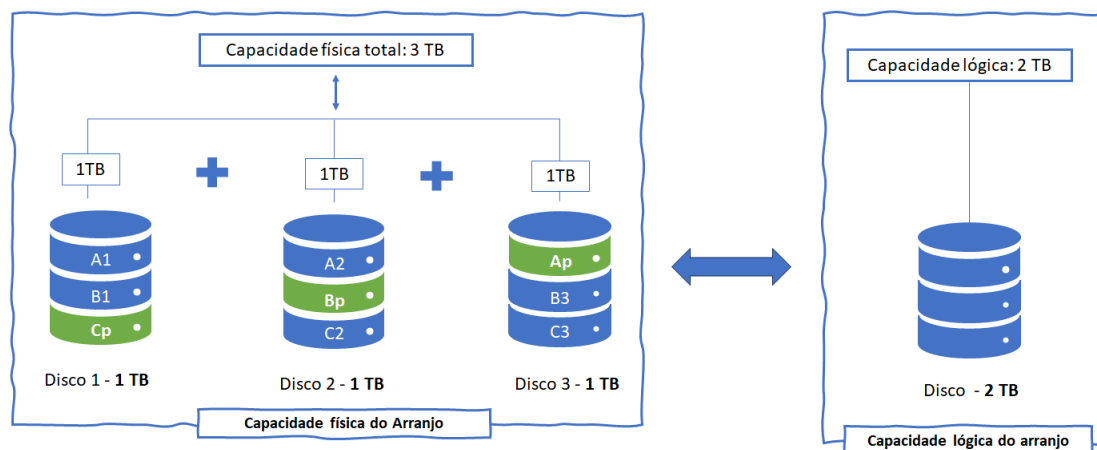
Figura 37 – Resultado da recuperação dos dados em um sistema configurado em RAID 5.

| Disco 1 | Disco 2 | Disco 3 | Disco 5 (novo) |
|---------|---------|---------|----------------|
| 100000  | 100011  | 100000  | 100011         |
| 100011  | 100100  | 100111  | 100000         |
| 110000  | 101111  | 100000  | 111111         |
| 100110  | 100101  | 100011  | 100000         |

fonte: Produção do próprio autor.

Para a implementação do RAID 5 é necessário no mínimo de 3 discos, para que tanto os dados quanto os bits de paridade possam ser distribuídos adequadamente pelo arranjo. Devido à distribuição dos bits de paridade pelo arranjo, a capacidade lógica é menor que a capacidade física dos discos. Por padrão, é reservada a capacidade de um disco para armazenar as informações de paridade.

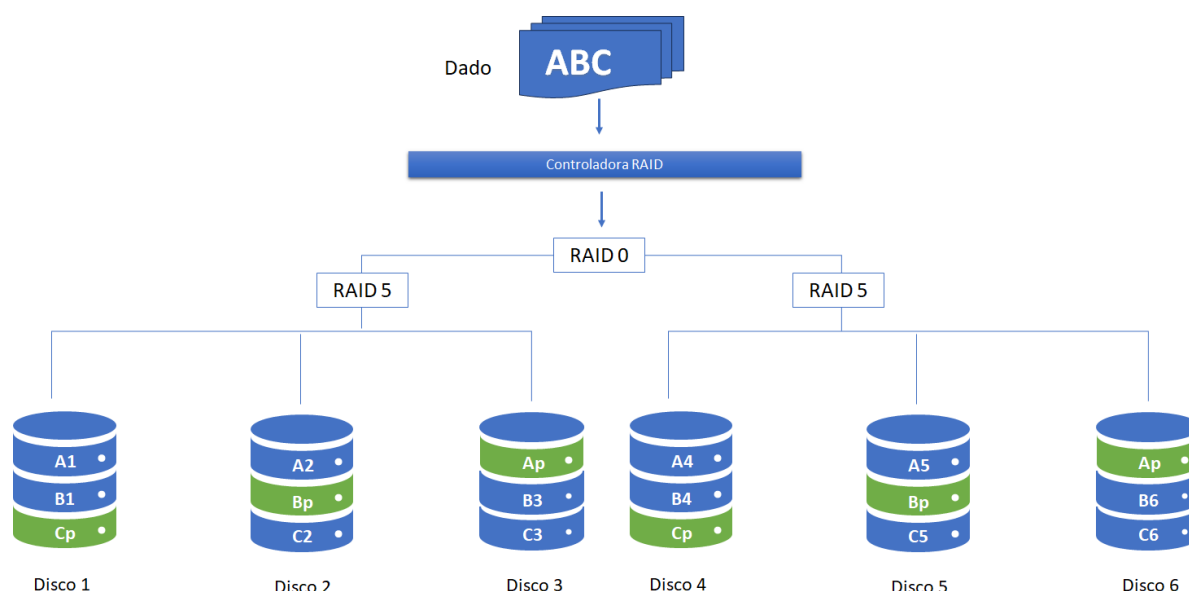
Figura 38 – RAID 5: Comparação entre a capacidade física dos discos e a capacidade lógica



fonte: Produção do próprio autor.

O RAID 5 também é utilizado na formação de novas combinações, como é o caso do RAID 50. O RAID 50, também conhecido como RAID 5+0, é uma técnica que mescla os níveis RAID 5 e RAID 0. Desse modo, a paridade distribuída pelo RAID 5 é combinada com a distribuição característica do RAID 0, com o objetivo de aprimorar o desempenho do RAID 5 sem comprometer a robustez na proteção dos dados. O RAID 50 requer no mínimo seis discos, conforme ilustra a Figura 39.

Figura 39 – RAID 50.



fonte: Produção do próprio autor.

Os sistemas configurados em RAID 50 oferecem um equilíbrio entre segurança e desempenho, tornando essa configuração adequada para aplicações que exigem ambas as características. As principais aplicações do RAID 50 são:

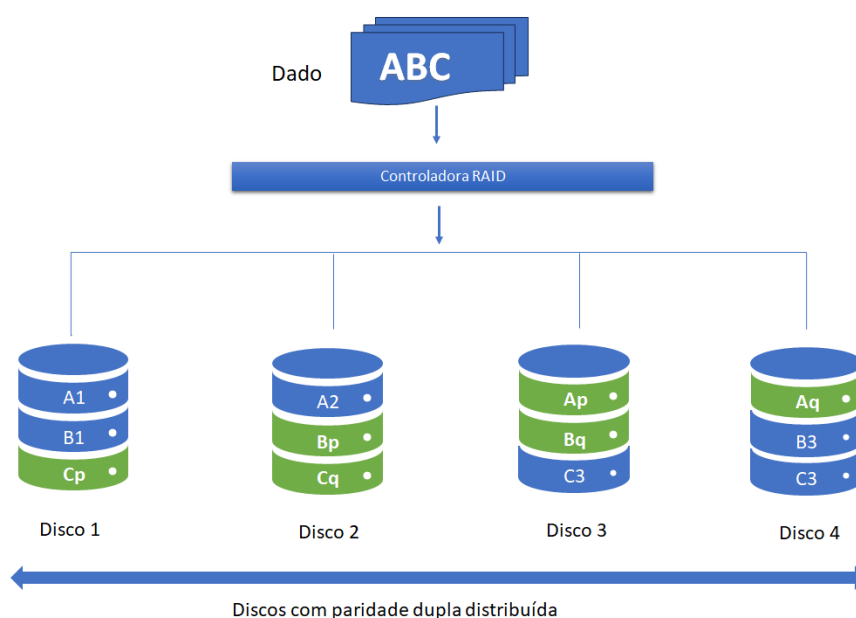
- Servidores de banco de dados: Essa técnica é usualmente aplicada em servidores de banco de dados que precisam de um alto desempenho em operações de leitura e gravação e proteção de dados contra perda de dados.
- Aplicações de alta disponibilidade: O RAID 50 também é usado em ambientes em que a alta disponibilidade é essencial, como servidores de *e-mail*.

