UNIVERSIDADE FEDERAL DA GRANDE DOURADOS - UFGD

Anderson Lopes de Barros

A Álgebra dos Códigos Corretores de Erros

Dourados - MS

2019

Anderson Lopes de Barros

A Álgebra dos Códigos Corretores de Erros

Dissertação apresentada ao final do Programa de Mestrado Profissional em Matemática em Rede Nacional - PROFMAT da Universidade Federal da Grande Dourados - UFGD como requisito parcial para obtenção do grau de Mestre em Matemática.

Universidade Federal da Grande Dourados - UFGD

Faculdade de Ciências Exatas e Tecnologia

Orientador: Prof[a]. Dr[a]. Ana Cláudia Machado Mendonça

Dourados - MS

2019

Dados Internacionais de Catalogação na Publicação (CIP).

B277á Barros, Anderson Lopes De

A Álgebra dos Códigos Corretores de Erros [recurso eletrônico] / Anderson Lopes De Barros. -- 2019.

Arquivo em formato pdf.

Orientadora: Ana Claudia Machado Mendonça Chagas.

Dissertação (Mestrado em Matemática)-Universidade Federal da Grande Dourados, 2019.

Disponível no Repositório Institucional da UFGD em:
https://portal.ufgd.edu.br/setor/biblioteca/repositorio

1. Códigos. 2. Paridade. 3. Vetor. 4. Matriz Geradora. 5. Código de Hamming. I. Chagas, Ana Claudia Machado Mendonça. II. Título.

Ficha catalográfica elaborada automaticamente de acordo com os dados fornecidos pelo(a) autor(a).

MINISTÉRIO DA EDUCAÇÃO
FUNDAÇÃO UNIVERSIDADE FEDERAL DA GRANDE DOURADOS
FACULDADE DE CIÊNCIAS EXATAS E TECNOLOGIA
PROGRAMA DE MESTRADO PROFISSIONAL EM MATEMÁTICA EM REDE NACIONAL - PROFMAT

Termo de Aprovação

Após a apresentação, arguição e apreciação pela banca examinadora, foi emitido o parecer APROVADO, para a dissertação intitulada: **"A Álgebra dos Códigos Corretores de Erros"**, de autoria de **Anderson Lopes de Barros**, apresentada ao Programa de Mestrado Profissional em Matemática da Universidade Federal da Grande Dourados.

Profa. Dra. Ana Cláudia Machado Mendonça Chagas
Presidente da Banca Examinadora

Profa. Dra. Irene Magalhães Craveiro
Membro Examinador (UFTM)

Prof. Dr. Jaime Rezende de Moraes
Membro Examinador (UEMS)

Dourados/MS, 26 de fevereiro de 2019

Dedicado a:

Meus Pais, João e Elizabethe, os quais sempre me disseram que estudo é ferramenta de mudança de realidade e apoiaram desde os primeiros passos, além do meu irmão, Alexsandro, que torcem e comemoram por cada passo dado;

Meus Primeiros professores, com os quais me fizeram ter o gosto pela matemática;

As pessoas que aparecem em nosso caminho da escola pública ao mestrado, e que estão lá ajudando e incentivando a não desistir dos objetivos.

Agradecimentos

Agradecer é um hábito que aprendemos desde pequenos, mas que a vida com seus percalços e dinâmicas nos forçam a fazer com menos frequência. Revisitar todos aqueles que fazem e fizeram parte da nossa vida e demonstrar gratidão, é uma qualidade que precisamos exercitar e por que não, incentivar.

Nessa caminhada, do tempo que tomei gosto pela matemática até este momento, muitas dificuldades, amarguras, decepções e desapontamentos já atormentaram os pensamentos, mas é graças a ajuda de pessoas que podemos nos reerguer, lutar e superar desafios.

Dessas diversas pessoas que entram na nossa vida e caminham conosco, o meu agradecimento a algumas delas e outras tantas que participaram da minha caminhada. Primeiramente, a Deus, pelas oportunidades oferecidas e pela vontade e ânimo que disponibiliza a mim, mesmo passando pelas mais diversas e tristes situações desde os tempos de Ensino Médio (as quais, alias, alguns de meus orientadores e colegas conheceram bem).

A meus pais, João(que me observa lá de cima), Elizabethe, minha mãe, parceira e confidente de todos os momentos, que garante que eu seja teimoso e batalhador pelas coisas que acreditamos e a meu irmão Alexsandro, que ainda tem muito a caminhar, mas que torce junto com meus familiares.

A minha família, em especial, aos tios Ademir e Valéria, de Presidente Prudente, que observam a distância parte dessa grande viagem que é de deslocar a dourados e ouviram bastante histórias sobre as coisas ocorridas nesse caminho.

Aos Amigos, professores dos tempos de escola e aqueles que acreditam e torcem por mim, lá de Euclides da Cunha, dentre eles em especial Carmelita Matias e Eulides Nicácio, Bete Calegari, Yolanda Sales e Suerley Negrão, que de certa forma atuaram na minha história no Ensino Básico e a entrada na Graduação. Também, aos amigos com os quais trabalho, colegas de auxílio nos desafios diários, Tchuska, Lurdinha,Vera, Luciana, Cleide, Karla, Tereza, entre tantos outros.

Das graduações até o acesso ao PROFMAT, agradeço àqueles que caminharam comigo nestes cursos e aos professores que se tornaram colegas nesse período, nomes que são tantos, mas citando nominalmente podem fazer deixar alguem de fora. Também, aos professores que acabaram por se tornarem colegas durante este período. Faço aqui também um agradecimento a Alessandra Dan, que junto comigo se aventurou nessa caminhada que foi realizar o mestrado, participando e inscrevendo-se no processo seletivo.

Um agradecimento aos meus colegas que me acompanharam e se fizeram presentes nesses tempos de PROFMAT(turma 2016), dos quais Anderson, Beatriz, Eduarda, Miquéias, Naiguiel, Rodrigo e Viviane. Estendo um especial agradecimento aos colegas Edvair, Éder e Katiuce (pelas viagens, e diversas histórias vividas durante todo esse tempo, além do auxílio no latex) além da Márcia(que além de dedicação extrema e fornecedora dos cafés que tanto tomei, é de uma tranquilidade e paciência que é um exemplo a ser seguido, sem contar o fato de que ainda me aconselhou bastante durante o preparo desta obra).

Finalizando os agradecimentos, expressando gratidão e parabenizando aos professores do PROFMAT, que entre conselhos, broncas e listas de exercícios, mostram que o mestrado exige dedicação, estudo e muitos cafés nas madrugadas, sempre incentivando-nos e não deixando desanimar-nos.

Por último, mas com o alto grau de importância que se dedicam a tarefa de ensinar, agradeço as minhas orientadoras (sim, são duas), Ana Cláudia, pelo suporte e auxílio que me dá desde o momento da definição do tema deste trabalho, incluindo os seminários mais intensivos nessa fase final, e a professora Irene Craveiro, nossa coordenadora, que desde o início da turma esteve acompanhando o passo a passo de todos os alunos, orientando, nos fazendo tomar ideia da realidade, puxando nossa orelha quando necessário, sempre vislumbrando a conclusão de nossos esforços. São dois "Anjos"pelos quais sou muito grato por todas as oportunidades vividas, pelo tempo que cederam e se dedicaram aos meus estudos. São pessoas como todos vocês que marcam as nossas vidas e fazem com que eu reveja o passado e possa dizer com satisfação: "A MATEMÁTICA MUDA VIDAS, e vale a pena lutar por essas oportunidades"

Muitíssimo Obrigado!

Dessa aventura, levo amigos pra vida toda!! É muito bom poder contar com todos vocês!!

"O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior Brasil (CAPES) - Código de Financiamento 001"

"Tudo o que um sonho precisa para ser realizado é alguém que acredite que ele possa ser realizado."

Roberto Shinyashiki.

Resumo

O trabalho com códigos corretores de erros é de grande utilidade em nossos dias, não somente para a avaliação de esquemas de identificação, como também nas transmissões e envio de diversos tipos de mídia utilizando os meios digitais. O que muitos não sabem é que a implementação desses códigos e os algoritmos que permitem a segurança ou ainda a recuperação destas informações utilizam esquemas de álgebra abstrata, conteúdo não trabalhado em sala de aula. O escopo desse trabalho é falar sobre a álgebra escondida no uso dos códigos e mostrar suas aplicações no dia a dia, apresentando um pouco sobre a teoria do códigos corretores de erros e mostrar sua aplicabilidade em situações do cotidiano, sugerindo um plano de trabalho que pode ser apresentado nas séries finais do Ensino Fundamental, realizando um paralelo com os conteúdos trabalhados no meio acadêmico, tais como relacionados a matrizes, vetores, paridade e algoritmos digitais.

Palavras-chaves: Códigos, paridade, matriz geradora, código de Hamming.

Abstract

The work related to error-correction codes is of great use today, not only for the validation of identification schemes, but also for the transmission and sending of various kinds of media using digital media. What many people do not know is that the implementation of these codes and the algorithmus that allow the security or even the retrieval of this information use abstract algebra schemes, unworked content in our graduation.The scope of this work is to present the hidden algebra in the use of codes and to show their applications on a day by day, presenting a bit about the theory of error-correcting codes and showing their applicability in everyday situations, suggesting a work plan that can to be presented in the final series of Elementary School, making a parallel with the contents worked in the academic environment such as the related to matrices, vectors, parity and digital algorithms.

Key-words: Codes, parity, generating matrix, Hamming code.

Lista de ilustrações

Lista de tabelas

Sumário

1 Introdução

Em nossas escolas, o desafio no ensino de matemática é provar, por meio de generalizações as estruturas e relações que conjuntos numéricos possuem. Algebrizar operações envolvendo esses números e as propriedades que apresentam é algo que pode ser elevado a um nível alto e tão importante que diversas ciências (além da matemática) se utilizam dela.

A álgebra que denominamos "Abstrata" é utilizada para complementar a álgebra elementar estudada no colégio, na qual são abordadas regras para manipular (somar, multiplicar, etc) expressões algébricas em que aparecem variáveis e números reais ou complexos. As ideias que este ramo da matemática trazem permitem estudar propriedades e padrões que distintos conceitos de matemática tem em comum e com a ajuda desses conceitos entender as operações e aplicações em áreas como a informática, por exemplo.

Parte-se da ideia de que, ao agrupar as coleções em conjuntos, existem operações que são comuns a elas e obedecem a estruturas. Assim, a algébra abstrata levou em conta que temos um conjunto númerico com elementos, que obedecendo uma operação binária resulta num novo elemento.

Como citado em (DIAS, 1994)

> "No novo cálculo se faz necessário instituir a relação de identidade; sem a qual não se poderia comparar as classes originárias com outras delas derivadas pelas operações basilares de . e +. Portanto, tome-se o símbolo = para identificá-la. Ou seja, considerando-se o símbolo = entre os símbolos que designam duas classes quaisquer (x e y, por exemplo) está a indicar-se que as classes têm os mesmos membros e denota-se: $x = y$... apresente-se a definição de sistema algébrico ou de álgebra abstrata; isto é, denominam-se Álgebra Abstrata ou Sistema Algébrico a um conjunto não vazio munido de um ou mais operadores binários sobre ele definidos. Logo, designado por A o conjunto em questão, tem-se $(A,*,\#)$ indica uma álgebra com dois operadores"

A partir dessa definição, como ponto de partida, é possível entender que ela se estrutura em propriedades e demonstrações as quais discorremos no capítulo seguinte. Para se ter uma ideia da importância destes conceitos, uma aplicação prática de álgebra abstrata se faz presente quando abordamos códigos corretores e erros, uma vez que, por meio de teorias, temos um conjunto de " palavras", fundada em conjuntos numéricos, que obedecem as estruturas que se realizam por meio de operação entre elementos desse conjunto.

Nas comunicações, tal importância se faz ainda mas presente, seja pela questão da segurança/confiabilidade, seja pela integridade da informação enviada/lida. O envio de

uma mensagem implica que ela seja clara, coesa e organizada segundo o código padrão (alfabeto/idioma). Quando comunicamos nos meios digitais, onde máquinas se utilizam de códigos numéricos, é comum que esta mensagem seja enviada de forma redundante, e que máquinas no emissor e no destinátario se utilizam de algoritmos para desvelar a mensagem enviada e tentar corrigi-la em caso de erros.

Os trabalhos ce Hamming, Galoy e Shannon permitiram que a partir da década de 40, a comunicação por meios digitais desse um salto em eficiência e qualidade na transmissão, e seus estudos são utilizados até hoje.

O que chamamos de Teoria da informação, na area de comunicações, é na verdade, uma aplicação matemática e algébrica dos códigos Corretores de erros, que abordaremos neste trabalho.

No capítulo 2, compreenderemos o processo de determinação das classes residuais, para a construção de um código sobre um corpo finito. O conjunto escolhido foi o conjunto $\mathbb{Z}_p$, com p primo, o qual provaremos que é um corpo finito. Finalizando o estudo do capítulo, apresentamos algumas aplicações práticas da congruência módulo m na validação dos códigos como o CPF e da nota de Euro, por meio do chamado dígito de verificação (DV).

No capítulo 3, estudaremos o espaço vetorial finito $\mathbb{Z}_p^n$. Sua importância é pelo fato que as "palavras" desse código podem ser entendidas como um vetores de coordenadas em $\mathbb{Z}_p$.

No capítulo 4, são apresentadas as ideias de criação e interpretação dos códigos lineares, assim como as matrizes responsáveis pela codificação e decodificação.

Por fim, no último capítulo, entendido o processo de criação e as palavras do código, já será possível entender com mais clareza o código de Hamming. Concluindo, daremos sugestões para um plano de aula no Ensino Fundamental, contemplando alguns conceitos associados aos códigos e às impressões sobre a álgebra presente neles, assim como o porquê de se abordar, em cada nível de ensino o uso dessa álgebra.

Como se vê, a álgebra dos códigos de erros é um campo vasto que parte das operações vistas em estruturas algébricas nos campos acadêmicos, mas não se restringem somente a teorias. A álgebra dos códigos pode ser trabalhada em outras modalidades do ensino básico até mesmo para mostrar sua importância e aplicabilidade.

2 O Anel das Classes Residuais

Na teoria nos números, independente para qual quantidade ou contagem que se faça, temos na beleza da matemática a ideia de que todo o número tem característica que os tornam próprios e especiais. Sejam primos, quadrados, ímpares, perfeitos, compostos, inteiros ou racionais, o que temos na verdade é que eles possuem características que permitem agrupá-los em conjuntos ou observar suas propriedades.

A teoria dos números é a área da matemática cujo objetivo é descobrir e estabelecer as relações profundas e sutis que números de tipos diferentes guardam entre si e ao longo da história, diversos pensadores, filósofos ou essencialmente matemáticos fizeram com que essa evoluísse, criando e provando teoremas dos mais diversos. Desde Pitágoras, Euclides, Fermat, Gauss, entre outros, foi possível entender as relações que alguns números possuem e até mesmo facilitar cálculos.

Gauss, em 1801, com 24 anos de idade, estabeleceu um novo conceito: o de congruência

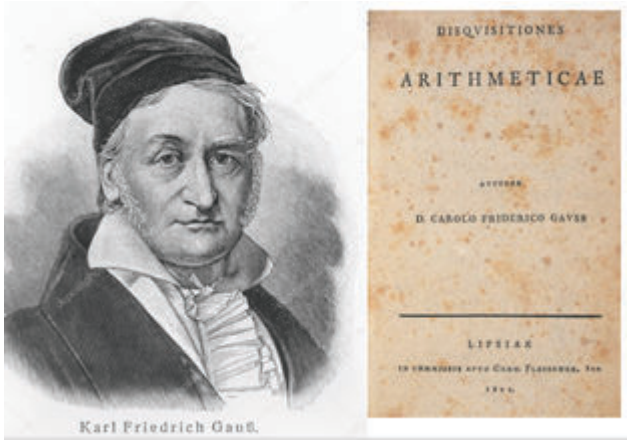


Figura 2.1 – Gauss e sua obra de referência para a teoria dos números

Fonte: autor/banco de imagens-internet

No capítulo destinado ao tema em seu livro, ele anuncia, de modo prático que:

Definição 2.1. *Se um número m divide a diferença $a - b$ ou $(b - a)$ de dois números inteiros a e b, então dizemos que a é congruente a b módulo m e denota-se*

$$a \equiv b \, mod \, m$$

O símbolo $\equiv$ é de importante contribuição, pois realiza uma analogia entre congruência e igualdades.

A congruência satisfaz as propriedades reflexiva, simétrica e transitiva. Ao enunciar essas teorias, e suas decorrentes propriedades para o conjunto dos números inteiros é definido o anel do conjunto $\mathbb{Z}_m$, formado pelas classes residuais módulo m, objeto de nosso capítulo. O desenvolvimento dos resultados, conceitos e exemplos desse capítulo foram baseados no livro (HEFEZ, 2008) .

2.1 O conjunto das classes residuais

O conjunto do números inteiros $\mathbb{Z}$ pode ser particionado em subconjuntos da seguinte maneira: Dado $x \in \mathbb{Z}$, $m > 1$, para todo $a \in \mathbb{Z}$, x deixa restos $0, 1, 2, \ldots$ ou, $m-1$ quando dividido por m. Dessa forma definimos as classes:

$$\begin{aligned}
&[0] = \{x \in \mathbb{Z}; x \equiv 0 \, mod \, m\} \\
&[1] = \{x \in \mathbb{Z}; x \equiv 1 \, mod \, m\} \\
&[2] = \{x \in \mathbb{Z}; x \equiv 2 \, mod \, m\} \\
&\vdots \\
&[m-1] = \{x \in \mathbb{Z}; x \equiv m-1 \, mod \, m\}
\end{aligned}$$

Quando $a \in \mathbb{Z}$, $[a] = \{x \in \mathbb{Z}; x \equiv a \, mod \, m\}$, chamamos de $[a]$ de classe residual módulo m. Denotamos $\mathbb{Z}_m$ o conjunto das classes residuais em $\mathbb{Z}$ módulo m. Então,

$$\mathbb{Z}_m = \{[a]; a \in \mathbb{Z}\}.$$

Proposição 2.1. *Sejam $a, b \in \mathbb{Z}$:*

i) Duas classes residuais são iguais se, e somente se, quando a é congruente a b módulo m, isto é,

$$[a] = [b] \Leftrightarrow a \equiv b \, mod \, m.$$

ii) Se a interseção de duas classes residuais é diferente de vazio, é por que de fato elas são iguais, isto é,

$$[a] \cap [b] \neq \varnothing \Rightarrow [a] = [b].$$

iii) A união das classes residuais módulo m resulta no próprio conjunto dos inteiros, isto é,

$$\bigcup_{a\in\mathbb{Z}}[a]=\mathbb{Z}.$$

Demonstração. i) Como $b \equiv b \, mod \, m$ (propriedade reflexiva), então

$$b \in [b] = \{x \in \mathbb{Z}; b \equiv b \, mod \, m\}.$$

Por hipótese, $[a] = [b]$, então b se encontra na classe residual de a, ou seja:

$$b \in [a] = \{x \in \mathbb{Z}; x \equiv a \, mod \, m\}$$

Logo $b \equiv a \, mod \, m$. Por fim, da propriedade de simetria, concluímos que $a \equiv b \, mod \, m$.

Para provar a recíproca, suponha $a \equiv b \, mod \, m$. Queremos provar que $[a] = [b]$.

Como as classes residuais na verdade são conjuntos, provemos que $[a] \subset [b]$. Para isso, com $x \in [a]$, então $x \equiv a \, mod \, m$.

Por hipótese, $a \equiv b \, mod \, m$ e $x \equiv a \, mod \, m$. Por simetria, obtemos $b \equiv a \, mod \, m$ e $a \equiv x \, mod \, m$. Valendo-nos da propriedade transitiva, $b \equiv x \, mod \, m$. Por simetria, mais uma vez, obtemos $x \equiv b \, mod \, m$ e, portanto, podemos dizer que esse x é um elemento de [b].

Analogamente se verifica que $[b] \subset [a]$, e finalmente concluímos:

$$[a] \subset [b] \, e \, [a] \supset [b] \Rightarrow [a] = [b] \; .$$

ii) Por hipótese $[a] \cap [b] \neq \varnothing$, ou seja, existe $c \in \mathbb{Z}$ tal que $c \in [a]$ e $c \in [b]$. Logo $c \equiv a \, mod \, m$ e $c \equiv b \, mod \, m$.

Segue do item *i*) que:

$$\begin{cases} c \equiv a \, mod \, m \Leftrightarrow [c] = [a] \\ c \equiv b \, mod \, m \Leftrightarrow [c] = [b] \end{cases} .$$

Logo, $[a] = [b]$.

iii) É claro que $\cup_{a\in\mathbb{Z}}[a] \subset \mathbb{Z}$. Provemos que $\mathbb{Z} \subset \cup_{a\in\mathbb{Z}}[a]$. Para isso, tomemos $a \in \mathbb{Z}$. Na divisão euclidiana da a por m, temos que $a \equiv a \, mod \, m$. Logo $a \in [a]$ e $a \in \cup_{a\in\mathbb{Z}}[a]$. Portanto, da reunião das classes residuais, obtêm-se que:

$$\cup_{a\in\mathbb{Z}}[a] = \mathbb{Z}.$$

□

Proposição 2.2. *Para cada $a \in \mathbb{Z}$, existe um e somente um elemento $r \in \mathbb{Z}$, com $0 \leqslant r \leqslant m-1$ tal que $[a] = [r]$.*

Demonstração. Seja $a \in \mathbb{Z}$, segue do algoritmo da divisão de Euclides que existem únicos q, $r \in \mathbb{Z}$, tais que:

$$a = mq + r \qquad 0 \leqslant r \leqslant m-1.$$

Logo, $a \equiv r \, mod \, m$. Assim, $[a] = [r]$.

□

Agora, vamos definir em $\mathbb{Z}_m$ uma adição e uma multiplicação como segue:

- Adição: $[a] + [b] = [a + b], \quad \forall a, b \in \mathbb{Z}$;
- Multiplicação: $[a].[b] = [a.b] \ , \quad \forall a, b \in \mathbb{Z}$;

Definidas as operações acima para as classes residuais, vale dizer que as operações são fechadas(que o resultado da operação entre dois elementos de um mesmo conjunto tem como resultado um elemento deste próprio conjunto) e que o resultado de $[a] + [b]$ e $[a].[b]$ independe do representante da classe que escolhemos. De fato:

$[a] = [a'] \Leftrightarrow a \equiv a' \, mod \, m$; $[b] = [b'] \Leftrightarrow b \equiv b' \, mod \, m$;

E, portanto, a soma dos elementos resultará a afirmação

$[a + b] = [a' + b'] \Leftrightarrow a + b \equiv a' + b' \, mod \, m$.

Atenção para o fato que sejam a e a' elementos da classe $[a]$ e b e b' elementos da classe $[b]$, decorre simultaneamente que :

$$\begin{cases} [a] = [a'] \Leftrightarrow a \equiv a' \, mod \, m \Leftrightarrow \exists p \in \mathbb{Z}; \, (a - a') = m\mathbf{p} \\ [b] = [b'] \Leftrightarrow b \equiv b' \, mod \, m \Leftrightarrow \exists q \in \mathbb{Z}; \, (b - b') = m\mathbf{q} \end{cases}$$

Da soma das equações acima, obtém-se

$$(a - a') + (b - b') = (a + b) - (a' + b') = m\mathbf{p} + m\mathbf{q} = m(\mathbf{p} + \mathbf{q}) \Leftrightarrow$$
$$(a + b) - (a' + b') = m(\mathbf{p} + \mathbf{q}).$$

Daí, temos que $a + b \equiv a' + b' \, mod \, m$ e $[a + b] = [a' + b']$.

Situação análoga se verifica para a operação $[a.b] = [a'.b']$, pois conhecidos os representantes da classe como definida acima, temos que:

$$\begin{cases} [a] = [a'] \Leftrightarrow a \equiv a' \, mod \, m \Leftrightarrow m \mid (a - a') \\ [b] = [b'] \Leftrightarrow b \equiv b' \, mod \, m \Leftrightarrow m \mid (b - b') \end{cases}$$

Como $a.b - a'b' = a.b - a'b' - ab' + ab' = a(b-b') + b'(a-a')$, temos que $m \mid (a.b - a'b')$, que equivale a dizer

$$[a'.b'] = [a.b]$$

Temos, portanto, que as operações soma e multiplicação independem do representante da classe.

Proposição 2.3. *Para quaisquer* $[a], [b], [c] \in \mathbb{Z}_m$*, temos que* $(\mathbb{Z}_m, +, .)$ *é um* ***anel*** *cujas seguintes propriedades são válidas:*

A1 Associativa: $([a] + [b]) + [c] = [a] + ([b] + [c])$;

P1 Associativa: $([a].[b]).[c] = [a].([b].[c])$;

A2 Comutativa: $[a] + [b] = [b] + [a]$;

P2 Comutativa: $[a].[b] = [b].[a]$; ***(Anel Comutativo).***

A3 Elemento Neutro: $\exists [0] \in \mathbb{Z}_m; [a] + [0] = [a], \forall a \in \mathbb{Z}_m$;

P3 Elemento Neutro: $\exists [1] \in \mathbb{Z}_m; [a].[1] = [a], \forall a \in \mathbb{Z}_m$; ***(Anel com Unidade).***

A4 Inverso Aditivo: Dado $[a] \in \mathbb{Z}_m$; *existe* $[-a] \in \mathbb{Z}_m$ *talque* $[a] + [-a] = [0]$;

A5 Distributiva: $[a].([b] + [c]) = [a].[b] + [a].[c]$.

Demonstração. De fato, sejam $[a], [b], [c] \in \mathbb{Z}_m$.

A1 $([a]+[b])+[c] = ([a+b])+c = [(a+b)+c] = [a+(b+c)] = [a]+[b+c] = [a]+([b]+[c])$.

P1 $([a].[b]).[c] = ([a.b].[c]) = [(a.b).c] = [a.(b.c)] = [a].[b.c] = [a].([b].[c])$.

A2 $[a] + [b] = [a + b] = [b + a] = [b] + [a]$.

P2 $[a].[b] = [a.b] = [b.a] = [b].[a]$.

A3 Sejam $[0] = \{x \in \mathbb{Z}; x \equiv 0 \, mod \, m\}$ e $[a] = \{x \in \mathbb{Z}; x \equiv a \, mod \, m\}$, classes residuais que pertencem ao conjunto $\mathbb{Z}_m$ podemos obter:

$$[a] + [0] = [a + 0] = [a], \text{ qualquer que seja } a \in \mathbb{Z}_m.$$

P3 Sejam $[1] = \{x \in \mathbb{Z}; x \equiv 1 \, mod \, m\}$ e $[a] = \{x \in \mathbb{Z}; x \equiv a \, mod \, m\}$, classes residuais que pertencem ao conjunto $\mathbb{Z}_m$ podemos obter:

$$[a].[1] = [a.1] = [1.a] = [a], \text{ qualquer que seja } a \in \mathbb{Z}_m.$$

A4 Seja $[0] = \{x \in \mathbb{Z}; x \equiv 0 \, mod \, m\}$ e $[a] = \{x \in \mathbb{Z}; x \equiv a \, mod \, m\}$ as classes residuais que pertencem ao conjunto $\mathbb{Z}_m$, existe $[-a] = \{x \in \mathbb{Z}; x \equiv m - a \, mod \, m\} \Leftrightarrow x \equiv -a \, mod \, m$ e de sua soma podemos obter:

$$[a] + [-a] = [a - a] = [0].$$

A5 $([b]+[c]).[a] = ([b+c]).[a] = [(b+c).a] = [(b.a)+(c.a)] = [b.a]+[c.a] = [b][a]+[c][a]$. □

Como $(\mathbb{Z}_m, +, .)$ satisfaz as propriedades para a soma e produto como descrito na proposição 2.3, $\mathbb{Z}_m$ também se denomina **anel das classes residuais módulo** m.

2.2 O Corpo Finito com um Número Primo de Elementos.

Agora iremos nos concentrar nas classes residuais módulo m de um número primo. Quando estamos trabalhando no caso em que $m = p$, com p primo, o conjunto das classes residuais ganha novas propriedades, ou seja, com relação a operação de multiplicação toda classe residual não nula tem elementos opostos, que nesse caso chamaremos o elemento inverso. Para isso, vamos definir alguns conceitos e validar resultados para caracterizar o conjunto das classes residuais que já sabemos que é um anel.

Definição 2.2. *Seja* $[a] \in \mathbb{Z}, [a] \neq [0]$*, dizemos que* $[a]$ *é invertível se existe* $[b] \in \mathbb{Z}$*, tal que* $[a].[b] = 1$*. Nesse caso, dizemos que* $[b]$ *é o inverso de* $[a]$*.*

Exemplo 2.1. *Observe que* $\mathbb{Z}_3 = \{[0], [1], [2]\}$*. Calculemos os elementos inversíveis de* $\mathbb{Z}_3$*.*

Solução: Temos que $[1].[1] = [1]$, então $[1]$ é o inverso do $[1]$.

Da mesma forma, $[2].[2] = [4] = [1]$ então $[2]$ é o inverso de $[2]$.

Logo em $\mathbb{Z}_3$, $\forall \, [a] \neq [0]$, $[a]$ é invertível. □

Exemplo 2.2. *Observe que* $\mathbb{Z}_4 = \{[0], [1], [2], [3]\}$*. Calculemos os elementos inversíveis* $\mathbb{Z}_4$*.*

Solução: Em $\mathbb{Z}_4$, $[1]$ e $[3]$ são invertiveis, entretanto $[2]$ não é invertível, pois não existe $x \in \mathbb{Z}_4$ tal que $[2].[x] = 1$. □

+	[0]	[1]	[2]	[3]
[0]	[0]	[1]	[2]	[3]
[1]	[1]	[2]	[3]	[0]
[2]	[2]	[3]	[0]	[1]
[3]	[3]	[0]	[1]	[2]

.	[0]	[1]	[2]	[3]
[0]	[0]	[0]	[0]	[0]
[1]	[0]	**[1]**	[2]	[3]
[2]	[0]	[2]	[0]	[2]
[3]	[0]	[3]	[2]	**[1]**

Tabela 2.1 – Tabelas de adição e multiplicação em $\mathbb{Z}_4$

Fonte: autor

Definição 2.3. *Seja $m \in \mathbb{Z}, m > 1$, um elemento $[a] \neq [0]$ em $\mathbb{Z}_m$ é chamado divisor de zero se existe $[b] \neq [0]$ tal que $[a].[b] = [0]$.*

Exemplo 2.3. *Os divisores de zero em $\mathbb{Z}_6$ são $[2]$ e $[3]$, pois $[2].[3] = [0]$.*

Proposição 2.4. *Um divisor de zero nunca é invertível.*

Demonstração. Suponha (por absurdo) que $[a]$ é um divisor de zero invertível, logo existe $[b] \in \mathbb{Z}_m$ tal que $[a].[b] = 1$. Como $[a]$ é divisor de zero, então existe $[c] \neq 0$, tal que $[a].[c] = 0$. Multiplicando por $[b]$ em ambos os lados de $[a].[c] = [0]$, temos

$$[b].[a].[c] = [b].[0]$$

$$([b].[a]).[c] = [0]$$

Como sabemos que $[1].[c] = [c]$. Logo $[c] = [0]$ e teremos uma contradição. Portanto, $[a]$ não é invertível. □

Definição 2.4. *Dado $m \in \mathbb{Z}$, com $m > 1$. O anel $\mathbb{Z}_m$ é chamado corpo se todo elemento não nulo de $\mathbb{Z}_m$ possuir inverso multiplicativo, ou seja,*

$$\forall [a] \neq [0] \in \mathbb{Z}_m, \exists\ [b] \neq [0] \in \mathbb{Z}_m, \text{ tal que } [a].[b] = 1.$$

Exemplo 2.4. *$\mathbb{Z}_5 = \{[0], [1], [2], [3], [4]\}$ é um corpo.*

Demonstração. Construindo a tabela de multiplicação para $\mathbb{Z}_5$ e determinando os inversos de cada classe residual, obtêm-se:

$$[1].[1] = [1]$$
$$[2].[3] = [6] = [1]$$
$$[3].[2] = [6] = [1]$$
$$[4].[4] = [16] = [1]$$

Logo, $\mathbb{Z}_5$ é um corpo. □

Proposição 2.5. *Um elemento $[a] \in \mathbb{Z}_m$ é invertível se, e somente se, $mdc(a, m) = 1$.*

Demonstração. Temos que $[a]$ é invertível se, e somente se, existe $[b] \in \mathbb{Z}_m$ tal que $[a].[b] = 1$. Assim, $[a.b] = [1]$, ou seja, $a.b \equiv 1 \, mod \, m$ se e somente se, existe $y \in \mathbb{Z}$, tal que $a.b-1 = my$, ou ainda $ab - my = 1$.

Observe que encontrar o inverso de um elemento $[a] \in \mathbb{Z}_m$ equivale a resolver a equação diofantina $ax + my = 1$.

Portanto, $[a] \in \mathbb{Z}_m$ é invertível se e somente se $mdc(a, m) = 1$. □

Proposição 2.6. $(\mathbb{Z}_m, +, .)$ *é um corpo se, e somente se, m é primo.*

Demonstração. Suponha que $\mathbb{Z}_m$ é um corpo e provemos que m é primo. Suponha por absurdo que m é composto, ou seja, $m = a.b$ com $1 < a, b < m$. Como $1 < a, b < m$ então $[a] \neq [0]$ e $[b] \neq [0]$. Observe ainda que $[0] = [m] = [a.b] = [a].[b]$.

Logo $[a].[b] = [0]$, com $[a] \neq [0]$ e $[b] \neq [0]$ e temos uma contradição, pois sendo $\mathbb{Z}_m$ um corpo, $[b]$ é inversivel e pela Proposição 2.4 não pode ser divisor de zero.

Para provar a recíproca, suponha m primo, logo $mdc(m, i) = 1$, $1 \leq i \leq m - 1$. Temos, pela Proposição 2.5, $(m, i) = 1$ se, e somente se, $[i]$ é invertível em $\mathbb{Z}_m$, com $i = 1, 2, \ldots, m - 1$. Logo, $\mathbb{Z}_m$ é um corpo. □

Para compreender melhor as considerações sobre o conjunto $\mathbb{Z}_m$, nas demonstrações ou definições nos capítulos seguintes, a referência a esta demonstração denotaremos corpo "$\mathbb{Z}_p$, com p primo", ou simplesmente $\mathbb{Z}_p$.

2.3 Alguns exemplos práticos de aplicação de Classes Residuais

A ideia do anel das classes residuais está presente em alguns exemplos práticos do nosso cotidiano. Sem dúvida, a maior aplicabilidade deles é na criação de dígitos verificadores, servindo para indicar e evitar erros de digitação. Na prática, eles evitam que ambientes seguros aceitem fraudes, como forma de comprovar a autenticidade daquele documento.

A chance de uma pessoa digitar errado um dígito de seu CPF(em um site, numa compra online, por exemplo) é muito maior do que a probalidade de errar dois dígitos, que é maior que de errar 3 e assim por diante.

De acordo com (EDC. . . ,), se verifica que:

> Mesmo sustituindo os números decimais por binários, os mecanismos não mudam. [· · ·] O que muda é o " poder de fogo " da detecção de

erros acima da capacidade de códigos. Por exemplo, o dígito vericador decimal pode detectar até 90% dos erros de dois dígitos ou mais. Já o seu equivalente binário, o bit de paridade, detecta apens 50% dos erros de dois bits ou mais.

Isto acontece simplesmente porque um dígito decimal tem muito mais valores distintos que um dígito binário (bit). No caso de um DV duplo como o do CPF, 99% dos erros são pegos, enquanto um código de 2 bits pega apenas 75% dos erros. A taxa de detecção é dada por:

$$p = \frac{b^n - 1}{b^n}$$

, onde:

n = número de dígitos verificadores ou bits do código;

p = probabilidade de pegar erros de mais de "n" dígitos;

b = base numérica (binário = 2, decimal = 10)

De fato:

Para 1 dígito decimal

$$p = \frac{10^1 - 1}{10^1} = \frac{10 - 1}{10} = \frac{9}{10} = 90\%$$

Para 2 dígitos decimais

$$p = \frac{10^2 - 1}{10^2} = \frac{100 - 1}{100} = \frac{99}{100} = 0,99 = 99\%$$

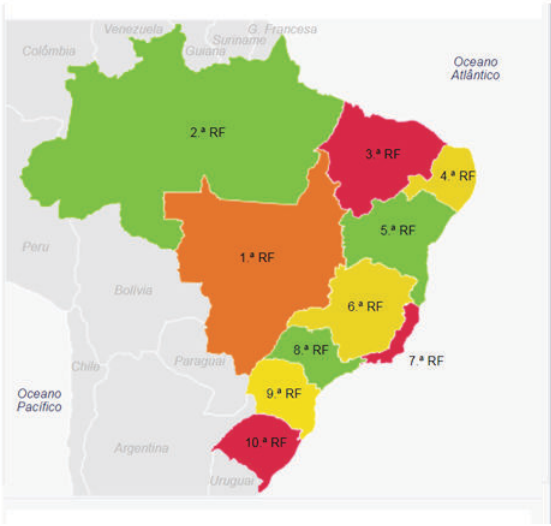
2.3.1 Os Dígitos de Verificação do CPF

O CPF, ou simplesmente Cadastro de Pessoa Física, é um documento de identificação, tal como o registro de nascimento, que a receita federal emite para armazenar informações dos contribuintes da receita federal, com informações fornecidas pelos mesmos ou seus responsáveis (no momento em que foi exigido à época de registro de nascimento). Serve para comprovar que a pessoa contribui com a Receita Federal ou é dependente de alguém que contribui e é necessário em algumas situações, como a abertura de conta em banco, crediário em lojas, cadastro em sites, inscrição em concursos, etc.

Figura 2.2 – O Cadastro de Pessoa Física - CPF

Fonte: Governo Federal.

Para evitar duplicidade, o CPF tem estrutura de 11 dígitos, dos quais os oito primeiros são aleatórios (possuindo algarismos de 0 a 9), o nono designa a região fiscal (de acordo com o local de origem), e os últimos dois dígitos (DV, Dígito Verificador) provém de um algarismo que usa a aritmética modular módulo 11.



REGIÃO FISCAL	DÍG.
1ª: DF, GO, MT, MS e TO;	1
2ª: AC, AP, AM, PA, RO e RR;	2
3ª: CE, MA e PI;	3
4ª: AL, PB, PE e RN;	4
5ª: BA e SE;	5
6ª: MG;	6
7ª: ES e RJ;	7
8ª: SP;	8
9ª: PR e SC;	9
10ª: RS.	0

Figura 2.3 – Regiões Fiscais- CPF

Fonte: Governo Federal. Adaptada pelo autor.

Basicamente, a obtenção dos dígitos verificadores são obtidos pelo algoritmo explicado abaixo:

Se $a_1a_2a_3a_4a_5a_6a_7a_8a_9$ é a sequência formada pelos nove primeiros dígitos, devemos multiplicar eles nesta mesma ordem, pelos números: $1, 2, 3, 4, 5, 6, 7, 8, 9$, e seguida somar os produtos obtidos.

Se este primeiro resultado obtido denominarmos S_1, o décimo dígito que denotaremos por a_{10}, deverá ser tal que $S_1 - a_{10}$ seja múltiplo de 11 ou $S_1 - a_{10} \equiv 0\,(mod\,11)$. Assim, obtemos que S_1 será o próprio resto da divisão por 11 da soma obtida.

Para obtermos o segundo dígito de controle, utilizamos o mesmo procedimento acrescentando o décimo dígito que acabamos de calcular e usando uma base de multiplicação de 0 a 9, nesta mesma ordem, pelos números: $0, 1, 2, 3, 4, 5, 6, 7, 8, 9$, assim obtendo uma soma S_2.

O décimo primeiro dígito que denotamos por a_1, será tal que $S_2 - a_{11}$ seja múltiplo de 11 ou $S_2 - a_{11} \equiv 0\,(mod\,11)$. Assim, obtemos que S_2 será o próprio resto da divisão por 11 da soma obtida.

Para fins didáticos, um provável CPF do Mato Grosso do Sul, poderia ser exemplificado com o número 850.098.821-99. Todavia, ele é falso, pois o primeiro digito verificador válido é 5 e não 9. Para provar tal afirmação, vamos seguir os passos do procedimento acima relatado.

Multiplicando os dígitos $8, 5, 0, 0, 9, 8, 8, 2, 1$ pelos números $1, 2, 3, 4, 5, 6, 7, 8, 9$ e calculando sua soma, S_1, encontramos:

$$S_1 = 8.1+5.2+0.3+0.4+9.5+8.6+8.7+2.8+1.9 = 8+10+0+0+45+48+56+16+9 = 192$$

Como $S_1 \equiv 5\,(mod\,11)$, o primeiro dígito de verificação é 5.

Como se pode ver, o CPF 850.098.821-99, por sí só atésta-se que é inválido pelo seu primeiro dígito verificador. Mesmo assim, continuaremos o procedimento para descobrir S_2 e o segundo dígito de verificação. Daí:

Multiplicando os dígitos $8, 5, 0, 0, 9, 8, 8, 2, 1, 5$ pelos números $0, 1, 2, 3, 4, 5, 6, 7, 8, 9$ e calculando sua soma, S_2, encontramos:

$$S_1 = 8.0 + 5.1 + 0.2 + 0.3 + 9.4 + 8.5 + 8.6 + 2.7 + 1.8 + 5.9 =$$
$$0 + 5 + 0 + 0 + 36 + 40 + 48 + 14 + 8 + 45 = 196$$

Como $S_2 \equiv 9\,(mod\,11)$, o segundo dígito de verificação é 9.

Na verdade, o CPF correto para MS com os dígitos de verificação corretos do nosso exemplo será 850.098.821-59.

2.3.2 As potências nas Congruências

Outro exemplo que podemos aplicar classes residuais é para determinar o resto da divisão de uma potência de um número inteiro positivo, por exemplo, queremos determinar o resto da divisão de 2^{100} por 11, para isso basta encontrar a classe de $[a] \in \{[0],[1],[2],[3],[4],[5],[6],[7],[8],[9],[10]\}$ tal que $[2^{100}] = [a]$.

Inicialmente, denominando os representantes da classe residual modulo 11, teríamos:

$$[0] = \{x \in \mathbb{Z}; x \equiv 0 \, mod \, m\}$$
$$[1] = \{x \in \mathbb{Z}; x \equiv 1 \, mod \, m\}$$
$$\vdots$$
$$[10] = \{x \in \mathbb{Z}; x \equiv m-1 \, mod \, m\}$$

E valendo da Proposição 2.1 e da propriedade abaixo, adaptada para uma melhor compreensão, segue que a relação de congruência módulo m divide $\mathbb{Z}$ em classes de equivalência as quais são chamadas de classe residuais ou classe dos restos das divisões por m. Daí, se $a \equiv b \, (mod \, 11)$, então $a^r \equiv b^r \, (mod \, 11)$, com $a, b, r \in \mathbb{Z}$. Vamos construir as potências de 2, e determinar a classe residual a qual respctivamente pertence. Temos

$2^0 \equiv 2^0 \, (mod \, 11) \Rightarrow 2^0 \equiv 1 \, (mod \, 11)$ *e* $1 \equiv 1 \, (mod \, 11) \Rightarrow [2^0] = [1]$,

$2^1 \equiv 2^1 \, (mod \, 11) \Rightarrow 2^1 \equiv 2 \, (mod \, 11)$ *e* $2 \equiv 2 \, (mod \, 11) \Rightarrow [2^1] = [2]$,

$2^2 \equiv 2^2 \, (mod \, 11) \Rightarrow 2^2 \equiv 4 \, (mod \, 11)$ *e* $4 \equiv 4 \, (mod \, 11) \Rightarrow [2^2] = [4]$,

$2^3 \equiv 2^3 \, (mod \, 11) \Rightarrow 2^3 \equiv 8 \, (mod \, 11)$ *e* $8 \equiv 8 \, (mod \, 11) \Rightarrow [2^3] = [8]$,

$2^4 \equiv 2^4 \, (mod \, 11) \Rightarrow 2^4 \equiv 16 \, (mod \, 11)$ *e* $16 \equiv 5 \, (mod \, 11) \Rightarrow [2^4] = [5]$,

$2^5 \equiv 2^5 \, (mod \, 11) \Rightarrow 2^5 \equiv 32 \, (mod \, 11)$ *e* $32 \equiv 10 \, (mod \, 11) \Rightarrow [2^5] = [10]$,

$2^6 \equiv 2^6 \, (mod \, 11) \Rightarrow 2^6 \equiv 64 \, (mod \, 11)$ *e* $64 \equiv 9 \, (mod \, 11) \Rightarrow [2^6] = [9]$,

$2^7 \equiv 2^7 \, (mod \, 11) \Rightarrow 2^7 \equiv 128 \, (mod \, 11)$ *e* $128 \equiv 7 \, (mod \, 11) \Rightarrow [2^7] = [7]$,

$2^8 \equiv 2^8 \, (mod \, 11) \Rightarrow 2^8 \equiv 256 \, (mod \, 11)$ *e* $256 \equiv 3 \, (mod \, 11) \Rightarrow [2^8] = [3]$,

$2^9 \equiv 2^9 \, (mod \, 11) \Rightarrow 2^9 \equiv 512 \, (mod \, 11)$ *e* $512 \equiv 6 \, (mod \, 11) \Rightarrow [2^9] = [6]$,

$2^{10} \equiv 2^{10} \, (mod \, 11) \Rightarrow 2^{10} \equiv 1024 \, (mod \, 11)$ *e* $1024 \equiv 1 \, (mod \, 11) \Rightarrow [2^{10}] = [1]$,

$2^{11} \equiv 2^{11} \, (mod \, 11) \Rightarrow 2^{11} \equiv 2048 \, (mod \, 11)$ *e* $2048 \equiv 2 \, (mod \, 11) \Rightarrow [2^{11}] = [2]$ e

$2^{12} \equiv 2^{12} \, (mod \, 11) \Rightarrow 2^{12} \equiv 4096 \, (mod \, 11)$ *e* $4096 \equiv 4 \, (mod \, 11) \Rightarrow [2^{12}] = [4]$.

Observa-se em destaque que a partir da potência 2^{10}, temos uma repetição dos restos, e, portanto,encontram-se representantes de classes residuais iguais, como num ciclo. Então podemos dizer que:

$[2^{10}] = [2^0], [2^{11}] = [2^1], [2^{12}] = [2^2], [2^{13}] = [2^3]$, e assim por diante.

Assim uma resolução "esperta" para o problema é escrever o número 100 como uma congruência modulo 10, e uma vez conhecidos os seus restos, verificar o esquema acima e identificar qual o representante da classe aritmética módulo 11. Calculando:

$$2^{10} \equiv 1 \, mod \, 11 \Rightarrow (2^{10})^{10} \equiv (1)^{10} \, mod \, 11 \Rightarrow 2^{100} \equiv 1 \, mod \, 11 \Rightarrow [2^{100}] = [1]$$

Portanto, a classe $[a]$ para o qual $[2^{100}] = [a]$ é o representante da classe $[1]$. Ou ainda, o resto da divisão de $[2^{100}]$ por 11 é 1.