#### 4.1.7 RAID 6: Paridade Dupla via Código RDP e Código EVENODD

O RAID 6 é uma técnica de armazenamento baseada na distribuição de dados com paridade dupla. Nos sistemas de armazenamento RAID 6, a principal característica é a capacidade de tolerar a falha de até dois discos simultaneamente. Isso significa que, se ocorrer a falha de dois discos em um sistema de armazenamento configurado com RAID 6, os dados ainda podem ser recuperados por meio da paridade. No entanto, se mais de dois discos falharem, a recuperação de dados se torna impossível, (SOMASUNDARAM G.; SHRIVASTAVA, A.; FERNANDES, A. P. , 2011).

A Figura 40, apresenta um exemplo de disco com configurações em RAID 6 tradicional, onde os dados são distribuídos pelos discos do conjunto e os bits de paridade também são distribuídos pelo pelos blocos do sistema a paridade  $P$ , é representada pelos blocos de paridade:  $A_p$ ,  $B_p$  e  $C_p$ . E a paridade  $Q$ , por sua vez, é representada pelos blocos de paridade:  $A_q$ ,  $B_q$  e  $C_q$ . Embora a arquitetura mais comum para o RAID 6 envolva a distribuição dos bits de paridade entre os discos, é importante destacar que nem todos os arranjos configurados em RAID 6 seguem esse método.

Figura 40 – RAID 6 com paridade distribuída.



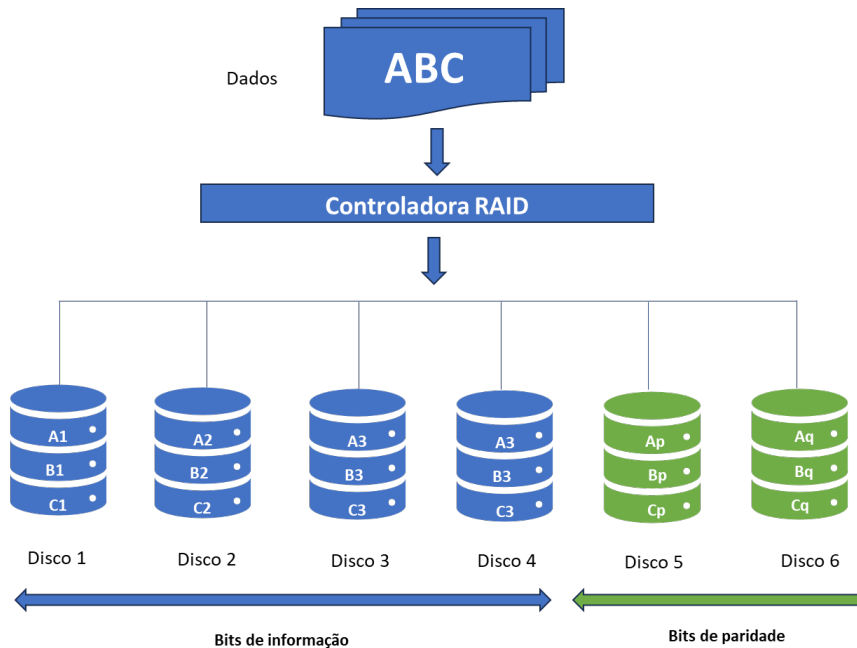
fonte: Produção do próprio autor.

Em sistemas de RAID 6, a correção de erros pode ser realizada por meio de uma vasta variedade de técnicas de detecção e correção de erro. Neste trabalho exploraremos a implementação do RAID 6 por meio dos códigos RDP e EVENODD, ambos códigos analisados na Subseção 2.2.4.

Os códigos RDP e EVENODD são aplicados em configurações de RAID 6 devido às suas características de proteção contra falhas de até dois discos simultaneamente. Nesses tipos de códigos, a

paridade é calculada utilizando técnicas baseadas em operações XOR. Ambos os códigos reservam dois discos do conjunto para o armazenamento de bits de paridade, conforme mostrado na Figura 41.

Figura 41 – RAID 6 com discos exclusivos para armazenamento de paridade.

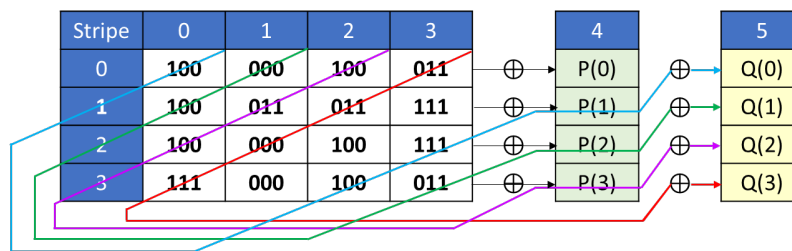


fonte: Produção do próprio autor.

**Exemplo 21.** Considere um conjunto de seis discos configurados em RAID 6, em que a mensagem  $u = (100000100011100011011111100000100111111000100011)$  deve ser armazenada. Neste arranjo RAID 6, os bits de informação da mensagem  $u$  são distribuídos em blocos entre os quatro primeiros discos, enquanto os bits de paridade são armazenados nos dois últimos discos. Para calcular os bits de paridade deste arranjo usando o código RDP, é necessário seguir os passos abaixo:

**Passo 1:** Distribuir os bits de dados pelos blocos dos disco, de acordo com a Figura 42:

Figura 42 – Código RDP: distribuição de dados no arranjo de disco em RAID 6.



fonte: Produção do próprio autor.

**Passo 2:** Calcular a paridade  $P$  e  $Q$ .

- Cálculo da paridade  $P$ : A verificação da paridade  $P$  via código RDP pode ser calculada por meio da técnica de paridade bidimensional entre as linhas da matriz, semelhante ao Exemplo 13, apresentado na Subseção 2.2.2. Assim, os valores de verificação para cada bloco da paridade  $P = \{P(0), P(1), P(2), P(3)\}$  resultam em:

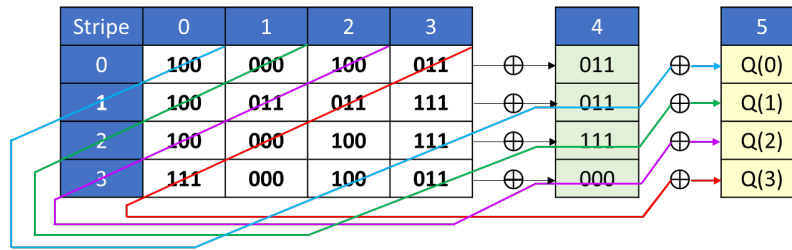
$$P(0) = 100 \oplus 000 \oplus 100 \oplus 011 = 011,$$

$$P(1) = 100 \oplus 011 \oplus 011 \oplus 111 = 011,$$

$$P(2) = 100 \oplus 000 \oplus 100 \oplus 111 = 111,$$

$$P(3) = 111 \oplus 000 \oplus 100 \oplus 011 = 000.$$

Figura 43 – Código RDP: adição dos bits de paridade  $P$  ao sistema em RAID 6.



fonte: Produção do próprio autor.

- Cálculo da Paridade  $Q$ : O cálculo da verificação das paridades  $Q$  segue uma lógica semelhante ao Exemplo 13. Para determinar as paridades  $Q = \{Q(0), Q(1), Q(2), Q(3)\}$ , deve utilizar os conceitos e técnicas do código RDP, discutidas na Subseção 2.2.4. Com este procedimento, obtém-se:

$$Q(0) = 100 \oplus 100 \oplus 111 \oplus 011 = 100,$$

$$Q(1) = 000 \oplus 100 \oplus 011 \oplus 111 = 000,$$

$$Q(2) = 100 \oplus 011 \oplus 100 \oplus 000 = 011,$$

$$Q(3) = 011 \oplus 011 \oplus 000 \oplus 111 = 111.$$

O resultado do cálculo da paridade  $P$  e  $Q$  é descrito na Figura 44.