2.3.3 Os ponteiros do relógio.

Outra aplicação prática das congruências é no relógio de ponteiro analógico, onde o dia é dividido em dois períodos de 12 horas cada. Acontece que neste exemplo, ignora-se os múltiplos de um dado número quando fazemos cálculos. Neste exemplo, é a aritmética módulo 12.

Se nesse momento são 5 horas, então daqui a 8 horas será 1 hora. A adição usual sugere que o tempo futuro deveria ser $5 + 8 = 13$, mas na verdade, tomando o mostrador do ponteiro analógico, esta é a resposta errada por que o relógio "volta para trás" a cada 12 horas; não existe "13 horas" no relógio do ponteiro. Dessa forma, o ponteiro das horas desloca-se para o 1.

Numa outra situação, se o relógio começa em $12:00$(meio dia) e 21 horas passam, então a hora será $9:00$ do dia seguinte, em vez de $33:00$. 12 é congruente não só a 12 mesmo, mas também a 0, assim a hora chamada "$12:00$" pode também ser chamada "$0:00$", pois $0 \equiv 12 \, mod \, 12$. Por isso, o ponteiro das horas obedece uma aritmética modular 12.

O mesmo vale para o ponteiro dos minutos, que em ciclos de 60 retoma sua posição inicial, ou seja, deslocando-se entre as posições 00 a 59.

2.3.4 Os números de segurança na nota do Euro.

Outro exemplo de aplicação, citado em (PINTO, 2006) descreve um dos mecanismos de segurança estabecido pelo Banco Central Europeu para notas de Euro que entraram em circulação em 1 de janeiro de 2002, baseado num sistema de congruência módulo 9.

Letra	País	Valor
L	Finlândia	4
M	**Portugal**	5
N	Áustria	6
P	Holanda	8
R	Luxemburgo	1
S	Itália	2
T	Irlanda	3
U	França	4
V	Espanha	5
X	Alemanha	7
Y	Grécia	8
Z	Bélgica	9

Figura 2.4 – Mecanismo de Segurança da Nota do Euro

Fonte: Helder Pinto (2006)

Basicamente, o número de série que podemos encontrar numa qualquer nota (verdadeira) de Euro é composto por uma letra seguida de onze algarismos. A letra representa o país do qual é fabricada a nota (por exemplo, na imagem, T indica essa nota como pertencente a Irlanda) e os onze algarismos representam o número de identificação da nota. O último de seus algarismos chamados dígito de controle ou verificador, é obtido da operação.

$$L + x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8 + x_9 + x_{10} + C \equiv 0 \,(mod\, 9)$$

Onde L é o valor atribuído ao pais respectivo e C é o dígito de controle.

Tomando como referência a nota representada acima, $T23186909697$, verifiquemos se 7 é o dígito verificador (DV) correto.

$$3 + 2 + 3 + 1 + 8 + 6 + 9 + 0 + 9 + 6 + 9 + C \equiv 0 \,(mod\, 9)$$

Perceba que a soma parcial resulta em 56, que sobre uma congruência módulo 9 resulta em:

$$56 + C \equiv 0 \,(mod\, 9) \Rightarrow$$
$$2 + C \equiv 0 \,(mod\, 9)$$

Daí o único número que C pode assumir para a soma ser congruente é 7. Portanto, 7 é verdadeiramente o dígito verificador.

Neste mesmo artigo, ele realiza uma crítica quanto à eficácia, quando, na pág 86 diz que:

> Que tipos de erros são detectados por este sistema? Em primeiro lugar, nenhuma transposição de dois (ou mais) algarismos é detectada devido à propriedade comutativa da adição (modular) e mesmo os erros singulares

> não são todos detectados. Se trocarmos um zero por um nove (ou vice-versa) o número de controle mantém-se inalterado.
>
> [...] De fato, se escrevemos um nove em vez de um zero (ou vice-versa) o número de controle mantém-se inalterado. Note-se que este é um problema prático grave, pois, se reparar no seu teclado, existe uma tecla nove junto de uma tecla zero, o que pode levar a que este engano seja frequente. Esta última insuficiência surge do fato de 9 ser igual a 0 na aritmética módulo 9, ou seja, tanto o nove como o zero têm o mesmo resto quando divididos por nove. O fato de este mecanismo não detectar com segurança estes dois tipos de erros (que são os tipos de erro mais frequentes), torna-o num sistema bastante fraco.

Como se observa nos exemplos acima, tais situações são inclusive motivadores para elaboração de planos de aulas para o ensino fundamental. Para tal, é possível introduzir a ideia do efeito causado na mudança de apenas um dígito e como ela influi na determinação do DV.[1]

Um outro exemplo da aplicabilidade da congruência e classe residual, este mais sintonizado com a aprendizagem em sala de aula, é explicar superficialmente a criação de um código através de uma lista de palavras, explicando a relação que a mesma tem com o conteúdo de matrizes e operações, para a segunda série do ensino médio. Neste caso, a codificação das palavras de um código pode ser feita com o auxílio de um matriz chave para "embaralhar" as letras, e o uso da sua inversa para desembaralhar, é a melhor estratégia. As operações de multiplicação e soma são ótimas opções para exploração e abordagem dos códigos corretores.

Mas para mostrar a álgebra dos códigos corretores, é preciso, antes de tudo, entender sob qual conjunto está estruturado e as operações e propriedades deste. Daí, utilizamos a abordagem de um espaço vetorial especial, o $\mathbb{Z}_p^n$, que abordaremos no capítulo seguinte.

[1] Digito verificador ou algarismo de controle é um mecanismo de autenticação utilizado para verificar a validade e a autenticidade de um campo numérico digitado, evitando dessa forma fraudes. São algarismos adicionais colocados e validados por um algoritmo para garantir a segurança. Documentos de identificação normalmente utilizam dígitos verificadores.

3 O Espaço Vetorial Finito $\mathbb{Z}_p^n$.

Como visto no Capítulo 2, vemos que o uso das classes residuais e a ideia de congruências é de grande valia para algumas aplicações no cotidiano e são facilitadores na definição de dígitos de verificação (como no caso do CPF). Isso é possível devido a escolha de quais algarismos e operações serão válidas para tal função.

Este capítulo é destinado a apresentar o espaço vetorial $\mathbb{Z}_p^n$, cujo o número de elementos é finito, essencial para o entendimento dos códigos lineares e suas propriedades, com suas respectivas propriedades. Ao operar com os diferentes tipos de números, estamos intuitivamente manipulando valores, sem que estes alterem suas estruturas.

A escolha do corpo $\mathbb{Z}_p$, não é aleatória, devido ao fato das operações e propriedades que o caracterizam estarem bem definidas e provadas no capítulo anterior. A partir dele, as n-uplas permitem a definição de um espaço vetorial, fundamental para o nosso trabalho.

Se pensamos a informação a ser transmitida como um pacote com n-uplas de valores ordenados, temos as palavras de um código fundamentado numa estrutura que segue a de um espaço vetorial. Neste capítulo, vamos entender o espaço vetorial e algumas dessas propriedades.

3.1 O Espaço Vetorial $\mathbb{Z}_p^n$

A ideia é considerar $\mathbb{Z}_p^n$, onde p é primo e $n \in \mathbb{Z}, n \geq 1$, que consiste do produto cartesiano de n cópias de $\mathbb{Z}_p$, ou seja,

$$\mathbb{Z}_p^n = \underbrace{\mathbb{Z}_p \times \mathbb{Z}_p \times \ldots \times \mathbb{Z}_p}_{n\ vezes}$$

e explorar a estrutura de espaço vetorial em $\mathbb{Z}_p^n$, juntamente com suas propriedades e conceitos inerentes das características que possuem os espaços vetoriais finitamente gerados. Apresentamos os resultados de espaços vetoriais, e verficando a aplicabilidade para subespaços de $\mathbb{Z}_p^n$, pois os subespaços são exemplos de espaços vetoriais. Percebemos que essas demonstrações análogas as demonstrações do caso geral que constam em livros de álgebra linear. A bibliografia de consulta para este capítulo se baseia nos autores (HEFEZ, 2016), (WINTERLE; STEINBRUCH, 1987), (ANTON; RORRES, 2001),(PULINO, 2012) e (BOLDRINI et al., 1978).

Em todo texto usaremos a notação $\mathbb{Z}_p^n$.

Segue, do Capítulo 2, que $\mathbb{Z}_p$=$\{[0],[1],[2],\ldots,[p-1]\}$ é um corpo. Nas próximas seções e capítulos iremos abandonar a notação $[a] \in \mathbb{Z}_p$ e apenas escreveremos a, mas trata-se de uma classe residual módulo p. Faremos isso para a notação de vetor não ficar muito carregada.

Temos que

$$u \in \mathbb{Z}_p^n \Leftrightarrow u = (a_1, a_2, \ldots, a_n) \ \ com \ a_1, a_2, a_n \in \mathbb{Z}_p.$$

Os elementos $u \in \mathbb{Z}_p^n$ são chamados *vetores*.

Segue da definição do corpo $\mathbb{Z}_p$ e do princípio multiplicativo (uma vez que o anel $\mathbb{Z}_p^n$ é formado por n-uplas de $\mathbb{Z}_p$) que a cardinalidade $\mid \mathbb{Z}_p^n \mid = p^n$, que torna o referido conjunto finito. Vamos então definir uma soma de dois vetores $u, v \in \mathbb{Z}_p^n$ e uma multiplicação de um escalar $\alpha \in \mathbb{Z}_p$ por um vetor $u \in \mathbb{Z}_p^n$ explorando a estrutura de espaço vetorial em $\mathbb{Z}_p^n$.

Para isso, sejam $u, v, w \in \mathbb{Z}_p^n$.

$$u \in \mathbb{Z}_p^n \Leftrightarrow u = (a_1, a_2, \ldots, a_n);$$
$$v \in \mathbb{Z}_p^n \Leftrightarrow u = (b_1, b_2, \ldots, b_n);$$
$$w \in \mathbb{Z}_p^n \Leftrightarrow u = (c_1, c_2, \ldots, c_n);$$

Definimos:

$+ : u + v = (a_1 + b_1, a_2 + b_2, \ldots, a_n + b_n)$ e

$. : \alpha.u = (\alpha a_1, \alpha a_2, \ldots, \alpha a_n)$, com $\alpha \in \mathbb{Z}_p$.

O conjunto $\mathbb{Z}_p^n$, com as operações + e . forma um **Espaço Vetorial** sobre $\mathbb{Z}_p$, ou seja, satisfazem as seguintes operações:

$A_1)$ $u + v = v + u$ (Comutativa);

$A_2)$ $(u + v) + w = u + (v + w)$(Associativa);

$A_3)$ $\exists\ e \in V$, denominado elemento neutro, tal que $u + e = e + u = u$;

$A_4)$ Para cada $u \in V$, tem-se $(-u) \in V$, chamado elemento oposto de u, tal que $u + (-u) = e$.

$M_1)$ $(\alpha\beta)u = \alpha(\beta u)$;

$M_2)$ $(\alpha + \beta)u = \alpha u + \beta u$;

$M_3)$ $\alpha(u + v) = \alpha u + \alpha v$;

$M_4)$ $1u = u$.

A demonstração que $\left(\mathbb{Z}_p^n, +, .\right)$ é um espaço vetorial pode ser encontrada em (WINTERLE; STEINBRUCH, 1987) , p. 20-21.

3.2 Os Subespaços de $\mathbb{Z}_p^n$

Definição 3.1. *Seja $W \subset \mathbb{Z}_p^n$, W não vazio. Dizemos que W é subespaço vetorial de $\mathbb{Z}_p^n$ se as operações de $\mathbb{Z}_p^n$ em W torna esse conjunto ainda um espaço vetorial sobre $\mathbb{Z}_p$.*

Proposição 3.1. *Seja $W \subset \mathbb{Z}_p^n$, W não vazio. Então W é subespaço de $\mathbb{Z}_p^n$, se satisfaz as propriedades:*

i) $(0, 0, \ldots, 0) \in W$;

ii) W *é fechado para a adição de vetores, isto é,* $u + v \in W, \forall u, v \in W$;

iii) W *é fechado na multiplicação por escalar, ou seja,* $\alpha u \in W, \forall u \in W \, e \, \forall \alpha \in \mathbb{Z}_p$.

Demonstração. Suponhamos W subespaço vetorial de $\mathbb{Z}_p^n$. Basta considerar que nos espaços vetoriais a adição de vetores e a multiplicação por escalar é fechada, e válida para $\alpha = 0$, ou seja, $0v \in W, \forall v \in W$. Logo, $0v = (0, 0, \ldots, 0) \in W$, e, portanto, o elemento neutro $= (0, 0, \ldots, 0)$ de V está em W.

Se W é espaço vetorial, então satisfaz todas as condições de espaço vetorial, em particular, satisfaz $(ii), (iii)$.

Para a recíproca deste teorema, tomemos os vetores $u, v \in W$ e um escalar $\alpha \in \mathbb{Z}_p$. temos que:

As propriedades $(A_1), (A_2), (M_1), (M_2), (M_3), (M_4)$ são herdadas de $\mathbb{Z}_p^n$, pois $W \subset \mathbb{Z}_p^n$. Vimos de $i)$ que $(0, \cdots, 0) \in W$ e $(0, \cdots, 0) + u = u$, para todo $u \in W$. Logo é válido (A_3).

Como $\alpha.u \in W$, para todo $u \in W$ e $\alpha \in \mathbb{Z}_p$, então $-u = (-1).u \in W$ e $u + (-u) = 0$, logo vale (A_4).

Portanto W é espaço vetorial.

□

Exemplo 3.1. *O subconjunto de um espaço vetorial formado apenas pelo vetor nulo $(0, 0, \ldots, 0)$ é um subespaço vetorial de $\mathbb{Z}_p^n$.*

Exemplo 3.2. *O próprio $\mathbb{Z}_p^n$ como subconjunto dele mesmo também é um subespaço vetorial. Estes dois subespaços são chamados triviais.*

3.3 Combinação Linear

Definição 3.2. *Um vetor v que é um elemento do espaço vetorial $\mathbb{Z}_p^n$ é denominado* ***combinação linear*** *dos elementos $v_1, v_2, \ldots, v_n \in \mathbb{Z}_p^n$ quando existem escalares $\alpha_1, \alpha_2, \ldots, \alpha_n \in \mathbb{Z}_p$, de tal forma que ele esse vetor pode ser determinado pela seguinte operação:*

$$v = \alpha_1 v_1 + \alpha_2 v_2 + \cdots + \alpha_n v_n.$$

Exemplo 3.3. *O elemento $v = (0,1) \in \mathbb{Z}_2^2$ é combinação linear dos elementos $v_1 = (1,0)$ e $v_2 = (0,1)$.*

De fato, v pode ser escrito como:

$$v = (0,1) = 0.(1,0) + 1.(0,1) = 0.v_1 + 1.v_2$$

Assim, existem os escalares $\alpha_1 = 0$ e $\alpha_2 = 1$ tais que v pode ser escrito como $v = \alpha_1 v_1 + \alpha_2 v_2$. Logo, v é combinação linear de v_1 e v_2.

Exemplo 3.4. *O elemento $v = (2,0,1) \in \mathbb{Z}_3^3$ é combinação linear dos elementos $v_1 = (1,0,0)$ e $v_2 = (0,0,1)$.*

De fato, v pode ser escrito como:

$$v = (2,0,1) = 2.(1,0,0) + 1.(0,0,1) = 2.v_1 + 1.v_2$$

Assim, existem os escalares $\alpha_1 = 2$ e $\alpha_2 = 1$ tais que v pode ser escrito como $v = \alpha_1 v_1 + \alpha_2 v_2$. Logo, v é combinação linear de v_1 e v_2. Partindo da premissa que um vetor pertencente ao espaço vetorial $\mathbb{Z}_p^n$ é escrito por meio de outros vetores operados, e estes, que sempre aparecerão, permitem estabelecer uma definição importante descrita na seção seguinte.

3.4 Subespaço Gerado

Definição 3.3. *Seja $S = \{v_1, v_2, \ldots, v_m\} \subset \mathbb{Z}_p^n$. O subconjunto $W \subset \mathbb{Z}_p^n$ formado por todos os elementos $u \in \mathbb{Z}_p^n$ que podem ser escritos como combinação linear dos elementos de S é chamado* ***subespaço gerado*** *por S e denotamos:*

$$\begin{aligned} W &= [S] = [v_1, \ldots, v_m] \\ &= \left\{ u \in \mathbb{Z}_p^n \mid u = \alpha_1 v_1 + \alpha_2 v_2 + \cdots + \alpha_m v_m = \sum_{i=1}^{m} \alpha_i v_i;\ \alpha_1, \ldots, \alpha_m \in \mathbb{Z}_p \right\} \end{aligned}$$

Proposição 3.2. *Seja $S = \{v_1, v_2, \ldots, v_m\} \subset \mathbb{Z}_p^n$, com p primo e $n \geq 1$ natural. Então $W = [S] = [v_1, \ldots, v_m]$ é um subespaço de $\mathbb{Z}_p^n$.*

Demonstração. Seja $S = \{v_1, v_2, \ldots, v_m\}$ um conjunto de m elementos de $\mathbb{Z}_p^n$. Para provar este resultado basta verificar que valem as condições de subespaço vetorial para $[S]$:

i) O elemento neutro de $\mathbb{Z}_p^n$ está em $[S]$, pois basta observar que $(0, 0, \ldots, 0) = 0v_1 + 0v_2 + \cdots + 0v_m$;

ii) Considere $u, w \in [S]$. Se $u \in [S]$, então $u = \alpha_1 v_1 + \alpha_2 v_2 + \cdots + \alpha_m v_m$, com $\alpha_1, \alpha_2, \cdots, \alpha_m \in \mathbb{Z}_p$. E se $w \in [S]$, então $w = \beta_1 v_1 + \beta_2 v_2 + \cdots + \beta_m v_m$, com $\beta_1, \beta_2, \cdots, \beta_m \in \mathbb{Z}_p$.

Temos que, $u + w = (\alpha_1 v_1 + \alpha_2 v_2 + \cdots + \alpha_m v_m) + (\beta_1 v_1 + \beta_2 v_2 + \cdots + \beta_m v_m) = (\alpha_1 + \beta_1)v_1 + (\alpha_2 + \beta_2)v_2 + \cdots + (\alpha_m + \beta_n)v_m$. Como $\alpha_i + \beta_i \in \mathbb{Z}_p$, temos que $u + w$ é também combinação linear dos elementos de S, logo $u + w \in [S]$;

iii) Considere $u \in [S]$ e um escalar $\beta \in \mathbb{Z}_p$. Se $u \in [S]$, então $u = \alpha_1 v_1 + \alpha_2 v_2 + \cdots + \alpha_m v_m$, com $\alpha_1, \alpha_2, \cdots, \alpha_m \in \mathbb{Z}_p$. Temos que $\beta u = \beta(\alpha_1 v_1 + \alpha_2 v_2 + \cdots + \alpha_m v_m) = \beta\alpha_1 v_1 + \beta\alpha_2 v_2 + \cdots + \beta\alpha_m v_m$. Como $\beta\alpha_i \in \mathbb{Z}_p$, temos que βu é também combinação linear dos elementos de S, logo $\beta u \in [S]$. Assim, provamos que $[S]$ é um subespaço vetorial de $\mathbb{Z}_p^n$.

□

O conjunto gerado por $S = \{v_1, v_2, \ldots, v_m\} \subset \mathbb{Z}_p^n$, $W = [v_1, \ldots, v_m]$ é chamado também de **subespaço gerado** por S, ou subespaço gerado por $v_1, v_2, \ldots, v_m$.

Exemplo 3.5. *O conjunto $W = \{(x_1, x_2, x_3) \in \mathbb{Z}_3^3 : x_1 = 0, x_2 \in \mathbb{Z}_3\}$ é um subespaço de $\mathbb{Z}_3^3$ gerado pelo vetor $v_1 = (0, 1, 1)$.*

Definição 3.4. *Um espaço vetorial V é **finitamente gerado** se existem $v_1, v_2, \cdots, v_m \in V$, tal que $V = [v_1, v_2, \cdots, v_m]$*

Exemplo 3.6. *O conjunto $S = \{(1, 0), (0, 1)\}$ gera o espaço vetorial $\mathbb{Z}_2^2$.*

Exemplo 3.7. *O conjunto $S = \{(1, 1)\} \subset \mathbb{Z}_2^2$ gera o subespaço $U = \{(x, y) \in \mathbb{Z}_2^2 \mid y = x\}$.*

De fato, tomando um elemento $u = (x, y) \in U$, temos que $y = x$, logo podemos escrever: $u = [(x, y) = (x, x) = x(1, 1)$, com $x \in \mathbb{Z}_2]$.

3.5 Dependência e Independência Linear em $\mathbb{Z}_p^n$

Definição 3.5. *Seja $S = \{v_1, v_2, \ldots, v_m\} \subset \mathbb{Z}_p^n$, com $m \leqslant n$ de $\mathbb{Z}_p^n$. Dizemos que o conjunto S é **Linearmente Independente (L . I)**, quando para $\alpha_1, \alpha_2, \ldots, \alpha_m \in \mathbb{Z}_p$ tais que*

$$\alpha_1.v_1 + \alpha_2.v_2 + \cdots + \alpha_m.v_m = (0, 0, \ldots, 0)$$

tivermos $\alpha_i = 0$, para todo $i = 1, ..., m$, ou seja $\alpha_1 = \alpha_2 = \ldots = \alpha_m = 0$.

*Para o conjunto S ser **Linearmente Dependente (L . D)**, basta existir pelo menos um $\alpha_i \neq 0$, tal que*

$$\alpha_1.v_1 + \alpha_2.v_2 + \cdots + \alpha_m.v_m = (0, 0, \ldots, 0).$$

Dos conceitos de dependência e independência linear decorrem as seguintes propriedades dadas nas seguintes proposições.

Proposição 3.3. *(1) Se um conjunto finito S de elementos de $\mathbb{Z}_p^n$ contém o elemento vetor nulo $(0, 0, \ldots, 0)$, então S é L . D.*

(2) Se $S = \{v\} \subset V$, com $v \neq (0, 0, \ldots, 0)$, então S é L . I.

(3) Se $S = \{v_1, v_2, \ldots, v_m\} \subset \mathbb{Z}_p^n$, é L . D, então um dos seus elementos é combinação linear dos demais.

Demonstração. Para (1) considere um conjunto contendo o vetor nulo,

$S = \{(0, 0, \ldots, 0), v_2, \ldots, v_m\} \subset \mathbb{Z}_p^n$. Então temos, $\alpha.(0, 0, \ldots, 0) + 0.v_2, \ldots, o.v_m = (0, 0, \ldots, 0)$, para todo $\alpha \in \mathbb{Z}_p$. Assim, existe pelo menos um $\alpha \neq 0$ que satisfaz a equação, o que mostra que o sistema S é L.D. .

Para (2), considere a equação

$$\alpha.v = (0, 0, \ldots, 0)$$

Como $v \neq (0, 0, \ldots, 0)$, então temos $\alpha = 0$, pois $\mathbb{Z}_p$ é um corpo. Assim, S é L.I.

Para (3), se admitirmos que S é L . D., por definição temos que existem escalares $\alpha_1, \alpha_2, \ldots, \alpha_m$ não todos nulos, tais que: $\alpha_1.v_1 + \cdots + \alpha_i.v_i + \cdots + \alpha_m.v_m = (0, 0, \ldots, 0)$

Supondo que $\alpha_i \neq 0$, temos o inverso multiplicativo de α_i. Multiplicando todos os elementos por α_i^{-1} obtemos:

$\alpha_i^{-1}.\alpha_1.v_1 + \cdots + \alpha_i^{-1}.\alpha_i.v_i + \cdots + \alpha_i^{-1}.\alpha_m.v_m = (0, 0, \ldots, 0)$.

Então

$\alpha_i^{-1}.\alpha_1.v_1 + \cdots + v_i + \cdots + \alpha_i^{-1}.\alpha_m.v_m = (0, 0, \ldots, 0)$.

Logo,

$v_i = -\alpha_i^{-1}.\alpha_1.v_1 - \cdots - \alpha_i^{-1}.\alpha_m.v_m$

Ou seja, o elemento v_i é combinação linear dos outros elementos de S. □

Proposição 3.4. *(1) Sejam S_1 e S_2 subconjuntos finitos e não vazios de $\mathbb{Z}_p^n$. Se S_1 é L.D. e $S_1 \subset S_2$, então S_2 também e L.D.*

(2) Sejam S_1 e S_2 subconjuntos finitos e não vazios de $\mathbb{Z}_p^n$. Se S_2 é L . I. e $S_1 \subset S_2$, então S_1 também e L . I.

(3) Se $S = \{v_1, v_2, \ldots, v_n\} \subset \mathbb{Z}_p^n$ é L . I. e para algum $v \in \mathbb{Z}_p^n$ tivermos que $S \cup \{v\} = \{v_1, v_2, \ldots, v_n, v\}$ é L . D. então o elemento v é combinação linear dos elementos de S.

Demonstração. Para (1) escrevemos os subconjuntos $S_1 = \{v_1, v_2, \ldots, v_j\}$ e

$S_2 = \{v_1, \ldots, v_j, \ldots, v_n\}$, não vazios de $\mathbb{Z}_p^n$. Ora, se S_1 é L . D, então existem escalares $\alpha_1, \alpha_2, \ldots, \alpha_j$ nem todos nulos, de forma que:

$$\alpha_1.v_1 + \alpha_2.v_2 + \cdots + \alpha_j.v_j = (0, 0, \ldots, 0)$$

E completando essa equação com zeros até utilizarmos todos os v_n elementos para descrever S_2 temos:

$$\alpha_1.v_1 + \cdots + \alpha_i.v_j + 0.v_{j+1} + \cdots + 0.v_n = (0, 0, \ldots, 0)$$

Podemos notar que nem todos os escalares desta igualdade serão nulos, portanto temos que S_2 é L . D.

Para (2), note que, das observações em (1), caso S_1 fosse L . D., então teríamos que S_2 também é L . D. Entretanto, S_2 é L . I, assim S_1, subconjunto com menos elementos, só pode ser L . I.

Para (3) se temos $S \cup \{v\}$ um conjunto L . D . Então, existem escalares $\alpha_1, \alpha_2, \cdots, \alpha_m, \alpha$, não todos nulos tais que:

$$\alpha_1.v_1 + \alpha_2.v_2 + \cdots + \alpha_m v_m + \alpha v = (0, 0, \ldots, 0) \tag{3.1}$$

Podemos verificar que um dos escalares não nulos é α, uma vez que se $\alpha = 0$ teríamos:

$$\alpha_1.v_1 + \cdots + \alpha_m v_m = (0, 0, \ldots, 0)$$

Mas, como S é L . I. esta última igualdade só vale se $\alpha_1 = \cdots = \alpha_m = 0$. Porém, neste caso teríamos $\alpha_1 = \cdots = \alpha_m = \alpha = 0$ e daí o conjunto $v_1, \ldots, v_m, v$ seria L . I., oque contradiz a hipótese dele de ser L . D. Assim, temos que $\alpha \neq 0$.

Podemos então multiplicar por α^{-1} a equação 3.1:

$\alpha^{-1}.\alpha_1.v_1 + \alpha^{-1}.\alpha_2.v_2 + \cdots + \alpha^{-1}.\alpha_m.v_m + v = (0, 0, \ldots, 0)$.

Logo

$v = -\alpha^{-1}.\alpha_1.v_1 - \alpha^{-1}.\alpha_2.v_2 - \cdots - \alpha^{-1}.\alpha_m.v_m$.

O que mostra que v é combinação linear dos elementos de S, ou seja, $v \in [S]$ □

Proposição 3.5. *Se um elemento é combinação linear dos demais, ele pode ser extraído do conjunto de geradores, sem mudar do subespaço gerado, ou seja:*

Se $S = \{v_1, \ldots, v_j, \ldots, v_m\} \subset [S - \{v_j\}]$, *então* $[S] = [S - \{v_j\}]$.

Demonstração. Suponhamos que v_1 é combinação linear dos demais elementos de S, então ele pode ser escrito como:

$$v_1 = \beta_2.v_2 + \cdots + \beta_m.v_m.$$

Considere um elemento $x \in [S]$, isto é, x pode ser escrito como combinação linear dos elementos de S:

$$\begin{aligned} x &= \alpha_1.v_1 + \alpha_2.v_2 + \cdots + \alpha_m.v_m \\ &= \alpha_1 \left(\beta_2.v_2 + \cdots + \beta_m.v_m\right) + \alpha_2.v_2 + \cdots + \alpha_m.v_m \\ &= \left(\alpha_1.\beta_2 + \alpha_2\right).v_2 + \cdots + \left(\alpha_1.\beta_m + \alpha_m\right).v_m \end{aligned}$$

O que mostra que x é combinação linear dos elementos de $[S - \{v_1\}]$, ou seja, $x \in [S - \{v_1\}]$.

Daí, supondo que $x \in [S - \{v_1\}]$ temos que $x = 0.v_1 + \gamma_2.v_2 + \cdots + \gamma_m.v_m$, ou seja $x \in [S]$.

Portanto, qualquer elemento de $[S]$ pertence a $[S - \{v_1\}]$ e vice versa, permitindo que o elemento v_1 possa ser extraído do conjunto de geradores. □

Exemplo 3.8. *Os elementos* $v_1 = (1, 2)$ *e* $v_2 = (3, 6)$ *do espaço vetorial* $\mathbb{Z}_7^2$ *são linearmente dependente.*

Basta verificar que: $(3, 6) = 3.(1, 2)$, ou seja, $v_2 = 3v_1$. Assim, v_2 é combinação linear de v_1.

Exemplo 3.9. *O subconjunto $S = \{(1,1,0,0),(0,1,0,2),(0,0,1,0),(0,2,4,4)\} \subset \mathbb{Z}_5^4$ é linearmente dependente.*

De fato, temos que:

$$0.(1,1,0,0) + 2.(0,1,0,2) + 4.(0,0,1,0) = (0,2,4,4)$$

ou seja, um dos vetores é combinação linear dos demais, assim o subconjunto é L . D.

Pela Proposição 3.5 podemos extrair $(0,2,4,4)$ do conjunto de geradores e temos:

$$[S] = [(1,1,0,0),(0,1,0,2),(0,0,1,0)]$$

, ou seja, S é gerado por esses três vetores.

Exemplo 3.10. *O conjunto $\{(1,0),(0,1)\}$ em $\mathbb{Z}_2^2$ é linearmente independente.*

De fato, a equação: $\alpha_1(1,0) + \alpha_2(0,1) = (0,0)$ só vale para $\alpha_1 = \alpha_2 = 0$. Assim, os vetores $(1,0),(0,1)$ são L . I.

3.6 Base e Dimensão de Subespaços de $\mathbb{Z}_p^n$

Definição 3.6. *Seja V um subespaço de um espaço vetorial $\mathbb{Z}_p^n$. Uma **base** de V é um conjunto finito $B = \{v_1,\ldots,v_n\} \subset \mathbb{Z}_p^n$ de elementos de V, tal que B é linearmente independente(L.I.) e gerador do espaço vetorial V, ou seja, qualquer elemento de V pode ser escrito como combinação linear dos elementos de B.*

Exemplo 3.11. *Tem-se que $\{(1,0),(0,1)\}$ é uma base para o espaço vetorial $\mathbb{Z}_2^2$, que tambem é conhecida como **base canônica** do $\mathbb{Z}_2^2$.*

Teorema 3.1. *Dado V um subespaço de $\mathbb{Z}_p^n$ e um subconjunto contendo m elementos que geram V, então é possível, a partir desses elementos, extrair uma base para V.*

Demonstração. Considere $B = \{v_1,\ldots,v_m\} \subset V$ o conjunto de elementos que geram V.

Se $B = \{v_1,\ldots,v_m\}$ é composto de vetores L.I. , por definição eles geram uma base, e não é necessário realizar demonstração .

Por outro lado, se o mesmo conjunto de vetores for L.D., então existem escalares $\alpha_1,\ldots,\alpha_m$ tal que seja válida a igualdade:

$$\alpha_1.v_1 + \alpha_2.v_2 + \cdots + \alpha_m.v_m = (0,0,\ldots,0)$$

Suponhamos, sem perda de generalidade, $\alpha_m \neq 0$. Podemos multiplicar todos os α_i's por α_m^{-1}, obtendo

$$\alpha_m^{-1}.\alpha_1.v_1 + \alpha_m^{-1}.\alpha_2.v_2 + \cdots + \alpha_m^{-1}.\alpha_m.v_m = (0, 0, \ldots, 0)$$

$$v_m = -\alpha_m^{-1}.\alpha_1.v_1 - \alpha_m^{-1}.\alpha_2.v_2 - \cdots + \alpha_m^{-1}.\alpha_{m-1}.v_{m-1}$$

Na prática significa dizer que v_m é combinação linear dos demais elementos. Pela Proposição 3.5, mesmo extraindo o vetor v_m, V continua sendo gerado por $\{v_1, \cdots, v_{m-1}\}$. E repetindo o processo uma quantidade finita de vezes, temos um subconjunto $\{v_1, \ldots, v_r\}$ com r elementos ($r \leq m$) que ainda geram V e são L.I., ou seja, formam uma base de V. Portanto, é possivel extrair um conjunto de vetores que geram uma base para V. □

Teorema 3.2. *Se $B = \{v_1, \ldots, v_m\}$ é uma base para o subespaço vetorial V, com m vetores, qualquer outro subconjunto de vetores L.I. que gera V tem no máximo m elementos.*

Demonstração. Considere $w = \{w_1, w_2, \cdots, w_m, \cdots, w_k\} \subset V$, com $k > m$. Mostraremos que W é linearmente dependente (L.D.).

Como $\{v_1, v_2, \cdots, v_m\}$ é base de V e cada $w_i \in V$, segue que

$$w_i = \alpha_{1i}.v_1 + \alpha_{2i}.v_1 + \cdots + \alpha_{mi}.v_m, \ i = 1, 2, \cdots, k \tag{3.2}$$

Considere também a combinação linear nula

$$\beta_1.w_1 + \cdots + \beta_k.w_k = (0, 0, \ldots, 0), \text{ com escalares } \beta_1, \cdots, \beta_k \in \mathbb{Z}_p \tag{3.3}$$

Podemos reescrever a equação de 3.2 em 3.3, substituindo cada $w_i = \alpha_{1i}.v_1 + \alpha_{2i}.v_1 + \cdots + \alpha_{ki}.v_k$, obtendo

$$\beta_1.w_1 + \cdots + \beta_k.w_k = (0, 0, \ldots, 0)$$
$$\beta_1.(\alpha_{11}.v_1 + \alpha_{21}.v_1 + \cdots + \alpha_{m1}.v_m) + \cdots + \beta_k.(\alpha_{1k}.v_1 + \alpha_{2k}.v_1 + \cdots + \alpha_{mk}.v_m) = (0, 0, \ldots, 0),$$
ou seja
$$v_1.(\beta_1.\alpha_{11} + \cdots + \beta_k.\alpha_{1k}) + \cdots + v_m.(\beta_1.\alpha_{m1} + \cdots + \beta_k.\alpha_{mk}) = (0, 0, \ldots, 0)$$

Como $\{v_1, \ldots, v_n\}$ é uma base para V, então este conjunto é L . I. Assim temos:

$$\begin{cases} \beta_1.\alpha_{11} + \beta_2.\alpha_{12} + \cdots + \beta_k.\alpha_{1k} = 0 \\ \beta_1.\alpha_{m1} + \beta_2.\alpha_{r2} + \cdots + \beta_k.\alpha_{mk} = 0 \end{cases}$$

Obtemos um sistema linear homogêneo com m equações e k incógnitas $\beta_1, \ldots, \beta_k$. E como o número de equações é menor que o número de incógnitas, o sistema admite solução não trivial. Assim, existem escalares não todos nulos $\beta_1, \ldots, \beta_k$, tais que:

$$\beta_1.w_1 + \cdots + \beta_k.w_k = (0, 0, \ldots, 0)$$

Portanto, $W = \{w_1, w_2, \ldots, w_k\}$ é L . D. Assim, qualquer conjunto com mais de k elementos é L .D., ou seja, qualquer conjunto L . I. possui no máximo m elementos.

□

Teorema 3.3. *Se V um subespaço de $\mathbb{Z}_p^n$, então duas bases quaisquer, A e B tem sempre o mesmo número de elementos.*

Demonstração. Sejam as bases $A = \{v_1, \ldots, v_m\}$ e $B = \{w_1, \ldots, w_k\}$ geradores de $V \subset \mathbb{Z}_p^n$. Se considerarmos que A gera V, então podemos dizer que é formado por vetores L.I..

Se $m < k$ e k é o número de elementos da base B, o conjunto B não será L.I. Contradizendo o Teorema 3.2. De modo análogo, não podemos ter $m > k$, pois o conjunto A também é L.I. Portanto, $m = k$. e assim, qualquer base de V tem o mesmo número de elementos. □

Segue do Teorema 3.3 a definição que daremos a seguir, que é o conceito de dimensão.

Definição 3.7. *A **dimensão** de um espaço vetorial $V \subset \mathbb{Z}_p^n$, $V \neq \varnothing$ é o número de elementos de uma base, cuja notação é $dim(V)$. Caso V seja formado apenas pelo vetor nulo, ou seja, $V = \{(0, 0, \ldots, 0)\}$ então sua dimensão $dim(V) = 0$.*

Exemplo 3.12. *Temos que $\{(1, 0), (0, 1)\}$ é uma base para o espaço vetorial $\mathbb{Z}_2^2$. Sua dimensão será $dim(\mathbb{Z}_2^2) = 2$.*

Teorema 3.4 (Teorema do Completamento)**.** *Seja um subespaço vetorial $V \subset \mathbb{Z}_p^n$. Qualquer conjunto finito de elementos linearmente independentes $(L.I)$ de V pode ser completado até formar uma base para V.*

Demonstração. Seja $dim(V) = m$ e $v_1, \ldots, v_r$ elementos L . I. em V pelo Teorema 3.3 $r \leq m$. Se os elementos $v_1, \ldots, v_r$ geram V, então $\{v_1, \ldots, v_r\}$ já é uma base. Agora, se isso não ocorre, então existe um $v_{r+1} \in V$ que não é combinação linear de $v_1, \ldots, v_r$ então $\{v_1, \ldots, v_r, v_{r+1}\}$ ainda é L . I., pois caso contrário a equação: $\alpha_1.v_1 + \ldots + \alpha_r.v_r + \alpha_{r+1}.v_{r+1} = (0, 0, \ldots, 0)$ seria verdadeira para $\alpha_{r+1} \neq 0$ e, então poderíamos multiplicar a equação por α_{r+1}^{-1} e determinar v_{r+1}, como combinação linear de $v_1, \ldots, v_r$, o que é uma contradição.

Se $\{v_1, \ldots, v_r, v_{r+1}\}$ gera V, então é uma base. Caso contrário, repetimos o processo até completar a base. Esse processo termina em um número finito de passos, uma vez que

pelo Teorema 3.3, não podemos ter um conjunto L . I. com mais de m elementos, já que $dim(V) = m$. □

Teorema 3.5. *Sejam $U, W \subset \mathbb{Z}_p^n$ subespaços vetoriais. Então, a dimensão do vetor soma $U + W$ é obtida pela relação*

$$dim(U + W) = dim(U) + dim(W) - dim(U \cap W).$$

onde $U \cap W$ reúne os elementos comuns aos dois subespaços.

Demonstração. Tomemos $\{v_1, \ldots, v_r\}$ os elementos de uma base para $U \cap W$. Logo $(dim(U \cap W) = r)$ pelo Teorema 3.4, podemos completar esse conjunto até obter uma base para U.

Seja $\{v_1, \ldots, v_r, u_1, \ldots, u_m\}$ esta base para U , $(dim(U) = r + m)$.

De maneira análoga, podemos completar $\{v_1, \ldots, v_r\}$ para obter uma base para W. Seja $\{v_1, \ldots, v_r, w_1, \ldots, w_k\}$ base para W, sua dimensão será $(dim(W) = r + k)$. Como $U = \langle v_1, \ldots, v_r, u_1, \ldots, u_m\rangle$ e $W = \langle v_1, \ldots, v_r, w_1, \ldots, w_k\rangle$ então temos que:

$$U + W = \langle v_1, \ldots, v_r, u_1, \ldots, u_m, w_1, \ldots, w_k\rangle$$

E provemos que esse conjunto é linearmente independente.

Considere a equação:

$$\sum_{i=1}^{r} a_i v_i + \sum_{j=1}^{m} b_j u_j + \sum_{l=1}^{k} c_l w_l = (0, 0, \ldots, 0) \tag{3.4}$$

Logo,

$$-\sum_{l=1}^{k} c_l w_l = \sum_{i=1}^{r} a_i v_i + \sum_{j=1}^{m} b_j u_j \tag{3.5}$$

O primeiro termo da última igualdade é uma combinação linear de elementos de W, logo pertence a W, o segundo termo é uma combinação linear de elementos de U, logo pertence a U.

Como vale a igualdade, então temos que o primeiro termo também pertence a U, assim:

$$\sum_{l=1}^{k} c_l w_l \in U \cap W.$$

Podemos escrevê-lo como combinação linear dos elementos da base de $U \cap W$.

$$\sum_{l=1}^{k} c_l w_l = \sum_{l=1}^{r} \alpha_i v_i \Leftrightarrow \sum_{l=1}^{k} c_l w_l - \sum_{i=1}^{r} \alpha_i v_i = (0, 0, \ldots, 0) \tag{3.6}$$

Como $\{v_1, \ldots, v_r, w_1, \ldots, w_k\}$ é L . I., pois é uma base para W, temos $c_1 = \ldots = c_k = \alpha_1 = \ldots = \alpha_r = 0$

Na equação 3.4, obtemos:

$\sum_{i=1}^{r} a_i v_i + \sum_{j=1}^{m} b_j u_j + \sum_{l=1}^{k} c_l w_l = (0, 0, \ldots, 0)$

Como $\{v_1, \ldots, v_r, u_1, \ldots, u_m\}$ é L.I., pois é uma base de U, temos que $a_1 = \ldots = a_r = b_1 = \ldots = b_m = 0$.

Portanto $\{v_1, \ldots, v_r, u_1, \ldots, u_m, w_1, \ldots, w_k\}$ é uma base de $U + W$. Logo:

$dim(U + W) = r + m + r + k - r = (r + m) + (r + k) - r$, ou seja,

$dim(U + W) = dim(U) + dim(W) - dim(U \cap W)$. □

Teorema 3.6. *Seja $B = \{v_1, \ldots, v_m\}$* ***uma base ordenada de*** *V, com $V \subset \mathbb{Z}_p^n$ e os elementos de B seguindo uma ordem fixa. Então, todo elemento de V pode ser escrito de modo* ***único*** *como* ***combinação linear*** *dos vetores da base.*

Demonstração. Considere $v \in V$ e que $v = \sum_{j=1}^{m} a_j v_j$ e $v = \sum_{j=1}^{m} b_j v_j$. Fazendo $v - v$ temos:

$$v - v = (0, 0, \ldots, 0) = \sum_{j=1}^{m} a_j v_j - \sum_{j=1}^{m} b_j v_j,$$

ou seja,

$$\sum_{j=1}^{m} (a_j - b_j) v_j = (0, 0, \ldots, 0)$$

O que implica que $a_j - b_j = 0$ se, e só se $a_j = b_j \forall j$, uma vez que $\{v_1, \ldots, v_m\}$ é L . I. Logo, o elemento $v \in V$ é escrito de modo único como combinação linear dos elementos da base ordenada de V. □

Como se vê, com a ideia de que subespaços de $\mathbb{Z}_p^n$ podem ser gerados a partir de alguns vetores, é possível, a partir de alguns vetores, determinar outros, gerando um conjunto. Vejamos na seção seguinte o procedimento para determinação destes geradores.

3.7 Processo Prático Para Determinar Bases de Subespaços de $\mathbb{Z}_p^n$

Primeiro Processo

Seja $W \subset \mathbb{Z}_p^n$ o subespaço vetorial gerado pelo conjunto de vetores $\{a_1, a_2, \ldots, a_m\}$, ou seja,

$$W = [a_1, a_2, \ldots, a_m].$$

É possível obter uma base de W, contida em $\{a_1, a_2, \ldots, a_m\}$, seguindo o seguinte procedimento:

Início do procedimento:

i Construir uma matriz M cujos vetores $a_1, a_2, \ldots, a_m$ são as colunas dessa matriz, ou seja, ela tem n linhas por m colunas;

$$M = [a_1, a_2, \ldots, a_m]$$

ii Realizar, nessa matriz, operações de escalonamento nas linhas, obtendo M'.

iii Observar na matriz M' quais colunas são colunas pivô, ou seja, contém um elemento não nulo e suas posições a esquerda, na mesma linha, mas como índice de coluna menor, contém zeros, como (LAY, 2007) ilustra representativamente abaixo:

iv Os vetores que geram a base a partir do subespaço W são os vetores a_i nos quais o respectivo índice na matriz escalonada apresenta pivô.

$$\begin{bmatrix} \blacksquare & a_0 & a_1 & a_2 \\ 0 & \blacksquare & a_3 & a_4 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0, \end{bmatrix} \quad \begin{bmatrix} 0 & \blacksquare & a_1 & a_2 & a_3 & a_4 & a_5 & a_6 & a_7 & a_8 \\ 0 & 0 & 0 & \blacksquare & a_9 & a_{10} & a_{11} & a_{12} & a_{13} & a_{14} \\ 0 & 0 & 0 & 0 & \blacksquare & a_{15} & a_{16} & a_{17} & a_{18} & a_{19} \\ 0 & 0 & 0 & 0 & 0 & \blacksquare & a_{20} & a_{21} & a_{22} & a_{23} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \blacksquare & a_{24}. \end{bmatrix}$$

Figura 3.1 – Exemplo de colunas pivô.

Fonte: David C. Lay/Algebra Linear e Aplicações.

Fim do procedimento.

Para melhor compreensão, vejamos o exemplo abaixo:

Exemplo 3.13. *Seja S subespaço de $\mathbb{Z}_2^5$ gerado pelo conjunto $A = \{a_1, a_2, a_3, a_4\}$ onde:*

$$a_1 = (1,0,1,1,0)\ ;\ a_2 = (1,1,0,0,0)\ ;\ a_3 = (0,1,1,1,0)\ ;\ a_4 = (1,0,0,1,0)$$

Vamos executar os passos do procedimento para encontrar uma base S contida em A. Fazemos isso considerando a matriz M cujas colunas são os vetores de A:

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

Escalonamos a matriz M usando operações de linha

$$M = \begin{pmatrix} 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix} \xrightarrow[L_4 = L_4 + L_1]{L_3 = L_3 + L_1} \begin{pmatrix} 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \xrightarrow[L_4 = L_4 + L_2]{L_3 = L_3 + L_2} \begin{pmatrix} \mathbf{1} & 1 & 0 & 1 \\ 0 & \mathbf{1} & 1 & 0 \\ 0 & 0 & 0 & \mathbf{1} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

Note que as colunas pivô, em destaque, são respectivamente as colunas 1, 2 e 4. Daí concluímos que a base S contida em A é formada pelos vetores.

$$a_1 = (1,0,1,1,0)\ ;\ a_2 = (1,1,0,0,0)\ ;\ a_4 = (1,0,0,1,0)\ ;$$

Apenas pra comprovarmos que os vetores a_1, a_2, a_4 geram uma base, do Teorema 3.1 vamos verificar se eles são L . I. De fato:

$\alpha_1.a_1 + \alpha_2.a_2 + \alpha_4.a_4 = (0,0,0,0)$

$\alpha_1.(1,0,1,1,0) + \alpha_2.(1,1,0,0,0) + \alpha_4.(1,0,0,1,0) = (0,0,0,0)$

$(\alpha_1 + \alpha_2 + \alpha_4, \alpha_2, \alpha_1, \alpha_1 + \alpha_4, 0) = (0,0,0,0)$

E organizando os sistemas de equações para a obtenção dos α_{is} temos

$$\Rightarrow \begin{cases} \alpha_1 + \alpha_2 + \alpha_4 = 0 \\ \alpha_2 = 0 \\ \alpha_1 = 0 \\ \alpha_1 + \alpha_4 = 0 \Rightarrow 0 + \alpha_4 = 0 \Rightarrow \alpha_4 = 0 \end{cases}$$

De fato, os vetores são L.I. e podem gerar uma base.