Figura 44 – Arranjo de disco em RAID 6 com paridades geradas via código RDP.

| Stripe | 0   | 1   | 2   | 3   | 4   | 5   |
|--------|-----|-----|-----|-----|-----|-----|
| 0      | 100 | 000 | 100 | 011 | 011 | 100 |
| 1      | 100 | 011 | 011 | 111 | 011 | 000 |
| 2      | 100 | 000 | 100 | 111 | 111 | 011 |
| 3      | 111 | 000 | 100 | 011 | 000 | 111 |

fonte: Produção do próprio autor.

**Exemplo 22.** Seja  $u = (1001011010010100001111011110100011011011)$  a mensagem a ser armazenada em um sistema de armazenamento de dados composto por sete discos configurados em RAID 6, o cálculo das paridades  $P$  e  $Q$ , pode ser realizado utilizando o código EVENODD. Para isso, deve-se seguir as orientações citadas abaixo.

**Passo 1:** Distribuição da mensagem  $u$ .

O primeiro passo consiste em dividir os bits da mensagem em blocos, como mostrado na Figura 45.

Figura 45 – Código EVENODD: distribuição de dados no arranjo de disco em RAID 6

| Stripe | 0  | 1  | 2  | 3  | 4  | 5    | 6    |
|--------|----|----|----|----|----|------|------|
| 0      | 10 | 01 | 01 | 10 | 10 | P(0) | Q(0) |
| 1      | 01 | 01 | 00 | 00 | 11 | P(1) | Q(1) |
| 2      | 11 | 01 | 11 | 10 | 10 | P(2) | Q(2) |
| 3      | 00 | 11 | 01 | 10 | 11 | P(3) | Q(3) |

fonte: Produção do próprio autor.

**Passo 2:** Cálculo da paridade  $P$ .

O cálculo da paridade  $P$  via código EVENODD, pode ser determinada por meio da técnica de paridade bidimensional entre as linhas da matriz, similar ao exemplo 29. Dessa forma, a paridade  $P$ , representada  $P(0), P(1), P(2), P(3)$ , será:

$$P(0) = 10 \oplus 01 \oplus 01 \oplus 10 \oplus 10 = 10,$$

$$P(1) = 01 \oplus 01 \oplus 00 \oplus 00 \oplus 11 = 11,$$

$$P(2) = 11 \oplus 01 \oplus 11 \oplus 10 \oplus 10 = 01,$$

$$P(3) = 00 \oplus 11 \oplus 01 \oplus 10 \oplus 11 = 11.$$

**Passo 3:** Cálculo da paridade  $Q$ .

O cálculo da paridade  $Q$  é diferente em relação ao código RDP, pois no código EVENODD, apresenta o fator  $h$ . A determinação da paridade  $Q = \{Q(0), Q(1), Q(2), Q(3)\}$  pode ser realizada considerando o trajeto identificado por cada linha, de acordo com a Figura 46.

- Determinado bits do fator  $h$ :

$$h(0) = 10 \oplus 00 \oplus 01 \oplus 11 \oplus 11 = 10.$$

- Calculando  $Q(0)$  com base na linha amarela:

$$Q(0) = 10 \oplus 01 \oplus 10 \oplus 11 \oplus h = 00.$$

- Calculando  $Q(1)$  seguindo a linha azul:

$$Q(1) = 01 \oplus 01 \oplus 10 \oplus 10 \oplus h = 10.$$

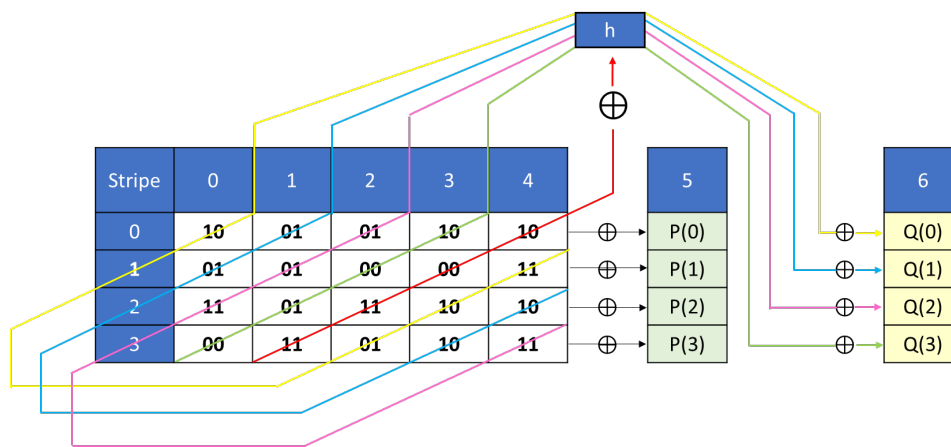
- Calculando  $Q(2)$ , por meio do caminhado identificado pela linha rosa:

$$Q(2) = 01 \oplus 01 \oplus 11 \oplus 11 \oplus h = 10.$$

- Por fim, calculando  $Q(3)$ , com base no trajeto identificado pela linha verde:

$$Q(3) = 10 \oplus 00 \oplus 01 \oplus 00 \oplus h = 01.$$

Figura 46 – Código EVENONDD: demonstração do cálculo da paridade  $Q$ .



fonte: Produção do próprio autor.

Uma desvantagem do RAID 6 é que ele exige mais capacidade física para atender à capacidade lógica desejada, o que pode aumentar o custo do projeto. Por exemplo, se sua aplicação requer 8 *terabytes* (TB) de capacidade lógica e você tem 4 unidades de armazenamento, ao configurar os discos em RAID 6, será necessário adicionar no mínimo 4 discos de 4 *terabytes* (TB) cada para alcançar essa capacidade. Isso resulta em um maior investimento em unidades de disco em comparação com algumas outras configurações de RAID.

Assim como o RAID 5, o RAID 6 é usado para a criação de níveis de RAID mais avançados, como o RAID 60. O RAID 60 é uma técnica que combina os níveis RAID 6 e RAID 0. No arranjo RAID 60, o RAID 0 é responsável pelo desempenho, enquanto o RAID 6 oferece confiabilidade por meio da implementação de paridade dupla. Essa fusão entre RAID 6 e RAID 0, proporciona uma solução equilibrada, combinando os benefícios de desempenho de leitura gerados pela maior quantidade de dados do arranjo com uma alta proteção de dados.

Os sistemas em RAID 60 são configurados com um mínimo de oito discos em seu arranjo. A implementação do RAID 60 é recomendada em aplicações que exigem desempenho e alta confiabilidade, especialmente em ambientes críticos, como instituições financeiras. É importante destacar que o RAID 60 requer o dobro do número de discos para armazenamento em comparação com o RAID 6, o que pode resultar em um custo relativamente mais alto e em uma maior complexidade de implementação.

## 5 ANÁLISE DE DESEMPENHO ENTRE NÍVEIS DE RAID

Ao longo dos anos, surgiram diversos modelos de RAID, cada um apresentando vantagens e desvantagens específicas. A escolha do modelo de RAID a ser implementado varia de acordo com requisitos e recursos disponíveis. Em geral, o RAID é adotado com o objetivo de aprimorar a taxa de transferência, aumentar a disponibilidade dos recursos e assegurar a proteção dos dados por meio da tolerância a falhas aplicada ao sistema. A Tabela 1 e a Tabela 2 apresenta uma breve comparação entre os principais níveis de RAID discutidos neste trabalho, destacando suas principais características e limitações:

Tabela 1 – Principais técnicas e códigos dos Níveis de RAID analisados

| Nível  | Distribuição | Espelhamento | Paridade | Principais Códigos                          |
|--------|--------------|--------------|----------|---------------------------------------------|
| RAID 0 | X            |              |          | Não possui                                  |
| RAID 1 |              | X            |          | Não possui                                  |
| RAID 2 | X            |              | X        | Código de Hamming                           |
| RAID 3 | X            |              | X        | Código matricial de paridade unidimensional |
| RAID 4 | X            |              | X        | Código matricial de paridade unidimensional |
| RAID 5 | X            |              | X        | Código matricial de paridade diagonal       |
| RAID 6 | X            |              | X        | Código RDP e EVENODD                        |

fonte: Produção do próprio autor.