Como abordado em (TAUSK, 2008), o método é eficaz porque se baseia em três fatos:

Afirmação 1. Se uma matriz M' é obtida de uma matriz M através de operações de escalonamento (operações de linha) então as relações lineares satisfeitas pelas colunas de M são as mesmas que as relações lineares satisfeitas pelas colunas de M'. Perceba que o conjunto solução de um sistema linear homogêneo não se altera quando escalonamos (com operações de linha) a sua matriz de coeficientes e que as colunas $a_1, a_2, \cdots, a_m$ da matriz M formam uma relação linear do tipo

$$\alpha_1.a_1 + \alpha_2.a_2 + \cdots + \alpha_m.a_m = 0$$

com $\alpha_1, \alpha_2, \cdots, \alpha_m$ soluções do sistema com matriz de coeficientes M.

Afirmação 2. Se M' é uma matriz escalonada $n \times m$ então as colunas pivô de M' formam um subconjunto linearmente independente de $\mathbb{Z}_p^n$.

Se olharmos as colunas pivô na matriz escalonada, vamos descobrir que os vetores formados por elas são L.I.

Afirmação 3. Se M' é uma matriz escalonada $n \times m$ então uma coluna de M' que não contém pivô pertence ao subconjunto de $\mathbb{Z}_p^n$ gerado pelas colunas-pivô de M' que a antecedem. Olhando com atenção o exemplo realizado acima, note que na matriz M', a terceira coluna não contém pivô, mas que pode ser resultado da operação:

$$a_3^{'} = a_1^{'} + a_2^{'}$$

Portanto, o processo descrito é uma forma de determinar bases de subespaços de $\mathbb{Z}_p^n$.

Segundo Processo

No livro "Álgebra Linear e aplicações" de (CALLIOLI et al., 2000), em sua página 80, é descrito um outro procedimento para se obter uma base de um subespaço $\mathbb{Z}_p^n$, dado por seus geradores ou onde seja possível encontrar esses geradores.

Para tanto, ele se baseia em três afirmações.

Sendo $W = [a_1, a_2, \ldots, a_r] \subset \mathbb{Z}_p^n$, ou seja, subespaço de $\mathbb{Z}_p^n$ valem:

- Se permutarmos dois vetores que lá estão presentes, não alteramos o subespaço gerado, ou seja,

$$W = [a_1, a_2, \ldots, a_i, \cdots, a_j, \ldots, a_r] = [a_1, a_2, \ldots, a_j, \cdots, a_i, \ldots, a_r]$$

- Para todo número $\alpha \in \mathbb{Z}_p$ têm-se que

$$W = [a_1, a_2, \ldots, a_i, \cdots, a_j + \alpha a_i, \ldots, a_r]$$

- Se $a_1, a_2, \cdots, a_r$ se apresentam na forma escalonada, com o número de zeros iniciais de a_2 , maior que o de a_1, então os vetores $a_1, \cdots, a_r$ formarão um conjunto L.I e $dimW = r$.

Vamos seguir os seguintes passos:

Início do Procedimento

i Construir uma matriz M onde os vetores $\{a_1, a_2, \cdots, a_r\}$ são as Linhas de M.

$$M = \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_r \end{pmatrix}$$

ii Realizar nessa matriz M operações de escalonamento em linhas, obtendo M'.

iii As linhas não nulas de M' são os vetores da base de W.

Fim do procedimento.

Uma ressalva importante é que os vetores da base encontrada não necessariamente estão contidos em $\{a_1, a_2, \cdots, a_r\}$.

Baseado neste procedimento descrito acima,vamos realizar um exemplo para a determinação de uma base.

Exemplo 3.14. *Seja W subespaço de $\mathbb{Z}_2^5$ gerado pelo conjunto $A = \{a_1, a_2, a_3, a_4\}$*

onde:

$a_1 = (1,0,1,1,0)$; $a_2 = (1,1,0,0,0)$; $a_3 = (0,1,1,1,0)$; $a_4 = (1,0,0,1,0)$

Realizando as operações do procedimento acima relatado e executando o escalonamento temos:

$$M = \begin{pmatrix} 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 \end{pmatrix} \xrightarrow[L_4 = L_4 + L_1]{L_2 = L_2 + L_1} \begin{pmatrix} 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix} \xrightarrow{L_3 = L_3 + L_2} \begin{pmatrix} 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

$$\xrightarrow{L_3 \leftrightarrow L_4} \begin{pmatrix} 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

As linhas não nulas da matriz $M' = \begin{pmatrix} 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$ serão os vetores da base de W. E para mostrar que $b_1 = (1,0,1,1,0), b_2 = (0,1,1,1,0)$ e $b_3 = (0,0,1,0,0)$ são geradores de uma base sobre $\mathbb{Z}_2^5$, vejamos se tais vetores são Linearmente Independentes.

Sejam $\beta_1, \beta_2, \cdots, \beta_r \in \mathbb{Z}_2$ e $b_1, b_2, \cdots, b_r$ vetores sobre $\mathbb{Z}_2^5$, eles serão L.I. quando:

$$\beta_1.b_1 + \beta_2.b_2 + \cdots + \beta_r.a_r = 0.$$

De fato:

$\beta_1.b_1 + \beta_2.b_2 + \beta_3.b_3 = (0,0,0,0,0)$

$\beta_1.(1,0,1,1,0) + \beta_2.(0,1,1,1,0) + \beta_3.(0,0,1,0,0) = (0,0,0,0,0)$

$(\beta_1, \beta_2, \beta_1 + \beta_2 + \beta_3, \beta_1 + \beta_2, 0) = (0,0,0,0,0)$

E organizando os sistemas de equações para a obtenção dos β_{is} temos

$$\Rightarrow \begin{cases} \beta_1 = 0 \\ \beta_2 = 0 \\ \beta_1 + \beta_2 + \beta_3 = 0 \\ \beta_1 + \beta_2 = 0 \end{cases}$$

ou seja $\beta_1 = \beta_2 = \beta_3 = 0$

Portanto, os vetores são L.I. e podem gerar uma base.

Como C é um espaço vetorial, se tivermos todos os elementos de C, observe que em ambos os processos obtemos os mesmos resultados, porém o processo citado em Callioli é o mais conhecido.

3.7.1 A Função Produto Interno em $\mathbb{Z}_p^n$

Considere a função:

$\langle ., . \rangle : \mathbb{Z}_p^n \times \mathbb{Z}_p^n \to \mathbb{Z}_p^n$

$\langle u, v \rangle = a_1.b_1 + a_2.b_2 + \cdots + a_n.b_n$, onde $u = (a_1, a_2, \ldots, a_n)$ e $v = (b_1, b_2, \ldots, b_n)$.

Essa função é um quase produto interno no espaço vetorial $\mathbb{Z}_p^n$.

Proposição 3.6. *A função quase produto interno satisfaz as seguintes propriedades:*

i) $\langle u, v \rangle = \langle v, u \rangle$, *para todo* $u, v \in \mathbb{Z}_p^n$

ii) $\langle u, v + w \rangle = \langle u, v \rangle + \langle u, w \rangle$ *para todo* $u, v, w \in \mathbb{Z}_p^n$;

iii) $\alpha\langle u, v\rangle = \langle \alpha u, v\rangle = \langle u, \alpha v\rangle$ *para todo* $\alpha \in \mathbb{Z}_p$ *e* $u, v \in \mathbb{Z}_p^n$.

Demonstração. i) Observe que é verdade, pois:

$$\langle u, v\rangle = a_1.b_1 + a_2.b_2 + \cdots + a_n.b_n = b_1.a_1 + b_2.a_2 + \cdots + b_n.a_n = \langle v, u\rangle \ .$$

ii) De fato, pois

consideremos $v + w = (b_1 + c_1, b_2 + c_2, \ldots, b_n + c_n)$ e obtemos.

$\langle u, v + w\rangle = a_1.(b_1 + c_1) + a_2.(b_2 + c_2) + \cdots + a_n.(b_n + c_n) =$

$= (a_1.b_1 + a_1.c_1) + (a_2.b_2 + a_2.c_2) + \cdots + (a_n.b_n + a_n.c_n)$

$= (a_1.b_1 + a_2.b_2 + \cdots + a_n.b_n) + (a_1.c_1 + a_2.c_2 + \cdots + a_n.c_n)$

$= \langle u, v\rangle + \langle u, w\rangle$

iii) De fato, pois

$\langle \alpha u, v\rangle = (\alpha a_1).b_1 + (\alpha a_2).b_2 + \cdots + (\alpha a_n).b_n$

$\alpha(a_1.b_1 + a_2.b_2 + \cdots + a_n.b_n) = \alpha\langle u, v\rangle.$

□

Observação: A operação definida na proposição 3.6 possui propriedades que são usuais no conceito de produto interno. Todavia a propriedade $\langle u, u\rangle = 0$ se, e somente se $u = 0$ não é satisfeito em $\mathbb{Z}_p^n$. Por exemplo, considere $u = (1, 1) \in \mathbb{Z}_2^2$ e observe que

$$\langle u, u\rangle = 1.1 + 1.1 = 2 \text{ e } u \neq 0.$$

Apresentadas as propriedades do Espaço Vetorial $\mathbb{Z}_p^n$, é possivel definir os conceitos de códigos, em especial os lineares, assim compreender melhor os algoritmos envolvidos na codificação e os bits de vericação/paridade na criação do mesmo.

4 Códigos Lineares: Conceitos e a Métrica de Hamming

Desde a criação dos primeiros algoritmos computacionais e a programação de máquinas locais e remotas, ou ainda a necessidade de transmissão para grandes distâncias, havia a preocupação sobre a garantia de que a informação transmitida fosse facilmente entendida dos dois lados da transmissão.

Fazendo uma analogia rasa, de fácil compreensão, imaginemos a atividade lúdica " telefone sem fio ". Escolhido o meio por onde a mensagem foi repassada, se em algum momento a mensagem se perdeu, parcial ou totalmente, desde a sua emissão, teremos um destinatário que não entenderá a mensagem e pedirá novamente, ou pior ainda, decifrará de modo errado a mensagem.

A aplicabilidade dos algoritmos nas comunicações e a integridade de que a informação chegará inteira, confiável e com menor possibilidade de perda, fez com que estudos nesse sentido avançassem, principalmente nos sistemas de comunicações.

(ALENCAR, 2007) afirma:

> Todos os sistemas de comunicações em pequenas ou altas proporções, estão sujeitos a pertubações causadas pela ação do ruído presente no canal. Este efeito pode acasionar a ocorrência de erros nas mensagens que são recebidas por um sistema receptor. Genericamente são chamados de erros aleatórios, quando este tipo de incidência ocorre de maneira esporádica e independente; já quando ocorrem em surtos, ou seja, em sequência de vários erros consecutivos, são denominados erros em rajadas (ALENCAR,2007)

Como precursores na teoria de códigos, podemos enunciar que eles acompanham o salto computacional no ínicio dos anos 40, onde a tecnologia era utilizada para tarefas complexas e posteriormente, os sistemas de controle digitais(SCD).

Segundo (MILIES, 2009) :

> A teoria teve início na decada de quarenta quando os computadores eram máquinas muito caras e apenas instituições de grande porte como o governo ou as universidades tinham condições de mantê-la. Eles usando-os para executar tarefas numéricas complexas, com calcular o órbita precisa de Marte ou fazer a avaliação estatística de um censo.

Quem primeiro se preocupou de forma sistemática com esse assunto foi Richard Wesley Hamming ($1915 - 1998$). Em 1950, ele publicou, no Bell System Technical Journal de abril, um trabalho com o título de: Error Detecting and Error Correcting Codes, que

pode ser considerado a primeira sistematização teórica sobre detecção e correção de erros ((ROCHOL, 2012), p, 249).

Além de Hamming, Shannon (com a sua teoria de códigos e a Teoria da Informação), Golay (cujo código foi usado pela espaçonave Voyager(em 1980 e 1981) para transmitir fotografias coloridas de Júpter e Saturno), e outros cientistas foram os grandes pioneiros que iniciaram o trabalho com este assunto e desenvolveram estudos e ideias que são usadas em nosso dia a dia, como por exemplo, a comunicação móvel (telefones celular), aparelhos de armazenamento de imagens digitais, internet e rádio entre outras utilidades.

As técnicas de paridade foram as primeiras técnicas a serem desenvolvidas e utilizadas. Devido sua simplicidade, ainda são largamente utilizadas em protocolos de comunicação baseados em caracteres.

Basicamente, na transmissão de uma informação, nos preocupamos com o meio por onde ela passará, o tamanho da quantidade (ou pacote) de informação a ser transmitida, e o algoritmo a ser utilizado.

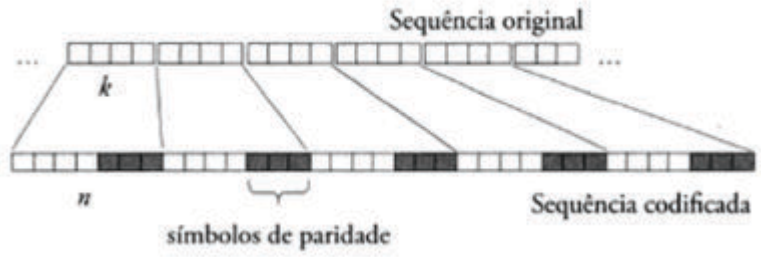


Figura 4.1 – Acrescentando Bits de Paridade ao pacote de informação

Fonte: Silvio Abrantes, pg 39

Nos meios digitais, por exemplo, que reconhecem apenas bits, dependendo do que se quer transmitir, temos um algoritmo para transformação do texto em código, geralmente do tipo $\mathbb{Z}_2$ (aceitando apenas bits 0 e 1), e a adição de bits de verificação ou simplesmente bits para verificação do erro no destinatário.

Vale destacar ainda que existem, na teoria de códigos os algoritmos corretores de erros e os verificadores de erros. Eles podem funcionar em conjunto ou isoladamente, dependendo da função a que se aplica, para garantir a seguridade da informação transmitida.

Ao transmitir uma informação ou dado com confiabilidade, independente do meio, é preciso planejar e escolher qual algoritmo é o mais adequado. Pensemos na dificuldade que seria enviar uma mensagem e ela chegar sem compreensão do outro lado, partida ou exigir que o emissor precise repeti-la até que fosse compreendida. Além de demandar

tempo e energia, dependendo de como isso é feito, ainda se torna inseguro ou inviável, cumprindo fracamente a função ou a ideia de comunicar.

Quando nos referimos a informação por meios digitais, velocidade, segurança e compreensão são fundamentais, o que demanda implementação de estratégias eficientes e que gastem o menor esforço possível.

Ao realizarmos uma transmissão em pacotes de dados, a informação a ser encaminhada nada mais é do que pacotes de dados, de estruturas que aceitam valores (ou bits) quaisquer, mas que ainda mantém sua estrutura, por possuir um tamanho fixo.

Costurando essa ideia com a álgebra abstrata, a mensagem a ser transmitida entre o meio nada mais é do que o bloco de informação cuja escrita se baseia da utilização de bits ou algorismos que pertencem a um conjunto. Independente de quantos e quais elementos possuam tais blocos de informação, o que temos é a certeza de que, utilizando-se da álgebra, são palavras de um "alfabeto", que são obtidos através de vetores ou combinação de elementos de um conjunto finito. (HEFEZ ABRAMO. M.L.T, 2008)

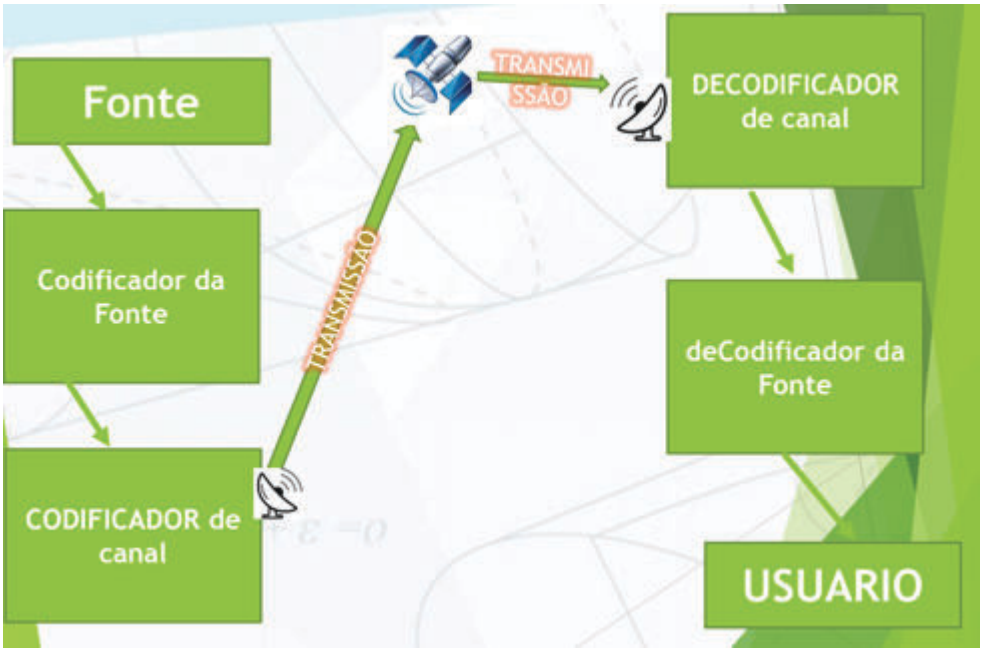


Figura 4.2 – Procedimento para a transmissão de Mensagem

Fonte: autor/banco de imagens-internet

Partindo dessa ideia, pensemos num conjunto não vazio (como o caracterizado anteriormente no capítulo anterior, $\mathbb{Z}_p^n$) na qual podemos definir operações de soma e de multiplicação por escalar. Com o auxílio da álgebra linear sobre corpos finitos, podemos dizer ainda que esses blocos de informação são na verdade sequências de bits organizados

como vetores, ou simplesmente n-uplas de elementos de $\mathbb{Z}_p$, como abordado em (HEFEZ ABRAMO. M.L.T, 2008) .

Ao realizar as diferentes operações definidas com estes elementos, de quantidade finita, cria-se o que definimos como espaço vetorial. Ao admitir, por exemplo, que uma coordenada do $\mathbb{Z}_p^2$, de coordenada genérica $V = (a, b)$, com a e b elementos do conjunto $\mathbb{Z}_p$, estamos construindo um espaço vetorial.

Neste capítulo, definiremos alguns conceitos relacionados a ideia de códigos, assim como representarmos e definirmos o que é um código, sua construção e codificação/decodificação por meio de matrizes. A metodologia segue os conceitos de (HEFEZ ABRAMO. M.L.T, 2008), definindo primeiramente o que é o código linear e a constituição de suas palavras, para posteriormente, apoiados na consideração que estas palavras podem ser codificadas por meio de matrizes, explicar o processo de criação da " novas palavras " e o processo de decodificação e comparação dos pacotes, através das distâncias e métricas.

4.1 Código Linear

Imaginemos uma grande quantidade de informação a ser transmitida de um meio para outro, como por exemplo este trabalho através de um e-mail. A máquina na qual este arquivo está armazenado não considera a estrutura e o valor do nosso " idioma ", entendendo na verdade que cada caracter assume valores 0 e 1 ou simplesmente bits de informação. Ao armazenar este arquivo na forma de um " idioma de máquina", temos o equivalente a um código. Entenda-se que para a transmissão, esse código precisa ser robusto e confiável o suficiente para não se perder na transmissão. Para tal, essa sequência é segmentada em blocos de mensagens, mensagens estas que fazem parte do código.

Cada palavra/mensagem deste código é vista como uma n-upla [1], cuja forma de saída da fonte é da forma binária. Daí, das operações de vetores vistas no capítulo anterior, podemos adicionar os bits para o aumento do comprimento da mensagem e podemos tanto codificá-la quanto decodificá-la.

Por ser uma sequência ordenada de n elementos, de forma binária, e poder ser vista como vetores, é possível mapeá-los dentro de um outro subespaço vetorial de tamanho maior, graças a propriedade de fechamento de subespaços.

Trazendo luz a analogia citada, perceba que as palavras escritas nesta obra são apenas fáceis de entender porque todas as palavras colocadas aqui fazem parte do nosso " idioma ". No momento em que escrevo qualquer coisa diferente das palavras " comuns " (lista de palavras que são compreendidas no nosso idioma-código), perde-se o valor da

[1] Uma n-upla é uma sequência ordenada de **n** elementos. Também é conhecida com **n-upla, u-tuplo** ou simplesmente **tupla**. Por exemplo, $(1, 0)$ é uma 2-upla binaria. Já a sequência $(1, 0)$ é outra 2-upla binária.

informação e portanto, é necessário reescrevê-la.

Para as transmissões digitais, a mensagem enviada faz parte da lista de palavras do código e graças a isso ela pode ser entendida no destinatário (depois de decodificada). O que temos, abordando de forma simples é que a mensagem transmitida guarda consigo a informação e que esta faz parte de uma lista de palavras presentes no código. O fato de ele ser linear permite que não apenas esses bits sejam ordenados, como as operações de codificações realizadas sobre ele modifiquem a mensagem, mas preservem a informação ao descodificar.

Para melhor entender os principais conceitos que fazem parte desse processo, vejamos as definições e proposições seguintes.

Definição 4.1. *Se $C \subset \mathbb{Z}_p^n$. Dizemos que C é um* ***código linear*** *se C é um subespaço vetorial de $\mathbb{Z}_p^n$. Se $p = 2$, C é chamado de código linear binário.*

A definição acima é valida para qualquer corpo finito, como abordado em ((HEFEZ ABRAMO. M.L.T, 2008), pg. 85-86).

Como $C \subset \mathbb{Z}_p^n$ e C é um subespaço vetorial de $\mathbb{Z}_p^n$, então existem $v_1, v_2, \ldots, v_k \in \mathbb{Z}_p^n$ vetores linearmente independentes, tais que

$$C = \langle v_1, v_2, \ldots, v_k \rangle = \{a_1v_1 + a_2v_2 + \cdots + a_kv_k, a_i's \in \mathbb{Z}_p\}$$

Daí, $dim_{\mathbb{Z}_p}C = k$, onde $dim_{\mathbb{Z}_p}C$ denomina-se a dimensão de código C.