Tabela 2 – Comparação de desempenho entre os Níveis de RAID

| Nível  | Taxa de transferência de gravação | Taxa de transferência de leitura | Tolerância a falhas |
|--------|-----------------------------------|----------------------------------|---------------------|
| RAID 0 | Alto                              | Alto                             | 0                   |
| RAID 1 | Médio                             | Alto                             | 1                   |
| RAID 2 | Médio                             | Médio                            | 1                   |
| RAID 3 | Baixo                             | Médio                            | 1                   |
| RAID 4 | Médio                             | Médio                            | 1                   |
| RAID 5 | Médio                             | Alto                             | 1                   |
| RAID 6 | Baixo                             | Alto                             | 2                   |

fonte: Produção do próprio autor.

Conforme visto no Capítulo 4, o RAID 0 é o único nível de RAID que não oferece proteção contra a perda de dados, ou seja, se um disco falhar todas as informações armazenadas nesse disco serão perdidas. Apesar do RAID 0 não fornecer redundância ao sistema, ele apresenta alto desempenho em operações de gravação e leitura. Geralmente, o RAID 0 é utilizado em cenários onde o desempenho e o custo são mais críticos do que a proteção dos dados, como por exemplo em configurações de jogos ou edição de vídeo, onde é mais importante obter acesso rápido aos dados do que proteger os dados contra as falhas de disco.

No RAID 1, os dados são armazenados em um disco, são simultaneamente replicados nos demais discos do arranjo. Devido a essa arquitetura, o RAID 1 oferece redundância ao sistema, pois se um disco falhar, todas as informações ainda estão disponíveis nos outros discos, garantindo a confiabilidade e disponibilidade dos dados.

Devido à sua complexidade e custo elevado, o RAID 2 não é mais usado no mercado. Isso se deve ao fato de que a implementação do RAID 2 demanda uma grande quantidade de discos para proporcionar efetivamente redundância e desempenho. Além disso, surgiram outros níveis de RAID que oferecem melhor desempenho com custos menores e codificação mais simples. O RAID 1 é indicado para ambientes que demandam armazenar dados críticos, como por exemplo servidores utilizados para o monitoramento de vídeo e armazenamento de arquivos bancários.

Embora o RAID 3 e 4 ofereçam maior simplicidade em comparação com o RAID 2, sua adoção não é tão difundida devido à popularização de outros níveis de RAID e à sua limitada tolerância a falhas.

O RAID 3 e RAID 4 são baseados no uso de códigos de paridade em formato matricial para fornecer redundância ao sistema. Ambos os modelos incluem um disco dedicado para armazenar as informações de paridade. A principal diferença entre o RAID 3 e o RAID 4 está relacionada à distribuição dos dados no sistema de armazenamento. No RAID 3, os dados são distribuídos em *bytes*, enquanto no RAID 4, os dados são distribuídos em blocos. Essa diferença é fundamental na organização dos dados, ocasionando uma influência direta sobre o desempenho e a eficiência de cada modelo.

O RAID 5 representa uma evolução em relação aos níveis anteriores de RAID, como RAID 3 e RAID 4. O RAID 5 é implementado por meio de código de matriz diagonal, no qual os dados e bit de paridade são distribuídos entre todos os discos, resultando em melhorias significativas nas taxas de leitura e gravação do sistema. O RAID 5 é indicado em aplicações que requerem um bom desempenho para leitura e desempenho moderado para gravação de dados, ao mesmo tempo em que garante a proteção aos dados. Atualmente, o RAID 5 é um dos tipos de RAID mais utilizados, sendo uma escolha frequente em ambientes empresariais e de armazenamento, como por exemplo: servidores de arquivos (*File Server*), servidores para virtualização, servidores de banco de dados, servidores de aplicativos e servidores de mídia.

O RAID 6 proporciona ao sistema a tolerância a falhas de até dois discos por meio da implementação de paridade dupla. Como discutido no Capítulo 4, o RAID 6 pode ser implementado por meio de diferentes códigos. Sua característica fundamental é a capacidade de tolerar falhas duplas. Devido a essa particularidade, o RAID 6 é aplicado em ambientes que demandam um aumento no fator de redundância e na segurança dos dados armazenados. No entanto, o cálculo adicional da segunda paridade resulta em uma perda de desempenho em comparação com o RAID 5.

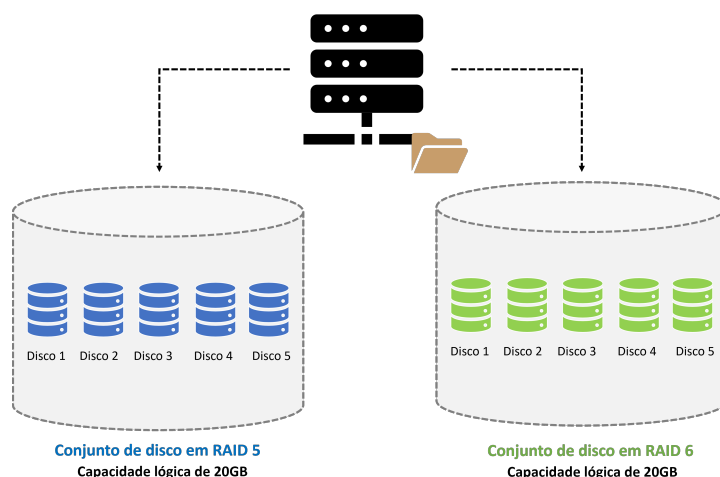
## 5.1 ANÁLISE DE UM AMBIENTE CONFIGURADO EM RAID 5 E RAID 6

Como mencionado anteriormente, os sistemas de RAID 2, 3 e 4 não são mais utilizados no mercado atual, em virtude do surgimento de técnicas mais avançadas, como os sistemas de RAID 5 e 6. Diante desse contexto, decidimos realizar uma simulação comparativa entre RAID 5 e RAID 6. Essa proposta de simulação pretende proporcionar uma avaliação prática entre esses níveis de RAID, visando proporcionar uma compreensão mais aprofundada desses sistemas de armazenamento distribuídos. Desta forma, pretendemos contribuir no processo de tomada de decisões, buscamos fornecer uma

análise que sirva como referência na escolha do sistema RAID mais adequado às aplicações específicas dos ambientes de armazenamento de dados em consideração.

Para analisar o desempenho entre RAID 5 e RAID 6, realizaremos a simulação de um servidor de arquivo. Esse tipo de servidor é projetado para o armazenamento de dados em ambientes empresariais, permitindo que os colaboradores acessem, salvem, editem e compartilhem arquivos de forma eficiente. Essa simulação se baseia na implementação de um servidor de arquivos composto por dois conjuntos de discos configurados em RAID 5 e 6, como especificado Figura 47.

Figura 47 – Arquitetura macro da simulação de um servidor de arquivo.



fonte: Produção do próprio autor.

Em ambientes de armazenamento de dados, como RAID 5 e RAID 6, é importante adotar as métricas de desempenho eficientes e adequadas para uma análise concisa do comportamento do sistema. Neste projeto, serão avaliados alguns dos principais parâmetros de desempenho associados ao sistema de armazenamento de dados, sendo eles:

- Velocidade de gravação:
- Velocidade de leitura;
- IOPS;
- Latência.

A **velocidade de gravação**, expressa em *megabytes* por segundo (MB/s), refere-se à taxa de transferência de dados na qual informações são gravadas nos discos. Neste estudo, a velocidade de gravação indica o desempenho do sistema em operações de armazenamento de dados em discos HDD e SSD.

A **velocidade de leitura**, também expressa em megabytes por segundo (MB/s), corresponde à taxa de transferência de dados na qual as informações são lidas nos discos. Neste contexto, a análise da velocidade de leitura será usada para avaliar o desempenho do sistema em operações de leitura, ou seja, avalia a velocidade em que um arquivo é encontrado pelo sistema e enviado para o usuário daquele equipamento.

O **IOPS** (operações de entrada e saída por segundo) é uma operação de entrada/saída por segundo realizada por um servidor físico ou máquina virtual. O IOPS é usado para monitorar as operações de leitura e gravação na camada de armazenamento de blocos do sistema de um computador ou sistema de armazenamento de dados.

A **latência**, medida em milissegundos (ms), representa o tempo de resposta entre uma operação de entrada/saída (E/S). Em outras palavras, é o intervalo de tempo desde o momento que a operação se inicia até o momento da conclusão efetiva desta operação. A escolha de analisar a latência neste trabalho é fundamentada na capacidade dessa métrica proporcionar uma avaliação mais aprofundada do desempenho dos arranjos de disco em RAID 5 e 6.

Para a realização da análise de desempenho de um sistema configurado em RAID é necessário utilizar *software* especializado para fornecer relatório detalhados sobre a eficiência, confiabilidade e capacidade de recuperação dessas configurações RAID específicas. Os *softwares* utilizado nesta simulação são descritos abaixo:

- **VirtualBox - Oracle:** é um *software* de virtualização de código aberto desenvolvido e mantido pela companhia Oracle. O VirtualBox é usado para viabilizar a criação e execução de máquinas virtuais em um computador físico, possibilitando a reprodução de uma variedade de cenários e emulando condições semelhantes às de ambientes reais. Como citado na Subseção 3.2, às máquinas virtuais são simuladores de computador que operam como instâncias isoladas de sistemas operacionais dentro de um computador físico. Em sistemas com máquinas virtuais é possível criar vários ambientes virtuais utilizando recursos de uma máquina física, (VIRTUALBOX., 2023.).
- **ATTO Disk Benchmark - ATTO Technology:** é uma ferramenta de avaliação de desempenho de unidades de discos rígidos (HDD), unidades de estado sólido (SSD) e conjuntos de RAID. O ATTO Disk Benchmark foi desenvolvido pela ATTO Technology, uma empresa especializada em soluções de armazenamento, especialmente em controladores de armazenamento e *software* de gerenciamento de armazenamento. Esse *software* é usado para fornecer uma análise detalhada e precisa das características de desempenho de velocidade de gravação e leitura, (CAMARGO, A. L.; SUETH, J. M.; FONTANA, V. S., 2019).
- **Iometer - Intel:** é um programa usado para verificar a performance do sistema de armazenamento. Originalmente desenvolvida pela Intel, a ferramenta Iometer tem a finalidade de medir o desempenho de entrada/saída (E/S) de sistemas de armazenamento. Como essa ferramenta é possível medir IOPS e latência de sistemas configurados em RAID, (INTEL, 2023.).

A simulação proposta neste trabalho consiste na criação de um servidor de arquivo virtual, utilizando o programa VirtualBox para criar uma máquina virtual capaz de simular as funcionalidades e características de um servidor de arquivo. Dessa maneira, podemos explorar em detalhes como cada sistema responde em termos de desempenho e recuperação de dados em sistemas de armazenamento configurados em RAID 5 e RAID 6.

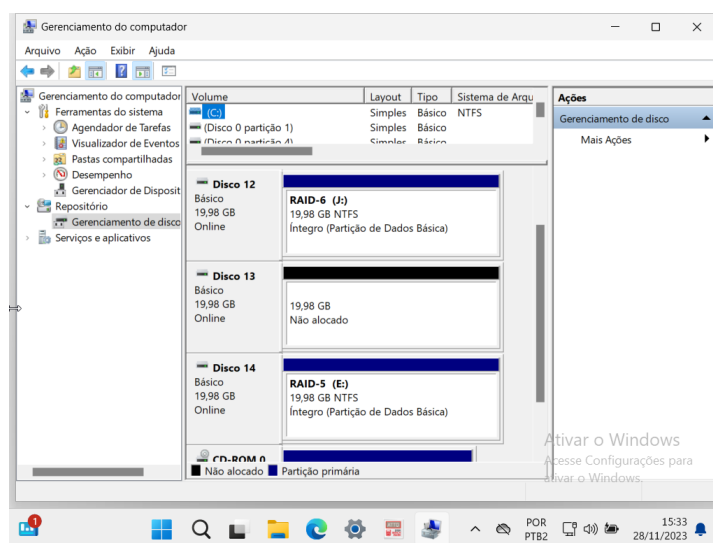
Para a implementação do servidor de arquivo no VirtualBox, utilizamos um computador físico com as seguintes especificações: um processador Intel de 8 núcleos, 8 *gigabytes* (GB) de memória de

acesso aleatório (RAM, *Random Access Memory*) e um disco SSD de 512 *gigabytes* (GB). Após a instalação do programa VirtualBox nesse *hardware*, procedemos à criação da máquina virtual com as seguintes características:

- Processador: 2 CPUs.
- Memória RAM: 4096 *megabytes* (MB).
- Armazenamento: 10 discos com 10 *gigabytes* (GB) por disco.

A criação da máquina virtual foi realizada com o objetivo de simular um servidor de arquivos, ou seja, um servidor dedicado ao armazenamento e compartilhamento de arquivos como documentos e planilhas. Após essa etapa inicial, procedemos à configuração de dois *pools* de armazenamento de dados, ou seja, dois conjuntos de dispositivos de armazenamento individuais que operam em conjunto como uma única unidade lógica. Cada conjunto de disco é formado por 5 discos individuais. O primeiro arranjo de disco, configurado em RAID 5, tem a capacidade lógica de 20 *gigabytes* (GB). Enquanto, o segundo arranjo de disco, configurado em RAID 6, possui a mesma capacidade lógica de 20 *gigabytes* (GB). A Figura 48 apresenta a tela do configurador da máquina virtual, no qual é demonstrado o volume de armazenamento criado para o RAID 5 e 6.

Figura 48 – Demonstração dos arranjos de discos configurados em RAID 5 e 6.



fonte: Produção do próprio autor.

Após a implementação dos dois arranjos de discos, o próximo passo consistiu em obter dados para serem gravados em ambos os conjuntos de discos. Para isso, utilizamos os dados públicos disponíveis no portal Dados Abertos do Governo Federal e no site do IBGE (Instituto Brasileiro de Geografia e Estatística). No portal Dados Abertos do Governo Federal, os dados coletados incluem áreas como governança de tecnologia da informação, informações educacionais e dados estratégicos (Governo Federal., Acesso em: 9 de novembro de 2023). Já no site do IBGE, os dados obtidos estão relacionados a pesquisas e estudos sobre geociências, estatísticas sociais e estatísticas econômicas (IBGE., Acesso em: 10 de novembro de 2023).

Posterior à coleta dos arquivos, organizamos as informações em duas pastas de arquivos distintas. A pasta 1 corresponde aos dados obtidos no portal Dados Abertos do Governo Federal e contém 2388 arquivos, totalizando uma volumetria de dados de 2,1 *gigabytes* (GB). Enquanto isso, a pasta 2, reservada para os dados obtidos no portal IBGE, essa pasta contém 1824 arquivos, resultando em uma capacidade de 2,9 *gigabytes* (GB). No total foram gravados 5 *gigabytes* (GB) de dados em cada arranjo de disco configurados com uma capacidade lógica de 20 *gigabytes* (GB). Devido à baixa capacidade dos arquivos encontrados em ambos os sites, optamos por criar cópias dos arquivos baixados, em cada pasta de arquivo, para alcançar uma capacidade de dados mais significativa, visando obter uma melhor análise dos comportamentos do RAID 5 e 6.

Com a conclusão da organização das pastas de arquivos e a gravação dessas pastas nos arranjos de discos, foram realizadas simulações detalhadas para medir o desempenho de cada arranjo configurado.