Proposição 4.1. *Se $C \subset \mathbb{Z}_p^n$ é um código linear de dimensão k, então $m = |C| = p^k$.*

Demonstração. Para $w \in C$,existem $a_1, a_2, \ldots, a_n \in \mathbb{Z}_p$ tais que $w = a_1v_1 + a_2v_2 + \cdots + a_kv_k$. Segue do princípio multiplicativo que temos p^k possibilidades para cada $a_1, a_2, \ldots, a_k \in \mathbb{Z}_p$ formando vetores $w \in C$. Portanto $|C| = p^K$. □

Observe que: $m = p^k \Leftrightarrow \log_p p^k = k = dim_{\mathbb{Z}_p}C$.

Para exemplificar a notação, escrevemos

Dado $C \subset \mathbb{Z}_p^n$ um código linear cuja base é $\{v_1, v_2, \ldots, v_k\}$ então k é a dimensão do código e n é o comprimento do código.

Para não tornar a notação carregada para a estrutura do vetor, ao citar as "palavras" do código, vamos omitir as vírgulas, representando o vetor $v = (v_1v_2...v_n)$ ao invés de $v = (v_1, v_2, ..., v_n)$.

Exemplo 4.1. *Seja $C = \{(0000), (0110), (1101), (1011)\}$ um código linear binário de comprimento 4 e dimensão 2, uma vez que: $C = [(0110), (1101)]$ e $\{(0110), (1101)\}$ são linearmente independente.*

De fato: $a.(0110) + b.(1101) = (0000)$. Logo

$$\begin{cases} b = 0 \\ a + b = 0 \rightarrow a = b = 0 \\ a = 0 \end{cases}$$

Portanto, só admite a solução trivial.

A partir deles, obtendo as outras palavras do código, já que:

$$(0000) = 0.(0110) + 0.(1101)$$
$$(0110) = 1.(0110) + 0.(1101)$$
$$(1101) = 0.(0110) + 1.(1101)$$
$$(1011) = 1.(0110) + 1.(1101)$$

4.2 Métrica de Hamming

Definição 4.2. *Sejam $u = (x_1x_2 \dots x_n)$ e $v = (y_1y_2 \dots y_n) \in \mathbb{Z}_p^n$. A distância de Hamming entre u e v é o número de coordenadas onde os dois vetores diferem e denotamos por $d(u, v)$, ou seja,*

$$d(u, v) = \mid \{i \in \{1, 2, \dots, n\} ; x_i \neq y_i\} \mid$$

Exemplo 4.2. *Sendo $u = (10111)$ e $v = (00101) \in \mathbb{Z}_2^5$, têm-se que*

$$d(u, v) = \mid \{i \in \{1, 2, 3, 4, 5\} , \, x_i \neq y_i\} \mid = \mid \{1, 4\} \mid = 2.$$

Segue direto da definição 4.2:

Proposição 4.2. *Sejam $u = (x_1x_2 \dots x_n)$ e $v = (y_1y_2 \dots y_n) \in \mathbb{Z}_p^n$. Sobre a distância entre u e v pode-se dizer que:*

1) $d(u, v) \geq 0, \forall u, v \in \mathbb{Z}_p^n$;

2) $d(u, v) = 0$ *se, e somente se,* $u = v$;

3) $d(u, v) = d(v, u), \forall u, v \in \mathbb{Z}_p^n$.

Observe que dados $u = (x_1x_2 \dots x_n)$ e $v = (y_1y_2 \dots y_n) \in \mathbb{Z}_p^n$, temos que

$$d(u,v) = |\{i \in \{1,2,\ldots,n\}; x_1 \neq y_i\}| = |\cup_{i=1}^{n}\{i, x_i \neq y_i\}| = \sum_{i=1}^{n} d(x_i, y_i).$$

Proposição 4.3. *Sejam* $u = (x_1x_2\ldots x_n)$ *e* $v = (y_1y_2\ldots y_n)$ *e* $w = (z_1z_2\ldots z_n)$ *em* $\mathbb{Z}_p^n$. *Então*

$$d(u,v) + d(v,w) \geq d(u,w).$$

Demonstração adaptada se encontra em (HEFEZ ABRAMO. M.L.T, 2008), página 5.

Como $d(u,v)$ satisfaz as propriedades dos itens da Proposição 4.2 e a Proposição 4.3 , dessa forma $d(u,v)$ é uma métrica chamada **Métrica de Hamming**.

Definição 4.3. *Seja* $a \in \mathbb{Z}_p^n$ *e* $r \in \mathbb{Z}$ *com* $r \geq 0$ *definimos o* ***disco*** *de raio* r *e centro* a *o seguinte subconjunto de* $\mathbb{Z}_p^n$*:*

$$D(a,r) = \left\{u \in \mathbb{Z}_p^n; d(u,a) \leq r\right\}.$$

Também, definimos a ***esfera*** *de centro* a *e raio* r *como*

$$S(a,r) = \left\{u \in \mathbb{Z}_p^n; d(u,a) = r\right\}.$$

Proposição 4.4. *Para todo* $c \in \mathbb{Z}_p^n$ *e todo número natural* $r > 0$ *temos que*

$$|\, S(a,r)\,| = \binom{n}{r}(p-1)^r.$$

A demonstração desta proposição é abordada em (MACHADO,), p. 31.

Observação: $S(a,r_1) \cap S(a,r_2) = \varnothing$ se $r_1 \neq r_2$. De fato, se $u \in S(a,r_1) \cap S(a,r_2)$ então $d(u,a) = r_1$ e $d(u,a) = r_2$. Logo, $r_1 = r_2$. Além disso,

$$D(a,r) = \bigcup_{i=0}^{r} S(a,i).$$

Proposição 4.5. *Para todo* $a \in \mathbb{Z}_p^n$ *e todo número natural* $r > 0$*, então:*

$$|\, D(a,r)\,| = \sum_{i=0}^{r} \binom{n}{i}(p-1)^i$$

Demonstração. Como $|\, D(a,r)\,| = |\cup_{i=0}^{r} S(a,i)\,|$ e $S(a,r_1) \cap S(a,r_2) = \varnothing$ se $r_1 \neq r_2$ então

$$|\, D(a,r)\,| = |\bigcup_{i=0}^{r} S(a,i)\,| = \sum_{i=1}^{r} |S(a,i)| = \sum_{i=1}^{r} \binom{n}{i}(p-1)^i.$$

□

Definição 4.4. *O **peso de Hamming** de um vetor $v = (x_1 x_2 \dots x_n) \in \mathbb{Z}_p^n$ é o número de dígitos não nulos do vetor v, o que denotamos $w(v)$, ou seja:*

$$w(v) = \mid \{i \in \{1, 2, \dots, n\} ; x_i \neq 0\} \mid$$

Exemplo 4.3. $u = (10110) \in \mathbb{Z}_2^5 \Rightarrow w(u) = 3$

Exemplo 4.4. $u = (01212110) \in \mathbb{Z}_3^8 \Rightarrow w(u) = 6$

Definição 4.5. *Seja $C \subset \mathbb{Z}_p^n$ um código linear, a **distância mínima** de C é dada por:*

$$d = d_{min} = min \{d(u, v); u, v \in C \, e \, u \neq v\} .$$

Observação: A distância mínima do código C equivalente ao peso mínimo do código.

Exemplo 4.5. *Para o código linear $C = \{(0000), (0110), (1101), (1011)\}$ teríamos as respectivas distâncias entre as palavras*

$$d(0000, 0110) = 2;$$
$$d(0000, 1101) = 3;$$
$$d(0000, 1011) = 3;$$
$$d(0110, 1101) = 3;$$
$$d(0110, 1011) = 3;$$
$$d(1101, 1011) = 2;$$

Logo a distância mínima de C é a menor de todas as distâncias entre palavras do código, ou seja, $d_{min} = 2$.

Exemplo 4.6. *Considere o cubo binário de vertices representando todas as palavras de 3 bits, onde os vertices vizinhos representam as palavras de distância 1. Temos a representação abaixo:*

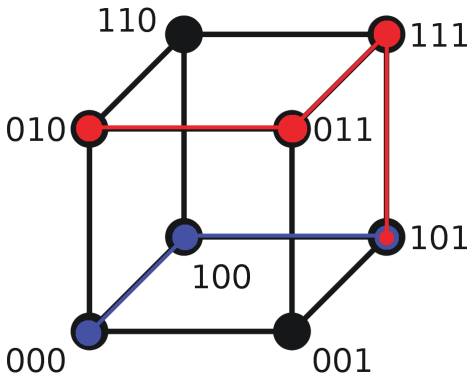


Figura 4.3 – Hipercubo Binário de 3 bits

Fonte: autor/banco de imagens-wikipedia

Geometricamente, as distâncias entre duas palavras são os menores caminhos entre os vertices. Assim, d(000,101)=2 e d(010,101)=3.

Para calcular a distância mínima, como visto acima, foi necessário encontrar as distâncias de todas as palavras, duas a duas, ou simplesmente $\binom{|C|}{2}$. A partir do momento que o comprimento e o número de palavras do código aumenta, esse esforço não se torna mais viável mecanicamente. Daí, é interessante o conhecimento algébrico de técnicas para torná-lo mais eficiente, que demandarão menos energia para estimar a distância mínima.

Proposição 4.6. *Seja $C \subset \mathbb{Z}_p^n$ um código linear. Se $c, c' \in C, c \neq c'$, então*

$$D(c,t) \cap D(c',t) = \varnothing, \text{ onde } t = \left[\tfrac{d-1}{2}\right].$$

Demonstração. Suponha por absurdo que existe $v \in D(c,t) \cap D(c',t)$. Logo, $d(v,c) \leq t$ e $d(v,c') \leq t$. Segue da desigualdade triangular, a Proposição 4.3 que:

$$d(c,c') \leq d(c,v) + d(v,c') \leq t + t = 2t \leq 2.\tfrac{d-1}{2} = d - 1.$$

Logo $d(c,c') \leq d-1 < d$ e isso é um absurdo, pois d é a distância mínima, logo $D(c,t) \cap D(c',t) = \varnothing$ □

Teorema 4.1. *Dado um código $C \subset \mathbb{Z}_p^n$ com distância mínima d, dizemos que esse código detecta até $d-1$ erros e corrige $k = \left[\frac{d-1}{2}\right]$ erros.*

Demonstração. Ver em (HEFEZ ABRAMO. M.L.T, 2008), p.6. □

Definição 4.6. *Um codigo linear $C \subset \mathbb{Z}_p^n$, com distãncia mínima d e capacidade de detecção k diz-se* ***perfeito*** *se :*

$$\bigcup_{c \in C} D(c,k) = \mathbb{Z}_p^n$$

Uma outra forma de se trabalhar com os códigos é utilizando a abordagem com matrizes, que discorremos nas seções a seguir.

4.3 Matriz Geradora do Código

Definição 4.7. *Sejam $\mathbb{Z}_p$, com p primo, um corpo finito de p elementos e $C \subset \mathbb{Z}_p^n$, $n \in \mathbb{Z}, n \geq 2$ um código linear. Considere $\beta = \{v_1, v_2, \ldots, v_k\}$ uma base ordenada de C. Denomina-se matriz geradora do código C a matriz G representada como*

$$G = \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_k \end{pmatrix}$$

Observação: A matriz geradora (G) de um código não é única, pois depende da base de C que estamos considerando.

Efetuando operações elementares de escalonamento sobre as linhas de G, é possível obter uma outra matriz geradora para o código C, em relação a uma outra base ordenada.

Exemplo 4.7. *Sejam $v_1, v_2, v_3, v_4 \in \mathbb{Z}_2^7$ tais que:*

$$v_1 = (1101000), v_2 = (0110100), v_3 = (1110010), v_4 = (1010001)$$

a) Prove que $\{v_1, v_2, v_3, v_4\}$ é linearmente independente (L. I).

b) Descrever as palavras do código $C \subset \mathbb{Z}_2^7$, gerado pelos vetores v_1, v_2, v_3, v_4.

c) Representar a matriz geradora de C.

*Resposta:***a)** Pela definição os vetores v_1, v_2, v_3, v_4 são linearmente independentes se os coeficientes a_1, a_2, a_3, a_4 da combinação linear:

$$a_1v_1 + a_2v_2 + a_3v_3 + a_4v_4 = (0000000)$$

forem $a_1 = a_2 = a_3 = a_4 = 0$.

Realizando a combinação linear para os vetores mencionados acima temos

$$a_1v_1 + a_2v_2 + a_3v_3 + a_4v_4 = (0000000)$$

$$a_1(1101000) + a_2(0110100) + a_3(1110010) + a_4(1010001) = (0000000)$$

E desenvolvendo, analisando cada uma das coordenadas desse vetor, encontramos o sistema:

$$\begin{cases} a_1 + a_3 + a_4 = 0 \\ a_1 + a_2 + a_3 = 0 \\ a_2 + a_3 + a_4 = 0 \\ a_1 = 0 \\ a_2 = 0 \\ a_3 = 0 \\ a_4 = 0 \end{cases}$$

Como se pode perceber naturalmente $a_1 = a_2 = a_3 = a_4 = 0$, temos a solução trivial e portanto $\{v_1, v_2, v_3, v_4\}$ é L . I.

b) As palavras que farão parte do código são resultado da combinação linear de

$$a_1v_1 + a_2v_2 + a_3v_3 + a_4v_4$$

Daí, a_1, a_2, a_3, a_4 assumirá 1 ou 0, e as combinações lineares possiveis são:

	COMBINAÇÃO LINEAR	PALAVRA
1	**0**.(1101000) + **0**.(0110100) + **0**.(1110010) + **0**.(1010001)	(0000000)
2	**1**.(1101000) + **0**.(0110100) + **0**.(1110010) + **0**.(1010001)	(1101000)
3	**0**.(1101000) + **1**.(0110100) + **0**.(1110010) + **0**.(1010001)	(0110100)
4	**0**.(1101000) + **0**.(0110100) + **1**.(1110010) + **0**.(1010001)	(1110010)
5	**0**.(1101000) + **0**.(0110100) + **0**.(1110010) + **1**.(1010001)	(1010001)
6	**1**.(1101000) + **1**.(0110100) + **0**.(1110010) + **0**.(1010001)	(1011100)
7	**1**.(1101000) + **0**.(0110100) + **1**.(1110010) + **0**.(1010001)	(0011010)
8	**1**.(1101000) + **0**.(0110100) + **1**.(1110010) + **1**.(1010001)	(0111001)
9	**0**.(1101000) + **1**.(0110100) + **1**.(1110010) + **0**.(1010001)	(1000110)
10	**0**.(1101000) + **1**.(0110100) + **0**.(1110010) + **1**.(1010001)	(1100101)
11	**1**.(1101000) + **1**.(0110100) + **1**.(1110010) + **0**.(1010001)	(0101110)
12	**1**.(1101000) + **1**.(0110100) + **0**.(1110010) + **1**.(1010001)	(0001101)
13	**0**.(1101000) + **1**.(0110100) + **1**.(1110010) + **1**.(1010001)	(0010111)
14	**1**.(1101000) + **0**.(0110100) + **1**.(1110010) + **1**.(1010001)	(1001011)
15	**0**.(1101000) + **0**.(0110100) + **1**.(1110010) + **1**.(1010001)	(0100011)
16	**1**.(1101000) + **1**.(0110100) + **1**.(1110010) + **1**.(1010001)	(1111111)

c) A matriz geradora do Código é:

$$G = \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_k \end{pmatrix} = \begin{pmatrix} 1101000 \\ 0110100 \\ 1110010 \\ 1010001 \end{pmatrix}$$

Definição 4.8. *Diremos que uma matriz geradora de código $C \subset \mathbb{Z}_p^n$, de dimensão k está na* ***forma padrão*** *se $G = \left[I_{k\times k} \mid A_{k\times(n-k)}\right]$, onde $I_{k\times k}$ é a matriz identidade de ordem k e A é a matriz de ordem $k \times (n-k)$.*

Exemplo 4.8. *Considere o código $C \subset \mathbb{Z}_2^7$ de matriz geradora $G = \begin{pmatrix} 1101000 \\ 0110100 \\ 1110010 \\ 1010001 \end{pmatrix}$*

do exemplo 4.7.

Temos que a dimensão de C é igual ao número de linhas de G, ou seja, 4. Vamos encontrar uma matriz equivalente a G com operações elementares de escalonamento de forma que a matriz esteja na forma padrão. Assim,

$$\begin{pmatrix} 1101000 \\ 0110100 \\ 1110010 \\ 1010001 \end{pmatrix} \xrightarrow[L_4=L_4-L_1]{L_3=L_3-L_1} \begin{pmatrix} 1101000 \\ 0110100 \\ 0011010 \\ 0111001 \end{pmatrix} \xrightarrow{L_4 = L_4 - L_2} \begin{pmatrix} 1101000 \\ 0110100 \\ 0011010 \\ 0001101 \end{pmatrix} \xrightarrow{L_3 = L_3 - L_4} \begin{pmatrix} 1101000 \\ 0110100 \\ 0010111 \\ 0001101 \end{pmatrix}$$

$$\xrightarrow{L_2 = L_2 - L_3} \begin{pmatrix} 1101000 \\ 0100011 \\ 0010111 \\ 0001101 \end{pmatrix} \xrightarrow{L_1 = L_2 - L_1} \begin{pmatrix} 1001011 \\ 0100011 \\ 0010111 \\ 0001101 \end{pmatrix} \xrightarrow{L_1 = L_1 - L_4} \begin{pmatrix} 1000110 \\ 0100011 \\ 0010111 \\ 0001101 \end{pmatrix}$$

Logo $G' = \begin{pmatrix} 1000110 \\ 0100011 \\ 0010111 \\ 0001101 \end{pmatrix}$ é uma matriz geradora do código C e está na forma padrão.(Ver seção 3.7)

4.4 Código Dual e Matriz Teste de Paridade.

Definição 4.9. *Dado $C \subset \mathbb{Z}_p^n$ um código linear, definimos o conjunto*

$$C^{\perp} = \left\{ v \in \mathbb{Z}_p^n; < v, c >= 0, \forall c \in C \right\}$$

Observe que $C^{\perp}$ é um subespaço de $\mathbb{Z}_p^n$. De fato: $C^{\perp} \neq \varnothing$, pois $< (0000000), c >= 0$, para todo $c \in C$. Sejam $u, v \in C^{\perp}$, ou seja,

$$< u, c >= 0, \forall c \in C.$$
$$< v, c >= 0, \forall c \in C.$$

Logo, $< u + v, c >=< u, c > + < v, c >= 0 + 0 = 0$ para todo $c \in C^{\perp}$, assim $u + v \in C^{\perp}$. Também temos $< \alpha.u, c >= \alpha. < u, c >= \alpha.0 = 0, \ \forall c \in C$. Logo, $\alpha.u \in C^{\perp}$. Com essas observações concluímos que $C^{\perp}$ é subespaço de $\mathbb{Z}_p^n$. Denominamos o conjunto $C^{\perp}$ de **código dual** de C.

Teorema 4.2. *Seja $C \subset \mathbb{Z}_p^n$ um código linear de dimensão k e matriz geradora G, então:*

$x \in C^{\perp}$ *se, e somente se,* $G.x^t = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$, *onde* $x \in \mathbb{Z}_p^n$ *e* x^t *denota a transposta da matriz* $x = (x_1 x_2 \ldots, x_n)$.

Demonstração. Seja $G = \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_k \end{pmatrix}$ a matriz geradora de C. Escrevemos $v_i = (v_{i1} v_{i2} \ldots v_{in}), i = 1, 2, \ldots, k$. Então:

$$G = \begin{pmatrix} v_{11} & v_{12} & \cdots & v_{1n} \\ v_{21} & v_{22} & \cdots & v_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ v_{k1} & v_{k2} & \cdots & v_{kn} \end{pmatrix}_{k \times n}$$

Suponha que $x = (x_1 x_2 \ldots x_n) \in C^{\perp}$. Então:

$$< x, v_i >= x_1 v_{i1} + x_2 v_{i2} \ldots x_n v_{in} = 0, \ i = 1, 2, \ldots, k.$$

Observe que:

$$G.x^t = \begin{pmatrix} v_{11} & v_{12} & \cdots & v_{1n} \\ v_{21} & v_{22} & \cdots & v_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ v_{k1} & v_{k2} & \cdots & v_{kn} \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} x_1 v_{11} + x_1 v_{12} + \cdots + x_n v_{1n} \\ x_1 v_{21} + x_2 v_{22} + \cdots + x_n v_{2n} \\ \vdots + \vdots + \ddots + \vdots \\ x_1 v_{k1} + x_2 v_{k2} + \cdots + x_n v_{kn} \end{pmatrix} = \begin{pmatrix} \langle x, v_1 \rangle \\ \langle x, v_2 \rangle \\ \vdots \\ \langle x, v_n \rangle \end{pmatrix}.$$

Portanto, $G.x^t = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$.

Para verificar a recíproca, suponha que $G.x^t = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$. Assim,

$$\begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} = G.x^t = \begin{pmatrix} v_{11} & v_{12} & \cdots & v_{1n} \\ v_{21} & v_{22} & \cdots & v_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ v_{k1} & v_{k2} & \cdots & v_{kn} \end{pmatrix}_{k \times n} \cdot \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} x_1v_{11} + x_1v_{12} + \cdots + x_nv_{1n} \\ x_1v_{21} + x_2v_{22} + \cdots + x_nv_{2n} \\ \vdots \\ x_1v_{k1} + x_2v_{k2} + \cdots + x_nv_{kn} \end{pmatrix} =$$

$$\begin{pmatrix} \langle x, v_1 \rangle \\ \langle x, v_2 \rangle \\ \vdots \\ \langle x, v_n \rangle \end{pmatrix}.$$

Queremos provar que $x \in C^{\perp}$, ou seja $\langle x, c \rangle = 0, \forall c \in C$. Sabemos que

$$\langle x, v_i \rangle = 0,\ i = 1, 2, \ldots, k.$$

Seja $c \in C$. Como $C = [v_1, v_2, \ldots, v_k]$, então $c = a_1v_1 + a_2v_2 + \cdots + a_kv_k$, $a_1, a_2, \ldots, a_k \in \mathbb{Z}_p$. Assim,

$$\langle x, c \rangle = \langle x, a_1v_1 + a_2v_2 + \cdots + a_kv_k \rangle =$$

$$\langle x, a_1v_1 \rangle + \langle x, a_2v_2 \rangle + \cdots + \langle x, a_kv_k \rangle$$

$$a_1\langle x, v_1 \rangle + a_2\langle x, v_2 \rangle + \cdots + a_k\langle x, v_k \rangle = 0 + 0 + \cdots + 0 = 0$$

Logo, $x \in C^{\perp}$. □

Teorema 4.3. *Seja $C \subset \mathbb{Z}_p^n$ um código linear de dimensão k e matriz geradora G, na forma padrão $G = (I_k | A_{k,n-k})$, então:*

a) $dim C^{\perp} = n - k$.

b) Se a matriz geradora de C estiver na forma padrão $G = (I_k \mid A)$, então $H = (-A^t \mid I_{n-k})$ é a matriz geradora de $C^{\perp}$.

A demonstração deste Teorema encontra-se em (HEFEZ ABRAMO. M.L.T, 2008) p. 94 e 95.

Teorema 4.4. *Seja $C \subset \mathbb{Z}_p^n$ um código de dimensão k com matriz geradora G. Uma matriz H de ordem $(n-k) \times n$ com coeficientes em $\mathbb{Z}_p$ e linhas linearmente independente é uma matriz geradora de $C^{\perp}$ se, e somente se, $G.H^t = \begin{bmatrix} 0 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 0 \end{bmatrix}_{k \times n-k}$.*

Demonstração. As linhas de H geram um subespaço vetorial de dimensão igual a $n-k$.

Sejam $h_1, h_2, \ldots, h_{n-k}$ linhas de H, logo $dim[h_1, h_2, \ldots, h_{n-k}] = n-k$, e assim, serão definidas:

$$h_1 = (h_{11}, h_{12}, \ldots, h_{1n})$$
$$h_2 = (h_{21}, h_{22}, \ldots, h_{2n})$$
$$\vdots$$
$$h_{n-k} = \left(h_{(n-k)1}, h_{(n-k)2}, \ldots, h_{(n-k)n}\right)$$

Sejam $g_1, g_2, \ldots, g_k$ linhas de G, logo $dim[g_1, g_2, \ldots, g_k] = k$, e assim, serão definidas:

$$g_1 = (g_{11}, g_{12}, \ldots, g_{1n})$$
$$g_2 = (g_{21}, g_{22}, \ldots, g_{2n})$$
$$\vdots$$
$$g_k = (g_{k1}, g_{k2}, \ldots, g_{kn})$$

Multiplicando G e H^t, encontramos:

$$G.H^t = \begin{bmatrix} g_{11} & g_{12} & \cdots & g_{1n} \\ g_{21} & \vdots & \ddots & \vdots \\ g_{k1} & g_{k2} & \cdots & g_{kn} \end{bmatrix}_{k \times n} \cdot \begin{bmatrix} h_{11} & h_{21} & \cdots & h_{(n-k)1} \\ h_{12} & \vdots & \ddots & \vdots \\ h_{1n} & h_{2n} & \cdots & h_{(n-k)n} \end{bmatrix}_{n \times n-k}$$
$$= \begin{bmatrix} \langle g_1, h_1 \rangle & \langle g_1, h_2 \rangle & \cdots & \langle g_1, h_{n-k} \rangle \\ \vdots & \vdots & \ddots & \vdots \\ \langle g_k, h_1 \rangle & \langle g_k, h_2 \rangle & \cdots & \langle g_k, h_{n-k} \rangle \end{bmatrix}_{k \times n-k} \tag{4.1}$$

Temos nas posições da matriz os produtos internos

$\langle g_i, h_j \rangle; \forall\, i = 1, 2, \ldots, j_k,\ j = 1, 2, \ldots, j_{n-k}$. Como $C = [g_1, g_2, \ldots, g_k]$, segue que para cada $c \in C$ temos $c = a_1.g_1 + a_2.g_2 + \ldots, a_k.g_k$.

Logo

$$\langle g_i, h_j\rangle = 0 \Leftrightarrow < c, h_i >= 0;\ i = 1, 2, \ldots, k,\ j = 1, 2, \ldots, n-k. \tag{4.2}$$

Pois

$\langle c, h_i\rangle = \langle a_1.g_1 + a_2.g_2 + \ldots + a_k.g_k, h_i\rangle =$

$= a_1.\langle g_1, h_i\rangle + a_2.\langle g_2, h_i\rangle + \cdots + a_k.\langle g_k, h_i\rangle = 0.$

Segue das Equações 4.1 e 4.2 que

$\langle g_i, h_j\rangle = 0;\ i = 1, \ldots, k,\ j = 1, \ldots, n-k \Leftrightarrow h_i \in C^{\perp};\ i = 1, \ldots, n-k.$

Ou seja, $h_i \in C^{\perp}$. Como $dimC^{\perp} = n-k$ e $dim\,[h_1, \ldots, h_{n-k}] = n-k$, então

$$C^{\perp} = [h_1, \ldots, h_{n-k}]$$

Portanto $G.H^t = \begin{bmatrix} 0 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 0 \end{bmatrix} \Leftrightarrow C^{\perp}$ é gerado pela linhas de H. □

Proposição 4.7. *Seja $C \subset \mathbb{Z}_p^n$ um código linear, então $\left(C^{\perp}\right)^{\perp} = C$.*

Demonstração. Sejam H e G matrizes geradoras de $C^{\perp}$ e C respectivamente

Suponha $dimC = k$, segue do Teorema 4.4

$H_{n-k,n} = H$ é matriz geradora de $C^{\perp}$ se, e somente se, $G.H^t = \begin{bmatrix} 0 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 0 \end{bmatrix}_{k\times n-k}$.

$$G.H^t = \begin{bmatrix} 0 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 0 \end{bmatrix}_{k\times n-k} \Leftrightarrow (G.H^t)^t = \left(\begin{bmatrix} 0 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 0 \end{bmatrix}_{k\times n-k}\right)^t \Leftrightarrow$$

$$(H^t)^t.G^t = \begin{bmatrix} 0 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 0 \end{bmatrix}_{n-k\times k}$$

$$\Leftrightarrow H.G^t = \begin{bmatrix} 0 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 0 \end{bmatrix}_{n-k\times k} \tag{4.3}$$

Segue da Proposição 4.7 que G é matriz geradora de $\left(C^{\perp}\right)^{\perp} = C$. □

Proposição 4.8. *Seja $C \subset \mathbb{Z}_p^n$ um código linear e seja H a matriz geradora de $C^\perp$. Então:*

$$v \in C \Leftrightarrow H.v^t = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}.$$

Demonstração. Temos do Teorema 4.2 que: $x \in C^\perp \Leftrightarrow G.x^t = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$, daí

$v \in \left(C^\perp\right)^\perp \Leftrightarrow H.v^t = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$. Como $\left(C^\perp\right)^\perp = C$, então $v \in C = \left(C^\perp\right)^\perp \Leftrightarrow$

$H.v^t = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$. □

Definição 4.10. *Seja $C \subset \mathbb{Z}_p^n$ um código linear, a matriz geradora de $C^\perp$ é chamada* ***matriz de teste de paridade.***

A Proposição 4.8 permite caracterizar os elementos de um código C por uma condição de anulamento. Essa condição é obtida por meio da matriz teste de paridade, ou seja,

$$v \in C \Leftrightarrow H.v^t = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}.$$

A matriz teste de paridade de um código linear $C \subset \mathbb{Z}_p^n$ contém informações sobre o peso do código $C, w(C) = d$. Esses resultados são característicos por meio dos Teoremas a seguir.

Teorema 4.5. *Se H é a matriz teste de paridade de um código $C \subset \mathbb{Z}_p^n$, o peso de $C, w(C) \geqslant s$ se, e somente se, quaisquer $s-1$ colunas de H são linearmente independentes.*

A demonstração é abordada em (HEFEZ ABRAMO. M.L.T, 2008), p. 98.

Teorema 4.6. *Seja H a matriz de paridade de um código $C \subset \mathbb{Z}_p^n$. O peso de $C, w(C) = s$ se, e somente se, quaisquer $s-1$ colunas de H são linearmente independentes e existem s colunas de H linearmente dependentes.*

Demonstração. Suponha que $w(C) = s$. Segue do Teorema 4.5 $w(C) = s \geqslant s \Leftrightarrow \forall s-1$ colunas de H são linearmente independentes. Resta provar que existem s colunas de H que são linearmente dependentes. De fato, se quaisquer k colunas de H são linearmente independentes com $k \geqslant s$, então segue do Teorema 4.5 que $k = k+1-1$ colunas de H são linearmente independentes se, e somente se, $w(C) \geqslant k+1$. Entretanto,

$w(C) \geqslant k+1 \geqslant s+1 > s$ e isto é uma contradição, pois $w(C) = s$. Portanto segue o resultado.

Reciprocamente, suponha que quaisquer $s-1$ colunas de H são linearmente independentes e existem s colunas de H que são linearmente dependentes. Queremos provar que $w(C) = s$. Segue do Teorema 4.5 que $w(C) = s$, pois por hipótese quaisquer $s-1$ colunas de H são L . I.

$$w(C) \geqslant s \Leftrightarrow w(C) = s \text{ ou } w(C) > s$$

Se $w(C) > s$ então $w(C) \geqslant s+1$.

Segue do Teorema 4.5 que $w(C) \geqslant s+1$, se e somente se, quaisquer $(s+1)-1 = s$ colunas de H são L . I. o que é um absurdo, pois existem s colunas de H que são L . D, por hipótese. Portanto $w(C) = s$. □

Corolário 4.1. *Considere os parâmetros (n, k, d) de um código $C \subset \mathbb{Z}_p^n$, onde $k = dimC$ e $d = w(C)$ é o peso do código C. Então $d \leqslant n-k-1$.*

Demonstração. Seja $H = [h^1, h^2, \ldots, h^n]$ a matriz teste de paridade de C, onde $[h^1, h^2, \ldots, h^n]$ são as colunas de H.

Dessa forma, **posto** de H é igual a $n-k$. Segue de (HEFEZ, 2016) em sua página 42, que posto coluna de H é igual ao posto linha de H, que no caso é $n-k$, ou seja, o espaço gerado pelas colunas de H tem dimensão igual a $n-k$.

Segue do Teorema 4.6 que $w(C) = d$ se, e somente se, quaisquer $d-1$ colunas de H é lineamente independentes e existem d colunas de H que são linearmente dependentes.

Dessa forma, quaisquer $d-1$ colunas de $\{h^1, h^2, \ldots, h^n\}$ gera um espaço de dimensão $d-1$ que está contido no espaço gerado por $h^1, h^2, \ldots, h^n$. Logo $d-1 \leqslant n-k$, ou seja, $d \leqslant n-k-k+1$ □

Para concluir o processo de codificação que foi desenvolvido nas seções 4.1, 4.2 e 4.3 temos que uma mensagem u é codificada com uma palavra código $x \in \mathbb{Z}_p^n$, que

em geral é um vetor de $\mathbb{Z}_p^n$, cujas as k primeiras coordenadas correspondem a própria mensagem, seguida de $n-k$ componentes, chamados símbolos de verificação e paridade, ou seja, $x = (x_1x_2 \dots x_n)$ com $x_1 = u_1$; $x_2 = u_2$; $\dots x_k = u_k$ onde $(u_1u_2 \dots u_k) \in C$. Os dígitos $x_{k+1}, \dots, x_n$ são escolhidos de forma a satisfazer

$H.\begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix}$, onde H é a matriz de paridade do código C. Por exemplo,

Exemplo 4.9. *Considere o código $C \subset \mathbb{Z}_2^6$, com $dimC = 3$ e matriz de paridade abaixo:*

$$H=\left[\begin{array}{ccc|ccc} 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \end{array}\right]$$

Temos que $H = [-A^t \mid I_{n-k}]$, logo

$$-A^t = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix} \text{ e } I_{6-3} = I_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Uma mensagem $u = (u_1u_2u_3)$ é codificada na palavra $x = (x_1x_2x_3x_4x_5x_6)$ cujos dígitos tem a seguintes características.

$$x_1 = u_1; x_2 = u_2; x_3 = u_3;$$

E os dígitos $x_4x_5x_6$ são tomada de maneira a satisfazer:

$$H.\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix} \Leftrightarrow \left[\begin{array}{ccc|ccc} 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \end{array}\right] . \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix} \Leftrightarrow$$

$$\begin{cases} x_2 + x_3 + x_4 = 0 \\ x_1 + x_3 + x_5 = 0 \\ x_2 + x_6 = 0 \end{cases} .$$

Dessa forma concluímos

Mensagem: $(u_1u_2u_3)$	Palavra código: $(x_1x_2x_3x_4x_5x_6)$
(000)	(000000)
(001)	(001110)
(010)	(010101)
(011)	(011011)
(100)	(100010)
(101)	(101100)
(110)	(110111)
(111)	(111001)

4.5 O Processo de Decodificação dos Códigos Lineares

O processo de decodificação no qual o decodificador tenta encontrar a palavra código mais próxima da mensagem recebida é denominada **decodificação.** Uma possibilidade é calcular a distância de $y \in \mathbb{Z}_p^n$ (mensagem recebida) a cada palavra de código em $C \subset \mathbb{Z}_p^n$ e corrigir pela palavra código mais próxima. Entretanto, essa ideia é inviável para corrigir mensagens com $dimC = k$ para k muito grande, pois requer p^k cálculos de distâncias e em seguida considerar a menor delas.

Dessa forma, iremos introduzir os conceitos de síndrome de um vetor $v \in \mathbb{Z}_p^n$ e de classes laterais de um código $C \subset \mathbb{Z}_p^n$ para descrever um método mais eficaz para decodificar palavras recebidas.

Definição 4.11. *Dados um código $C \subset \mathbb{Z}_p^n$ com matriz teste de paridade H e $v \in \mathbb{Z}_p^n$. O vetor $S(v) = H.v^t$ é chamado de* ***síndrome*** *de v.*

A matriz de verificação de paridade H de um código $C \subset \mathbb{Z}_p^n$ vai assumir um papel principal de detecção e correção de erros. Se a sindrome é 0, o vetor está no código e a mensagem recebida, após decodificada, é a mesma que a transmitida. Caso contrário,temos um vetor erro $\varepsilon \in \mathbb{Z}_p^n$, definido como $\varepsilon = r - c$, onde r é a mensagem recebida e $c \in C$ é a mensagem transmitida. Segue da Proposição 4.8 que $H.c^t = 0 \Leftrightarrow c \in C$. Assim, para todo $c \in C$ temos:

$$H.\varepsilon^t = H.\,(r-c)^t = H.\left(r^t - c^t\right) = H.r^t - H.c^t = H.r^t - 0 = H.r^t \tag{4.4}$$

Agora apresentaremos o conceito de classes laterais relacionando esse conceito com o vetor síndrome para dar seguimento ao estudo da decodificação de códigos lineares.

Definição 4.12. *Sejam $C \subset \mathbb{Z}_p^n$ um código linear e $a \in \mathbb{Z}_p^n$. O conjunto*

$$a + C = \{a + x; x \in C\}$$

é chamado classe lateral de C.

Proposição 4.9. *Seja $C \subset \mathbb{Z}_p^n$ um código linear com matriz de paridade H e $u, v \in \mathbb{Z}_p^n$.*

$$S(u) = S(v) \Leftrightarrow u \in v + C.$$

Demonstração. $S(u) = S(v) \Leftrightarrow H.u^t = H.r^t \Leftrightarrow H.(u^t - v^t) = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix} \Leftrightarrow$

$$H.(u - v)^t = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix} \Leftrightarrow u - v \in C \Leftrightarrow u \in v + C. \qquad \square$$

Proposição 4.10. *Seja $C \subset \mathbb{Z}_p^n$ um código linear com $dimC = k$, então:*

I. $v + C = v' + C \Leftrightarrow v - v' \in C$;

II. $(v + C) \cap (v' + C) \neq \varnothing \Leftrightarrow v + C = v' + C$

Demonstração. Abordada em (MACHADO,), p.39 e (HEFEZ ABRAMO. M.L.T, 2008), p. 107. $\square$

Proposição 4.11. *Seja $C \subset \mathbb{Z}_p^n$um código linear com $dimC = k$. então:*

I. $\mid v + C \mid = p^k$

II. $\mathbb{Z}_p^n = C \cup (a_1 + C) \cup (a_2 + C) + \ldots + \cup (a_t + C)$

Demonstração. Abordada em (MACHADO,), p.39 e (HEFEZ ABRAMO. M.L.T, 2008), p. 107. $\square$

Exemplo 4.10. *Seja $C \subset \mathbb{Z}_2^6$ um código linear dado por:*

$$C = \{(000000), (001110), (010101), (011011)(100011), (101101), (110110), (111000)\}$$

Seguem algumas classes laterais de C :

$(000000) + C = (000000), (001110), (010101), (011011)(100011), (101101), (110110), (111000)$
$(001110) + C = (000000), (001110), (010101), (011011)(100011), (101101), (110110), (111000)$

Percebe-se que se uma palavra não faz parte de $C \subset \mathbb{Z}_2^6$, suas respectivas classes laterais tambem não resultarão em palavras do código. Tomemos como exemplo as palavras $c_1 = (000001)$ e $c_2 = (100000)$ temos abaixo suas respectivas classes laterais:

$$(000001) + C = (000001), (001111), (010100), (011010)(100010), (101100), (110111), (111001)$$
$$(100000) + C = (100000), (101110), (110101), (111011)(000011), (001101), (010110), (011000)$$

Teorema 4.7. *Seja C um código linear em $\mathbb{Z}_p^n$ com capacidade de correção β. Se $r \in \mathbb{Z}_p^n$ e $c \in C$, tais que $d(c, r) \leqslant \beta$ então existe um único vetor $\varepsilon \in \mathbb{Z}_p^n$ com $w(\varepsilon) \leqslant \beta$, cuja síndrome é igual a síndrome de r tal que $c = r - \varepsilon$.*

Demonstração. Seja H a matriz teste de paridade de C tal que $H = [h^1, h^2, \ldots, h^n]$. Tome $\varepsilon = r - c$. Temos que: $w(\varepsilon) = d(\varepsilon, 0) = d(r - c, 0) = d(r, c)$. Logo $w(\varepsilon) = d(c, r) \leqslant \beta$.

Além disso,

$H.\varepsilon^t = H(r - c)^t = H.(r^t - c^t) = H.r^t - H.c^t = H.r^t$ pois $c \in C$, ou seja, ε e r tem a mesma síndrome.

Agora vamos provar a unicidade de ε tal que $w(\varepsilon) \leqslant \beta$ e $H.\varepsilon^t = H.r^t$. Para isso sejam $\varepsilon = (\alpha_1\alpha_2 \ldots \alpha_n)$ e $\varepsilon' = (\alpha_1'\alpha_2' \ldots \alpha_n')$ tais que $w(\varepsilon) \leqslant \beta$, $w(\varepsilon') \leqslant \beta$, $H.\varepsilon^t = H.r^t$ e $H.\varepsilon'^t = H.r^t$. Temos que $H.\varepsilon^t = H.\varepsilon'^t$, então

$$\sum_{i=1}^{n} \alpha_i.h^i = \sum_{i=1}^{n} \alpha_i'.h^i \tag{4.5}$$

Daí:

$$(\alpha_1 - \alpha_1').h^1 + (\alpha_2 - \alpha_2').h^2 + \cdots + (\alpha_n - \alpha_n').h^n = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix}. \tag{4.6}$$

Com a capacidade de correção do código é $\beta = \left[\frac{d-1}{2}\right]$, onde $d =$ distância mínima do código. Temos que

$$\beta = \left[\tfrac{d-1}{2}\right] \leqslant d - 1.$$

Ou seja, $2\beta \leqslant d - 1$. Além disso $w(\varepsilon) \leqslant \beta$ significa que há no máximo β entradas não nulas do vetor ε. De forma análoga, temos no máximo β entradas não nulas do vetor ε'.

Suponha $\varepsilon - \varepsilon' \neq 0$ como $w(\varepsilon - \varepsilon') \leqslant w(\varepsilon) + w(\varepsilon') \leqslant 2\beta$ então $\varepsilon - \varepsilon' = (\alpha_1 - \alpha_1', \ldots, \alpha_m - \alpha_m')$ tem no máximo 2β componentes tal que $\alpha_i - \alpha_i' \neq 0$. Logo

segue da equação 4.6 que há uma relação linear entre $2\beta \leqslant d-1$ colunas de H. Como quaisquer $d-1$ colunas de H são linearmente independentes (veja Teorema 4.5) temos que $\alpha_i = \alpha'_i; i = 1, 2, \ldots, n$. Portanto, $\varepsilon = \varepsilon'$. □

Definição 4.13. *Sejam $C \subset \mathbb{Z}_p^n$ um código linear com $dimC = k$ e $a_1 + C, \ldots, a_t + C, t = q^{n-k}$ as classes laterais de C. Um vetor $v \in a_i + C$, para algum $1 \leqslant i \leqslant t$, é chamado elemento líder da classe $a_i + C$ se o peso de $v, w(v)$ é mínimo nessa classe.*

Proposição 4.12. *Seja $C \subset \mathbb{Z}_p^n$ um código linear com distância mínima d. Se $u \in \mathbb{Z}_p^n$ tal que $w(u) \leqslant \left[\frac{d-1}{2}\right]$, então u é o único elemento líder de sua classe.*

Demonstração. Suponhamos que $u, v \in \mathbb{Z}_p^n$, com $w(u) \leqslant \left[\frac{d-1}{2}\right]$ e $u \in v + C$, ou seja, $u - v \in C$. Temos que

$$w(u-v) \leqslant w(u) + w(v) \leqslant \left[\tfrac{d-1}{2}\right] + \left[\tfrac{d-1}{2}\right] \leqslant \tfrac{d-1}{2} + \tfrac{d-1}{2} = d-1.$$

Como $u - v \in C$ e $w(u-v) \leqslant d-1 < d$, então $u - v = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix}$, ou seja, $u = v$. □

Para encontrar elementos lideres de classes laterais, basta tomarmos elementos de u tais que $w(u) \leqslant \left[\frac{d-1}{2}\right]$. Segue da Proposição 4.12 que cada um desses elementos é líder de uma e somente uma classe.

Suponha que o código C possa corrigir até β erros, assim $\beta = \left[\frac{d-1}{2}\right]$, com d a distância mínima do código C. Segue da Proposição 4.11 que se $u \in \mathbb{Z}_p^n$ tal que o peso de $u, w(u) \leqslant \left[\frac{d-1}{2}\right]$, então u é o único vetor líder de sua classe. Então vamos nos concentrar em listar os valores $u \in \mathbb{Z}_p^n$ tal que $w(u) \leqslant \left[\frac{d-1}{2}\right]$. Em seguida calcule a síndrome cada vetor u, líder da classe.

Exemplo 4.11. *Seja o Código C de dimensão 3 em $\mathbb{Z}_2^6$ cuja matriz geradora é*

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 \end{bmatrix}$$

Vamos decodificar a palavra recebida $r = (110100)$ usando os líderes de classes laterais.

Note que a matriz geradora G já está na forma padrão. O código C gerado pela matriz G tem as seguintes palavras:

$$C = \{(000000), (001110), (010101), (011011)(100011), (101101), (110110), (111000)\}$$

Se definirmos as classes laterais desse código, seus líderes serão as palavras com o dígito 1 aparecendo alternadamente na posição a_i, com peso igual a 1. Assim os líderes de classe lateral serão: $(100000), (010000), (001000), (000100), (000010), (000001)$.

A sua matriz teste de paridade H na forma padrão, necessária para a decodificação da palavra recebida pode ser escrita como:

$$H = [-A^t \mid I_{6-3}] = [-A^t \mid I_3], \text{ onde } A = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix} \text{ e assim } -A = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}.$$

Logo,

$$H = \left[\begin{array}{ccc|ccc} 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 \end{array}\right] = \begin{bmatrix} h^1 & h^2 & h^3 & h^4 & h^5 & h^6 \end{bmatrix}$$

Observe que $h^3 = h^1 + h^2$, ou seja, existem h^1, h^2 e h^3 colunas lineamente dependente. Então segue do Teorema 4.6 que $d = w(C) = 3$.

Assim o código C pelo Teorema 4.1 pode corrigir até $\beta = \left[\frac{3-1}{2}\right] = \left[\frac{2}{2}\right] = 1$. Agora, vamos listar $u \in \mathbb{Z}_2^6$, tais que $w(u) \leqslant 1$.

Mensagem: Líder da classe Lateral	Síndrome
000000	000
100000	011
010000	101
001000	110
000100	100
000010	010
000001	001

Seja $r = 110100$, a síndrome da palavra, $S(r)$, determina-se:

$$S(r) = H.r^t = \left[\begin{array}{ccc|ccc} 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 \end{array}\right] . \begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}.$$

De acordo com a tabela acima $\varepsilon = (000010)$

Logo, $\varepsilon = r - c$, onde c é a palavra enviada, ou seja, $(000010) = (110100) - c$. Isto é $c = (000010) + (110100) = (110110)$.

Agora é possível generalizar o procedimento para corrigir erros de uma palavra recebida. Para isso os dados de entrada são: um código $C \subset \mathbb{Z}_p^n$ que pode corrigir até $\left[\frac{d-1}{2}\right]$ erros, onde d é a distância mínima e $r \in \mathbb{Z}_p^n$ é a palavra recebida.

Procedimento:

Passo 1 - Calcule $u \in \mathbb{Z}_p^n$ tal que $w(u) \leqslant \left[\frac{d-1}{2}\right]$.