Para observar a taxa de transferência de gravação e leitura, analisamos as velocidades de gravação e leitura, por meio do *software* ATTO Disk Benchmark, uma ferramenta especializada na medição da taxa de transferência leitura/gravação em conjuntos de RAID. Para utilizar essa ferramenta, tivemos que fazer *download*). O funcionamento do ATTO Disk Benchmark é simples, ele testa a velocidade de leitura e gravação de disco rígido em diferentes tamanhos de bloco de dados. Ao iniciar a ferramenta, escolhemos o conjunto de disco em RAID 5 e configuramos a ferramenta com uma faixa de tamanho de transferência variando de  $2^n$  para  $n = (3, 4, \dots, 8)$ . Os resultados exibem as velocidades de gravação (*write*) e de leitura (*read*). Além disso, o *software* exibe um gráfico referente a velocidade de transferência em diferentes blocos.

Para avaliar a taxa de transferência de gravação e leitura, utilizamos o ATTO Disk Benchmark, uma ferramenta especializada em medir essas taxas em conjuntos de RAID. Após o *download* e instalação da ferramenta na máquina virtual, configuramos os testes para avaliar conjuntos de discos em RAID 5 e, posteriormente, RAID 6. Durante os testes, ajustamos a faixa de tamanho de transferência para abranger diferentes potências de 2, variando de  $2^n$  para  $n = (3, 4, \dots, 8)$ . Os resultados foram apresentados em uma tabela e em um gráfico, destacando as velocidades de gravação e leitura para os diversos tamanhos de bloco testados. Os resultados destas simulações estão disponíveis na Seção 5.2.

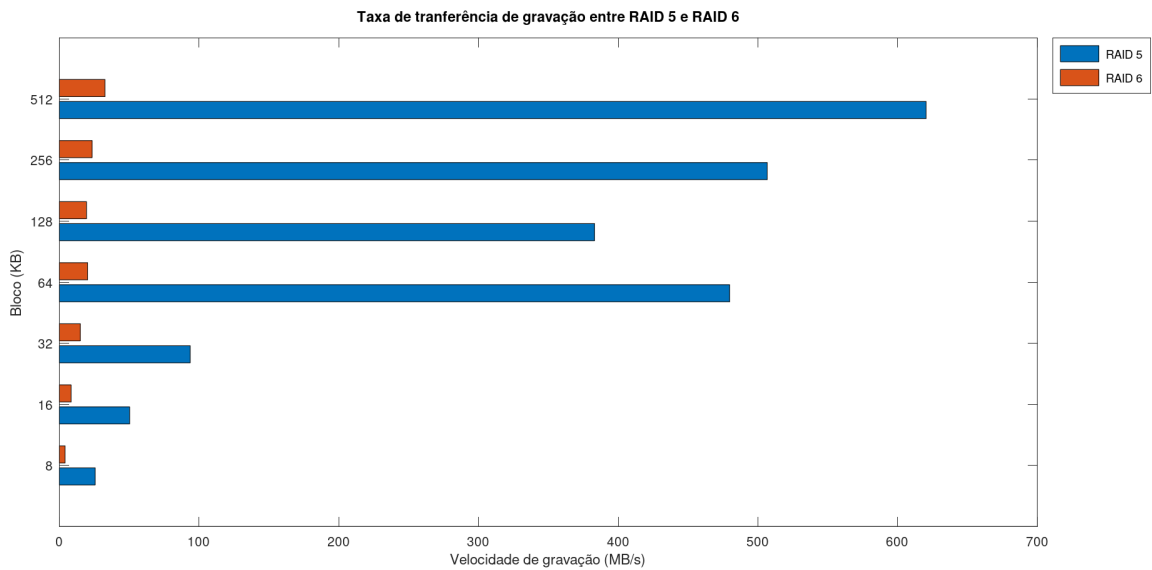
Para analisar o desempenho dos sistemas em termos de IOPS e latência, utilizamos o Iometer, um *software* de gerenciamento de disco focado no monitoramento do desempenho de sistemas de armazenamento distribuído. Utilizamos essa ferramenta para medir IOPS e a latência média do sistema, embora ela ofereça aos usuários inúmeras métricas para avaliar o desempenho. Para utilizar essa ferramenta, realizamos o *download* do Iometer e procedemos à instalação. Em seguida, configuramos a ferramenta por meio da seção "Destinos de Disco" (*Disk Targets*), na qual selecionamos os conjuntos de discos em RAID 5 e 6. Posteriormente, ajustamos as configurações específicas utilizando a opção "Acesso às Especificações" (*Access Specifications*), onde definimos a quantidade de simulações e outras configurações gerais. Com as configurações concluídas, iniciamos os testes. A ferramenta gera um relatório abrangente que inclui os resultados da análise IOPS, latência e outras métricas de desempenho. Os resultados da simulação serão discutidos na Seção 5.2, fornecendo uma compreensão detalhada do desempenho dos conjuntos de discos em RAID 5 e 6.

## 5.2 RESULTADOS DA SIMULAÇÃO

A proposta de análise de desempenho de um servidor de arquivo consistiu na criação de uma máquina virtual configurada com dois arranjos de discos em RAID 5 e 6. Em cada conjunto de disco, foram armazenados um total de 5 *gigabytes* (GB) de dados, considerando as duas pastas de arquivo.

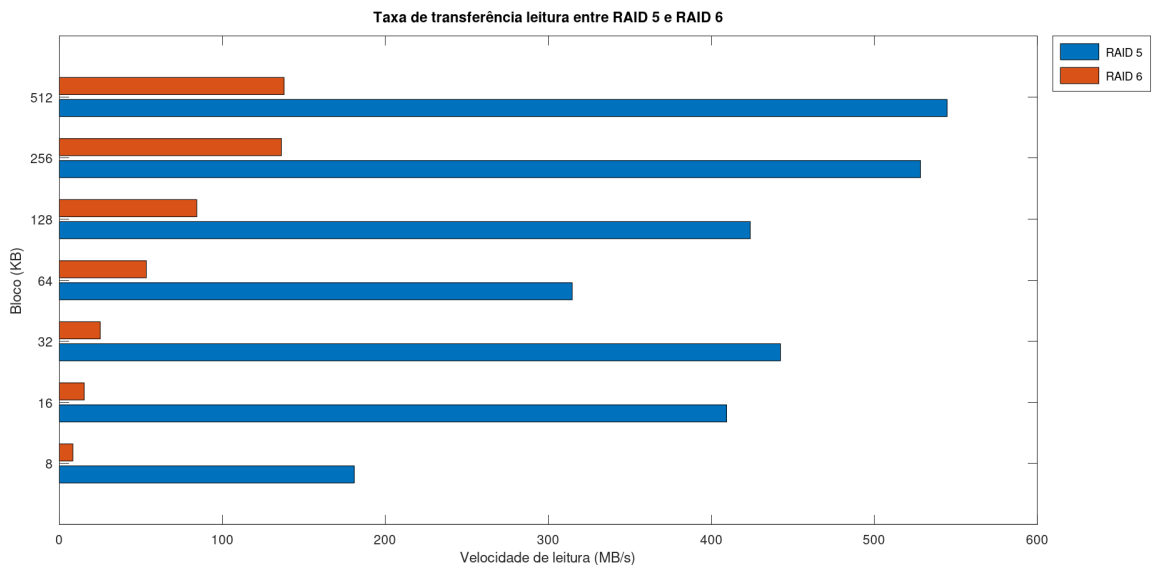
Os resultados da simulação da taxa de transferência de gravação e leitura, obtidos por meio da ferramenta ATTO Disk Benchmark, podem ser visualizados nas Figura 49 e Figura 50. Em ambos gráficos, analisamos o parâmetro de velocidade de gravação e leitura em blocos de dados com tamanhos variando de  $2^n$  para  $n = (3, 4, \dots, 8)$ . Ambos os gráficos foram gerados utilizando o *software* Octave.

Figura 49 – Resultados: taxa de transferência de gravação.



fonte: Produção do próprio autor.

Figura 50 – :Resultados: taxa de transferência de gravação.



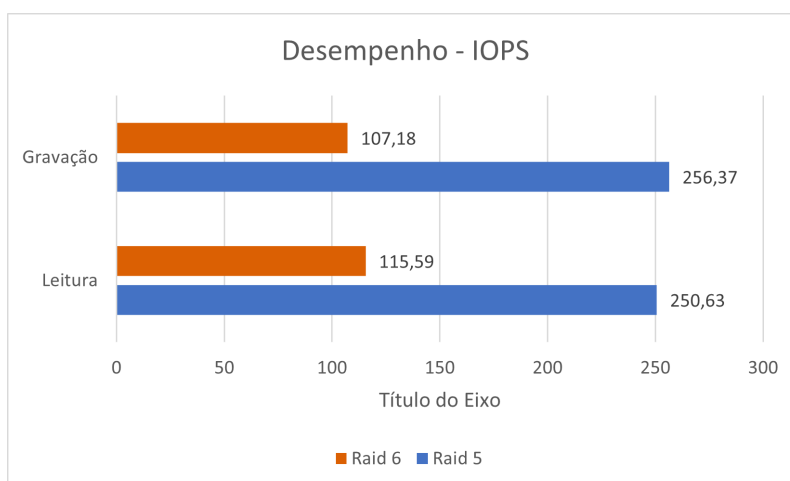
fonte: Produção do próprio autor.

Ao analisar o gráfico da taxa de transferência de gravação ilustrado na Figura 49, é possível observar que a velocidade de gravação do arranjo configurado em RAID 5 demonstra uma taxa de transferência melhor que o arranjo de disco configurado em RAID 6. Essas diferenças de comportamento ocorrem devido à simplicidade do cálculo de paridade no RAID 5 em comparação com os cálculos mais complexos de paridade realizados no RAID 6, o que impacta diretamente na velocidade de gravação e leitura do sistema. Os resultados de desempenho, referentes à taxa de transferência de leitura, apresentados na Seção Figura 50, indicam que o RAID 5 possui uma melhor taxa de transferência de leitura em comparação com o RAID 6.

Outra métrica importante a ser considerada na análise de desempenho durante a simulação de um servidor de arquivos virtual configurado com dois conjuntos de 5 discos em RAID 5 e 6 é o número de operações de entrada/saída por segundo (IOPS, *input/output operations per second*). Este indicador de desempenho tem a função de avaliar a eficiência e a capacidade de resposta do sistema de armazenamento, influenciando diretamente a dinâmica do acesso aos dados. Para analisar IOPS desses conjuntos de dados, foram feitas 5 testes através da ferramenta Iometer.

Os resultados obtidos na simulação de IOPS, apresentado na Figura 51, comprovam que sistema de armazenamento configurado em RAID 5, em comparação com o RAID 6, demonstra ser um sistema com uma maior capacidade de armazenar e processar. Isso ocorre, pois o RAID 5 apresenta uma taxa de operações de entrada/saída por segundo maior que as taxas geradas pelo RAID 6. Dessa forma, pode-se afirmar que o RAID 5 é um método mais eficiente em termos de desempenho do que o RAID 6, tanto em relação as operações de leitura quanto as operações de gravação.

Figura 51 – Resultados: IOPS.

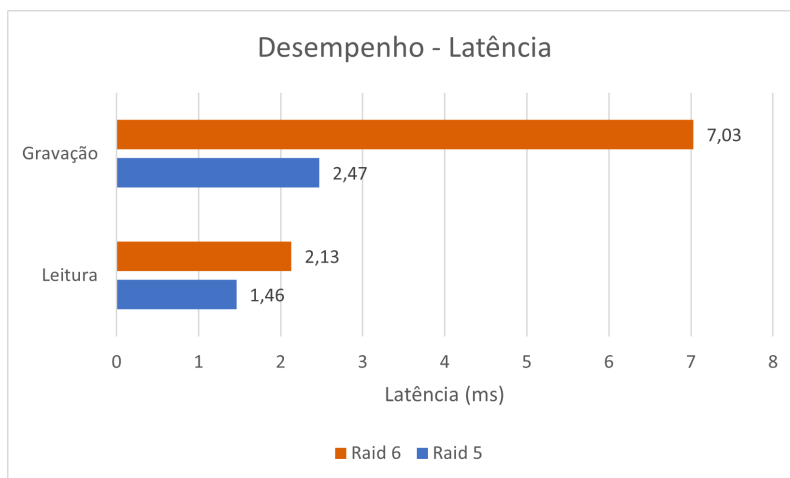


fonte: Produção do próprio autor.

Com base nos resultados apresentados na Figura 52, o RAID 5 demonstrou ser mais eficiente que o RAID 6 em termos de desempenho de leitura e gravação. A latência de gravação apresentou uma maior diferença de desempenho entre os dois níveis de RAID. Isso ocorre devido à utilização de dupla paridade nos sistemas configurados em RAID 6, o que implica em cálculos adicionais durante as operações de gravação, resultando em uma latência (tempo de resposta de gravação) superior quando comparado ao RAID 5. Apesar dessa pequena diferença nos valores de latência de leitura entre RAID

5 e RAID 6, é importante destacar que o RAID 5 demonstrou um melhor desempenho em relação à latência de leitura quando comparado ao RAID 6.

Figura 52 – Resultados: latência média.



fonte: Produção do próprio autor.

## 6 CONCLUSÃO

O presente trabalho de conclusão de curso apresentou um estudo sobre os principais conceitos relacionados ao RAID, destacando as estratégias de paridade aplicadas em alguns níveis de RAID por meio do código de Hamming, códigos matriciais e de paridade, código RDP e código EVENODD. Neste trabalho, também foram realizadas análises de desempenho para avaliação comparativa dos sistemas de RAID abordados. Além disso, uma simulação prática foi realizada para avaliar o desempenho de um sistema de armazenamento distribuído ao configurar conjuntos de discos em RAID 5 e 6.

Como foi apresentado, o RAID é um importante método para a prevenção e recuperação de dados usado nos sistemas de armazenamento distribuído. Cada nível de RAID proporciona benefícios específicos. O RAID 0 aplicado em situações que exigem um desempenho elevado, sem a necessidade de alta disponibilidade de dados. Por outro lado, o RAID 1 é indicado para aplicações que requerem uma alta disponibilidade de dados e necessitem de poucos discos, pois o RAID 1 pode ser implementado em um conjunto com no mínimo dois discos. O RAID 5 é um dos principais modelos de RAID aplicado em servidores. Isso ocorre devido à sua capacidade de proporcionar capacidade e confiabilidade ao sistema. O RAID 5 é usado para acrescentar redundância aos conjuntos de discos HDD ou SSD, devido a paridade obtidas por meio de códigos matriciais e de paridade diagonal, onde o cálculo da paridade é realizado por operações de XOR. Por outro lado, o RAID 6 oferece um nível mais elevado de tolerância a falhas alcançado por meio da utilização de códigos como RDP e EVENODD, que oferecem tolerância a falhas duplas. O RAID 6 é indicado para ambientes críticos ou que fazem uso de discos rígidos com tecnologia de 7200 rotações por minuto (RPM, *revolutions per minute*) os quais são mais propensos a falhas.

A simulação proposta contribuiu para uma avaliação prática e comparativa entre dois dos níveis mais utilizados no mercado, o RAID 5 e 6. A análise comprovou que o RAID 5 apresenta melhores taxas de desempenho em termos de velocidade e latência de gravação. Isso se deve ao fato de que o conjunto de discos configurado em RAID 6 perde desempenho durante o cálculo e gravação dos bits de paridade dupla, ao contrário do RAID 5, onde a paridade é simples. Ao observar os parâmetros de leitura, notamos que os resultados entre RAID 5 e 6 geralmente são semelhantes. É fundamental destacar que o tipo de disco influencia o desempenho do sistema de armazenamento distribuído. Os discos SSD oferecem melhor desempenho em comparação com os discos HDD, embora os SSDs apresentem um custo superior. Portanto, além do tipo de RAID, também é essencial levar em consideração o tipo de disco usado na configuração do RAID para atender às expectativas do projeto tanto no âmbito técnico quanto financeiro.

Para continuação do trabalho, é possível fazer a análise do desempenho dos níveis de RAID quando aplicados a sistemas de armazenamento distribuídos compostos por unidades discos rígidos e unidades de estado sólido. A investigação desses cenários proporcionará uma compreensão mais abrangente das complexidades de desempenho em ambientes de armazenamento variados. Adicionalmente, é possível desenvolver um protocolo específico para avaliar a eficácia da recuperação de dados em níveis de RAID que fazem uso de técnicas de paridade. Este protocolo pode ser projetado para examinar

como esses sistemas respondem em diferentes situações, como falhas de disco, corrupção de dados e outros cenários de perda de informações.

## REFERÊNCIAS

- SHVIDKIY, A. A.; SABELIEVA, A. A.; ZARUBIN, A. A. Caching methods analysis for improving distributed storage systems performance. Systems of Signal Synchronization, Generating and Processing in Telecommunications (SYNCHROINFO), Kaliningrad, Rússia, 2021.
- ANCIOTO, A. S. R. Simulador para ensino de algoritmos de escalonamento de discos. UNIFACCAMP – Centro Universitário Campo Limpo Paulista., Campo Limpo Paulista, SP - Brasil, 2018.
- ARAUJO, A. C. M.; TEIXEIRA, J. F.; SANTOS, G. T. Sinais de dados em telecomunicações: Detecção e correção de erros. UFSC - Universidade Federal de Santa Catarina, 2020.
- CALDEIRA, J.; MATOS, R. A. O. Códigos corretores de erros e uma breve introdução aos códigos de hamming. Revista v.16.n.6, Curitiba, PR - Brasil, 2023.
- CAMARGO, A. L.; SUETH, J. M.; FONTANA, V. S. Analise de desempenho de sistema operacional por meio das ferramentas bootracer, performance test, atto hd e super pi. Instituto de Ensino Superior Do Espírito Santo, 2019.
- CARVALHO, A. V. F. **SSD vs HDD: Um Estudo sobre a viabilidade e os impactos do uso de SSDs como dispositivo de armazenamento secundário em detrimento dos HDDs.** Picos, PI- Brasil: Universidade Federal do Piauí., 2014.
- CHEN, Y. . Efficiency comparison of row-diagonal parity and evenodd encoded check disk repair algorithms. nternational Symposium on Advances in Informatics, Electronics and Education (ISAIEE), Frankfurt, Alemanha, 2022.
- DELGADO, J.; RIBEIRO, C. Arquitetura de computadores. Grupo GEN, Rio de Janeiro, RJ - Brasi, v. 5. ed., 2017.
- Governo Federal. **Dados Abertos.** Acesso em: 9 de novembro de 2023.
- HANXU, H.; YUNGHSIANG, H. S.; KENNETH, S. W.; HUI, L. A unified form of evenodd and rdp codes and their efficient decoding. IEEE Transactions on Communications, 2018.
- IBGE. **Instituto Brasileiro de Geografia e Estatística.** Acesso em: 10 de novembro de 2023.
- INTEL. **Iometer.** 2023. Disponível em: <<https://iometer.org/>>.
- KIM, J.; YU, J. H.; KANG, H.; SHIN, J. H.; JEONG, H.; NOH, S. Y. . Performance analysis of distributed file system based on raid storage for tapeless storage. IEEE Access , vol. 11, pp., Rio de Janeiro, RJ - Brasil, 2023.
- KONDO, E. Estudo comparativo de algoritmos de ecc aplicados à memória nand flash. UNISINOS - Universidade do Vale do Rio dos Sinos, São Leopoldo,RS - Brasil, 2017.
- LEE, K.; YU, G.; HWANG, Y.; JUN, B.; SON, H.; SONG, Y. H. . Ecc-aided raid for reliability improvement of ssd. IEEE Global Communications Conference, Rio de Janeiro, RJ - Brasil, 2022.
- LIMA, A. C. Segurança de dados e big data. Editora Senac, São Paulo, SP - Brasil, v. 1. ed., 2021.
- MAZIERO, C. A. Sistemas operacionais: Conceitos e mecanismos. UFPR – UNIVERSIDADE FEDERAL DO PARANÁ., Curitiba, Paraná - Brasil, 2019.

- MUNHOS, F. L. O. Arquitetura de solid state driver (ssd)s. Universidade Estadual Paulista – Júlio de Mesquita Filho, 2018.
- OLIVEIRA, G. Vinícius; Correa, Ricardo. **Tecnologias de Discos Magnéticos**. Campinas, SP - Brasil: UNICAMP - Universidade de Campinas, 2021.
- PATTERSON, D. A.; GIBSON, G.; KATZ, R. H. . A case for redundant arrays of inexpensive disks(raid). University of California, Berkeley, California, 1988.
- PERTIN, D.; KEMPEN, A.; PARREIN, B.; NORMAND, N. Comparison of raid 6 erasure codes. Universite de Nantes, Nantes, França, 2015.
- PIONTKOSKI, Gilvano. Comparativo de desempenho de consultas sql entre banco de dados em memória ram e em ssd. UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ, Pato Branco, PR - Brasil, 2017.
- REINSEL, David. GANTZ, John. RYDNING, John. The digitization of the world from edge to core. <https://www.seagate.com/files/www-content/our-story/trends/files/idc-seagate-dataage-whitepaper.pdf>, 2018.
- RIBEIRO, M. P. Informação e codificação: Conceitos básicos para a comunicação digital - codificação de canal. Pearson,, Rio de Janeiro, RJ - BR, v. 1. ed., 2021.
- RYAN, W. E.; LIN, S. Channel codes: Classical and modern. Cambridge University Press, New York, EUA, v. 1. ed., 2009.
- SAINI, M. L.; SHARMA, V. K.; KUMAR, A. An efficient single and double error correcting block codes with low redundancy for digital communications. International Conference on Advancement in Computation and Computer Technologies (InCACCT)., Gharuan, Índia, 2023.
- SILBERSCHATZ, A. Fundamentos de sistemas operacionais. Grupo GEN, Rio de Janeiro, RJ - Brasil, v. 9. ed., 2015.
- SOMASUNDARAM G.; SHRIVASTAVA, A.; FERNANDES, A. P. . Armazenamento e gerenciamento de informações. Grupo A, Rio de Janeiro, RJ - Brasil, v. 1. ed., 2011.
- STAIR, R. M.; REYNOLDS, G. W.; Brayant, R. Princípios de sistemas de informação. Cengage Learning, São Paulo, SP - Brasil, v. 4. ed., 2021.
- STALLINGS, W. Arquitetura e organização de computadores. Pearson, São Paulo - Brasil, v. 10. ed., 2017.
- TANENBAUM, A. S.; STEVEN, M. V. Sistemas distribuídos: princípios e paradigmas. Pearson, São Paulo, SP - Brasil, v. 2. ed., 2007.
- TANENBAUM, A. S.; Woodhul, A. S. Common raid disk data format specification. Pearson,, Porto Alegre, RS - Brasil, v. 3. ed., 2009.
- THOMASIAN, A. Mirrored and hybrid disk arrays: Organization, scheduling, reliability, and performance. gabi, 2018.
- VIRTUALBOX. **VIRTUALBOX**. 2023. Disponível em: <<https://www.virtualbox.org/>>.
- WU, R.; WU, Y.; WANG, M.; J WANG, L. . An efficient raid6 system based on xor accelerator. International Conference on Computer Communication and the Internet (ICCCI), Nagoya, Japão, 2021.

XIANG, L.; XU, Y.; LUI, J.C.S.; CHANG, Q. A hybrid approach to failed disk recovery using raid-6 codes: Algorithms and performance evaluation. University of Science and Technology of China, New York, EUA, 2021.