Passo 2 - Calcule a síndrome do vetor u obtido no passo 1.

Passo 3 - Faça a tabela, cuja a primeira coluna é o vetor u do passo 1 e a segunda coluna é a síndrome $S(u)$ do passo 2;

Passo 4 - Calcule $H.r^t = S(r)$. Se a síndrome de r estiver na tabela do passo 3, cujo u é o elemento líder então faça $c = r - u$, senão $S(r)$ não está na tabela e mensagem tem mais do que $\left[\frac{d-1}{2}\right]$ erros e, portanto, não pode ser corrigida.

Fim do Procedimento:

O procedimento de decodificação é eficaz pelo fato de que se baseia num produto de matrizes, e no caso de um código linear, o procedimento é ainda mais simples, seguindo passos semelhantes:

Procedimento:

Passo 1 - Recebida a palavra $r \in \mathbb{Z}_p^n$, realize o produto das matrizes $u = H.r^t$;

Passo 2 - Se no passo 1, o vetor u é igual ao vetor nulo, as palavras transmitida e recebida são as mesmas, o que indica que não houve interferências. Caso contrário, assumimos que o código apresentou o erro em um dígito e o vetor coincide com o índice de uma das colunas da matriz teste de paridade, H, onde tal coluna é a posição em que ocorreu o erro;

Passo 3 - Como explicado no passo 2, para obter a palavra transmitida basta modificar a palavra recebida r, trocando o dígito incorreto na posição indicada.

Fim do Procedimento:

Como se pode ver, o processo de codificação/decodificação em $\mathbb{Z}_p^n$ de um código linear é facilitado pelas ideias de paridade, mas principalmente, pela descoberta da matriz de paridade de um código. Para tal, é necessario entender as ideias de álgebra relacionadas a teoria da informação, num algoritmo que, embora intuitivo, tem em suas estruturas muito das álgebras abstrata e linear.

5 O Código Linear de Hamming e a abordagem de códigos no ensino básico.

Admitir que a evolução das tecnologias é algo de suma importância na velocidade das descobertas nos mais diferentes segmentos é, sem dúvida alguma, permitiu um grande salto no século XX. Dividir com as máquinas a tarefa de realizar cálculos complexos, " aprender " com base no erro e sugerir correções foram motivos que encerraram guerras, permitiram o homem conquistar " outros espaços ", e simular e prever eventos.

Levando em conta só o fato da transmissão de dados, temos dinamismo e agilidade. Com os devidos protocolos e compactação de pacotes, temos segurança e redundância. Se pensarmos que o rádio era a única fonte de transmissão de dados há menos de um século, observamos um grande salto de transmissão de dados para os dias atuais.

Entretanto, encaremos os fatos. Nos anos quarenta, o que chamávamos de computador era na verdade grande salas ocupadas por máquinas que faziam estes mesmos cálculos realizando rotinas sequenciais e lineares. Rotinas de programação que precisavam ser exatas e meticulosas, sob o risco de perderem informações ou simplesmente a não execução delas. A " inteligência " do computador se media na sequência de algoritmos e a velocidade com a qual ele os executa. Nem precisamos dizer o quão caro é dispor dessa tecnologia e o quão restrita ela era.

Voltamos ao ano de 1947, especificamente aos laboratórios Bell e o problema do matemático Richard Wesley Hamming. Ao perceber que parte dos algoritmos colocados em cartões perfurados não era executado pelos computadores, ele se pergunta com a capacidade de processamento que esses computadores possuem, não fosse possível detectar e corrigir erros através de um algoritmo.

Nos anos seguintes, suas descobertas e as de C . E . Shannon, em 1948 e suas publicações em 1950 no " The Bell Journal " permitiram o desenvolvimento do campo da teoria da informação e códigos corretores.

A teoria da informação e dos códigos está firmada na álgebra linear e abstrata, tão importante que entende a informação como bloco de conteúdo fundamentada num alfabeto, da qual temos palavras formadas de acordo com o que se quer transmitir. Como estão fundamentadas num código, é possível a construção de matrizes chaves e adicionar bits que garantam a integridade dos dados, como os bits de paridade, e podendo com as técnicas sugeridas por Hamming codificar e decodificar a mensagem mesmo que o meio de transmissão seja ruidoso.

Vale lembrar aqui que as descobertas de Hamming e Golay permitiram que outros

códigos computacionais desevolvessem, tais como a leitura dos dados de hd, mesmo com uns poucos cluster com erros, a criação de algoritmos para validar a autenticidade de um documento, entre outros.

Neste capítulo, rapidamente falamos sobre o código de Hamming e a exemplificação do algoritmo, baseado nos conceitos citados em (HEFEZ ABRAMO. M.L.T, 2008) e exemplificado em trabalhos como o de (MEC, 1998),(DUTRA, 2018),(MALAGUTTI et al., 2010),(OBMEP,) e (GONZAGA,). Em seguida, abordamos um pouco sobre uma alternativa para se ensinar códigos corretores de erros, trazendo um pouco sobre o que se fala sobre o assunto no documento referencial do ensino. Como falamos de códigos computacionais, apresentar ao final um exemplo de aula onde tais conceitos podem ser aplicados.

5.1 O Código de Hamming Binário

Buscando atender o problema inicial da busca na correção do erros, Richard Hamming tentou implementar uma estratégia para a identificação e a correção do erro.

Considerando que o código binário aceita apenas " zeros " e " uns ", e tratando essa sequências como palavras de um código, no qual os bits se repetem um número determinado de vezes, podemos definir o código e entender a estratégia de codificação e decodificação.

Introduzido por Hamming, em 1950, o mais famoso desses códigos, o Hamming $(7, 4, 3)$ é o exemplo pelo qual a teoria da informação teve seu grande salto, e a partir destes outros códigos puderam ser melhor compreendidos. Basicamente, ele considerava os 4 bits de dados de uma palavra original, e acrescentava a ele três bits de verificação (ou paridade). O algoritmo é funcional ao ponto de que o erro de um bit pode ser tanto localizado como corrigido, mesmo com o meio de transmissão apresentando ruído.

Considerando que as palavras de um código são bits de dados que aguardam ser transmitidos, que elas são fundamentadas num alfabeto (ou corpo finito) da forma $\mathbb{Z}_2^n$, podemos construir duas matrizes G e H, respectivamente as matrizes geradora e teste de paridade, e assim codificar ou decodificar cada palavra transmitida.

Definição 5.1. *Um (n, M, d) código é um código que tem M palavras de dimensão n e distância mínima d.*

Definição 5.2. *Um código de Hamming de parâmetros (n, k, d) de ordem m sobre $\mathbb{Z}_2^n$ é um código com matriz teste de paridade H_m de ordem $m \times n$, cujas colunas são os elementos de $\mathbb{Z}_2^m - \{(0, 0, \cdots, 0)\}$ numa ordem qualquer .*

Ou seja, um código de Hamming de ordem m possui comprimento $n = 2^m - 1$ e dimensão $k = 2^m - m - 1$.

Exemplo 5.1. *A matriz H*

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

É uma matriz de um código de Hamming$(3, 2)$ *e parâmetros* $(7, 4, 3)$. *De fato:*

$$m = 3;\ n = 2^3 - 1 = 7;\ k = 2^3 - 3 - 1 = 4;\ n - k = 7 - 4 = 3$$

Teorema 5.1. *Seja* $m \geqslant 2$. *Então*

i) *Ham*$(m, 2)$ *tem parâmetros* $[2^m - 1, 2^m - m - 1, 3]$;

ii) *Ham*$(m, 2)$ *é um código perfeito.*

Demonstração. i) Da Definição 5.2, Ham$(m, 2)$ tem comprimento $| \mathbb{Z}_2^m - \{(0 \cdots 0)\} | = 2^m - 1$ e dimensão $k = 2^m - m - 1$. Provaremos que a distância mínima é $d = 3$. Observando a matriz de paridade de um código, considere as colunas c_i, com $i = 1, \ldots, 2^m - 1$,. Sabemos que nenhuma dessas colunas é nula, e que comparadas entre-si, duas a duas, elas não são iguais. Assim, podemos admitir que duas delas no mínimo são L.I. Pelo Teorema 4.6 uma terceira coluna sendo resultado soma de outras duas, na matriz H , podemos dizer que para cada duas colunas L.I., temos 3 colunas L.D. . Logo $d(Ham(m, 2)) = 3$.

ii) Como $d = 3$, então $\beta = \left[\frac{d-1}{2}\right] = 1$ é a capacidade de correção de C. Logo

$$|D(c, 1)| = \binom{n}{1} = 1 + n \text{ , e assim}$$
$$|\bigcup_{c \in C} D(c, \beta)| = (1 + n).2^k = [1 + 2^m - 1].2^{n-m} = 2^n = |\mathbb{Z}_2^n|$$

□

OBSERVAÇÕES: Para códigos de Hamming binários existe uma técnica para decodificação que é resumida a seguir.

Segue do Teorema 4.1 e da Proposição 4.12 que o código de Hamming (n, k, d) corrige até $\dfrac{3-1}{2} = 1$ erro e $w(C) \leqslant 1$, onde $c \in \mathbb{Z}_2^m$ e o único líder de sua classe.

Considere que as colunas $c_1, c_2, \ldots, c_n$, constituintes da matriz de paridade. Siga os seguintes passos:

1. Recebido $y \in \mathbb{Z}_2^n$, calcular o sindrome $S(y) = H.y^t$.

2. Se $S(y) = 0$, assumir que não ocorreram erros de transmissão e a palavra transmitida é y.

3. Se $S(y) \neq 0$, então $S(y)$ é uma coluna de H e, se estas estão por ordem crescente, assumir que ocorreu um erro na coordenada i correspondente ao número $S(y)$ na base 2, e alterar o bit naquela posição.

Exemplo 5.2. *Considere a matriz de paridade do código hamming* $(7, 4, 3)$

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

Se a palavra recebida $y = 1101011$, ao determinarmos $S(y)$ encontramos o valor $S(y) = 110$. Pelo fato de $S(y) \neq 0$, podemos imediatamente identificar que o erro está no sexto dígito, pois $S(y)$ é igual a sexta coluna na matriz de paridade. Assim, decodificamos como a palavra 1101001.

5.2 Sugestões da abordagem de Códigos de Erros no Ensino Básico

A teoria dos códigos tem aplicabilidade não apenas na informática como no desenvolvimento de sistemas práticos de segurança implementados nos diversos setores.

Sem dúvida, se focarmos nossa atenção apenas no procedimento de codificação/-decodificação de uma mensagem, temos envolvidas as operações aritméticas e o uso das matrizes. Se a vertente a ser trabalhada fosse a dos mecanismos de criptografia e validação, poderíamos citar a maior parte das operações envolvendo a aritmética.

Os documentos que tratam da edução brasileira sugerem o trabalho com os códigos e sua importância na matemática. Segundo os PCN (MEC, 1998) recursos da informática auxiliam na aprendizagem de matemática e de outras áreas de ensino. Assim neste cenário cabe à escola o desafio de incorporar ao seu trabalho novas formas de se comunicar e conhecer. Deve-se oferecer um educação tecnológica voltada a alguns conteúdos dentro de sua estrutura e linguagem de diferentes aplicações da informática, sendo também ressaltada no PCN (MEC, 2000).

A recém aprovada Base Nacional Curricular Comum (BNCC) dá ênfase ao uso no campo dos códigos corretores de erros quando exalta o uso da matemática como contextualizadora de situações práticas, como citado em (DUTRA, 2018), de um trecho reproduzido abaixo:

> A BNCC reconhece os benefícios que a cultura digital tem promovido nas esferas sociais. O avanço tecnológico e a multiplicação de celulares, smartphones e computadores estão diretamente ligados ao hábito de consumo dos jovens. Diante dessas interações multimidiáticas e multimodais, a proposta da Base é trabalhar com uma intervenção social que contextualiza o uso da tecnologia ao currículo aplicado, desenvolvendo essa que é uma das dez competências gerais citadas pelo documento.

Ainda, são elencadas competências que relacionam a matemática e a tecnologia, como na competência " Utilizar processo e ferramentas matemáticas, inclusive tecnologias digitais disponíveis, para modelar e resolver problemas cotidianos, sociais, de outras áreas do conhecimento, validando estratégias e resultados."

5.2.1 PLANO DE AULA PARA O ENSINO FUNDAMENTAL

O plano sugerido abaixo é um exemplo de como o ensino fundamental pode trabalhar alguns desses conceitos na criação de um código.

OBJETIVOS:

- Perceber como funciona o sistema binário de numeração e realizar um paralelo com o sistema decimal.
- Resolver problemas do sistema binário com o auxílio da calculadora, entendendo o processo de conversão binário decimal.
- Compreender, praticando,simulando a estratégia do processo de decodificação de uma mensagem.
- Utilizar do grupo de operações básicas para descobrir algebricamente um caractere oculto.

DURAÇÃO DAS ATIVIDADES 5 a 6 aulas (50 minutos)

RECURSOS DIDÁTICOS Computador, projetor, quadro, folha de atividades, calculadora e papel.

CONTEÚDOS:

- Apresentação do sistema binário de numeração e sua estrutura.
- Sistemas computacionais e paridade.

ORIENTAÇÕES METODOLÓGICAS:

As orientações dispostas aqui são apenas sugestões, uma vez que o professor pode moldar a aula a seu interesse.

Para turmas de alunos cujo conhecimento é ainda superficial, é interessante apresentar um breve histórico, exibindo o vídeo "História dos Números", de (TVESCOLA, 2018). O vídeo apresenta aos alunos a utilização dos números e de alguns sistemas de numeração.

Nosso sistema de base 10, o sistema decimal, é formado por por dígitos de zero a nove. Porém existem outros sistemas numéricos, os quais são usados para os mais variados propósitos:

- A duração de um dia: 0 hora, 1 hora, 2 horas, 3 horas, ..., 23 horas, 1 dia e assim por diante. Ou seja, a base utilizada é 24, pois a cada 24 horas nós temos um dia.
- A contagem das horas, cuja base é 60, pois a cada hora temos 60 minutos e cada minuto 60 segundos...
- Sistemas Octa e Hexadecimal: Alternativas mais compactas ao sistema binário, utilizados por muitas máquinas e transmissões.

É possível ao professor explorar outros sistemas de numeração presentes na história da matemática, como introdução, embora o objetivo seja mostrar a utilidade do sistema computacional binário e suas possibilidades.

Em seguida, apresentar a esquematização do sistema binário de base 2, como explicado na aula de Aritmética da OBMEP, aula 12 – Bases de numeração.

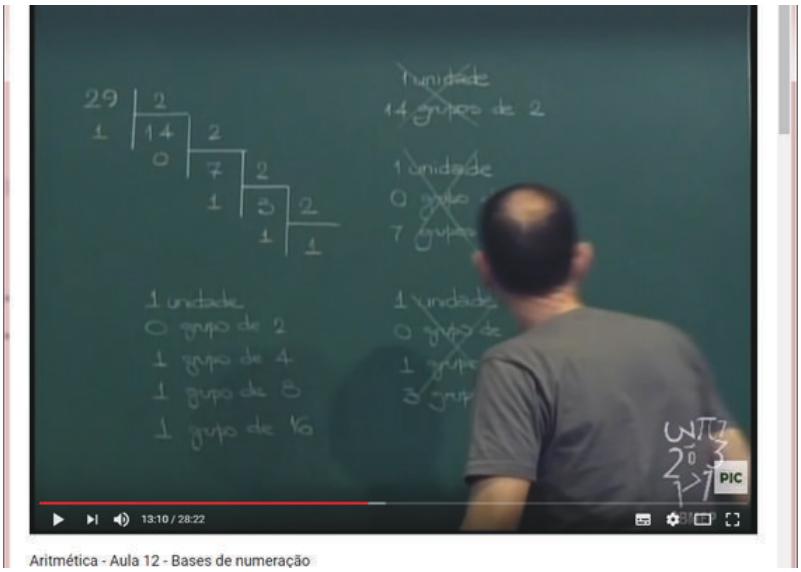


Figura 5.1 – Aula 12(OBMEP)- Sistemas de Numeração

Fonte: youtube: disponivel em https://www.youtube.com/watch?v=4Bu9zJJEIpY

Como atividade, o professor pode sugerir, por exemplo, representar os números 13 e 129 no sistema binário.

Observe a a figura abaixo:

Dividendo	Divisor	Quociente	Resto
13	2	6	1
6	2	3	0
3	2	1	1
1	2	0	1

Dividendo	Divisor	Quociente	Resto
129	2	64	1
64	2	32	0
32	2	16	0
16	2	8	0
8	2	4	0
4	2	2	0
2	2	1	0
1	2	0	1

Figura 5.2 – Conversão Decimal - Binário

Fonte: do autor.

Como se pode observar, seguindo o sentido de leitura das setas, temos que os números 13 e 129 ficam assim representados:

$$13 = (1101)_2, \text{ ou seja, } 13 = 1.2^3 + 1.2^2 + 0.1 + 1.$$

$$129 = (10000001)_2, \text{ ou seja, } 129 = 1.2^7 + 0.2^6 + 0.2^5 + 0.2^4 + 0.2^3 + 0.2^2 + 0.2^1 + 1.$$

O professor deve sugerir outros números para que o aluno faça a conversão de uma base para outra, primeiramente de decimal para binário, explorando a divisão e os restos, como também de binário para decimal, revisitando as ideias relativas a importância do valor posicional dos dígitos.

Para apresentar paridade, vemos basicamente quando um número é par ou não.

Pergunte ao seu aluno e peça que ele responda, argumentando com exemplos questionamentos como "Afinal por que o número é par? e por que ele é impar? O que será que acontece quando somamos, subtraimos e multiplicamos números pares ou impares?"

Passadas as explicações, é possivel mostrar que a ideia de paridade pode ser aplicada em outras situações (SIM/NÃO, LIGADO/DESLIGADO, 0/1).

Na computação, por exemplo prático, entenda-se que toda a informação é representada através dos simbolos 0 ou 1.

No contexto das telecomunicações e envio de mensagens, paridade refere-se ao número de bits '1' de um determinado número binário. Para assinalar a paridade, é adicionado, no final ou no início de uma sequência binária, um **dígito binário de paridade**.

A paridade de um bit (ou mais) é utilizada para detectar erros nas transmissões, já que o seu cálculo é extremamente simples. Por exemplo, se for anexado um bit de paridade extra a cada byte transmitido, um erro pode ser detectado se a paridade do byte não coincidir com o bit de paridade.

Assim, imagine que vamos transmitir por exemplo a informação 1100101. Acrescentar um dígito de paridade é colocar (normalmente no final ou no início da mensagem)o dígito correspondente.

Para compreender isso melhor, imagine que queira colocar um dígito de paridade par(ou seja, o total de 1's dá um resultado par)no final da mensagem, e o nossa mensagem ficará

$$1100101 \rightarrow 1100101\mathbf{0}$$

Pois 1100101 = 1+1+0+0+1+0+1= 4, que é um número par, e daí o digito do final será **0**.

SUGESTÃO DE ATIVIDADE: A TABELA ASCII.

A tabela ASCII do inglês American Standard Code for Information Interchange, é um código alfanumérico que estabelece uma correspondência entre a linguagem binária que é entendida pelo computador e os símbolos que utilizamos para comunicar. Nos anos 60, proposto por Robert W. Bemer, o código ASCII (American Standard Code for Information Interchange) foi adotado como código padrão para a comunicação de informações.Assim, seria possível que computadores de diversos fabricantes conseguissem entender melhor os códigos.

Considere as vogais do nosso alfabeto convertidas na notação binária através da tabela ASCII representada abaixo

Alfabeto ASCII			
A	1000001	N	1001110
B	1000010	O	1001111
C	1000011	P	1010000
D	1000100	Q	1010001
E	1000101	R	1010010
F	1000110	S	1010011
G	1000111	T	1010100
H	1001000	U	1010101
I	1001001	V	1010110
J	1001010	W	1010111
K	1001011	X	1011000
L	1001100	Y	1011001
M	1001101	Z	1011010

Figura 5.3 – Trecho da Tabela ASCII - Código Binário, 1963.

Fonte: computerhistory.org(Editado pelo autor).

Com a tabela ASCII em mãos, converta os caracteres usados em seu nome e escreva da forma binária.

Imagine que este pacote de bits receberá a adição de um bit de paridade par no final, para que tenha uma maior confiabilidade sem perda de informação durante a transmissão. Como seu nome estará formatado?

Se tomássemos como exemplo as letras do nome LUIZA, que na tabela ASCII correspondem respectivamente a $L = 1001100$, $U = 1010101$, $I = 1001001$, $Z = 1011010$, $A = 1000001$, após a adição do bit de paridade par ao final teriamos:

$$10011001\ 10101010\ 10010011\ 10110100\ 10000010$$

AVALIAÇÃO

A aula proposta aqui é uma maneira de apresentar a aplicação prática e o raciocínio algebrico na construção de um código, sua codificação e a checagem de erros bit a bit. Dessa forma, a avaliação que se espera é aquela que seja diagnóstica, processual e contínua, ou seja, realizada ao longo de todas as aulas, se possível em grupos produtivos nas atividades que demandam um tempo maior.

Critérios a serem observados:

- Participação no desenvolvimento das atividades. Respondeu? Produziu?
- O Raciocínio é adequado? Interagiu e argumentou durante as atividades?
- Entendeu corretamente o uso o bit de paridade, produzindo na atividade em que se deveria determinar o último bit?

6 Palavras Finais.

Os códigos, de uma maneira geral hoje são de grande valia nas telecomunicações, assim como na criação de estratégias de segurança para os dados computacionais. Em sala de aula, esses conteúdos normalmente são apresentados como textos base para que professores e alunos sejam apresentados de uma forma rasa aos algoritmos de validação.

Certamente, o trabalho com códigos corretores permite não apenas a exploração do raciocínio matemático como a verificação de algumas propriedades que permeiam a álgebra e aritmética. Dessa forma é possível explorar muito mais opções de conteúdo na hora da apresentação do sistema binário e mostrar que assim como outros sistemas de base ele é fechado para as operações e de significativa importância quando focamos nossa atenção nos sistemas computacionais.

As possibilidades de exploração deste conteúdo são muitas, atacando não apenas habilidades do Ensino Fundamental como habilidade do ensino médio e ainda explorando desde conteúdos simples como paridade até conteúdos mais complexos como o entendimento de maneira algébrica dos algoritmos de validação ou como citado no capítulo 2 o uso das classe residuais.

Uma das reclamações frequentemente ouvidas no ensino básico é que o nosso aluno se torna dependente de ferramentas eletrônicas. A pura e simples aplicação de um algoritmo durante as aulas de matemática e o exercício realizado pelo aluno em papel, mesmo que para um código muito grande, de certa maneira auxilia na apreensão da técnica, e no caso deste algoritmo ser de fácil entendimento, o aluno pode inclusive investigar outras alternativas mais rápidas tornando-se não só investigador, mas também um aprendiz capaz de mobilizar seus saberes.

Ao ingressar no ensino superior, mesmo que ele não tenha sido formalmente apresentado aos conteúdos de álgebra abstrata, um conhecimento na criação de um algoritmo de codificação e de decodificação certamente o tornará preparado para não somente compreender as técnicas utilizadas como um melhor aprofundamento do conteúdo.

Percebe-se portanto que os códigos corretores de erros poderiam ser melhor explorados e a álgebra é uma ferramenta que pode ser trabalhada em algumas séries do ensino básico em diferentes níveis de acordo com o interesse e disposição de profissionais e alunos.

Embora a álgebra da qual os códigos de erros se fundamentam na abstrata, as operações e procedimentos permeiam a aritmética e a construção do algoritmo para a criação da estratégia de embaralhamento da mensagem. Nos bancos do ensino regular, a contribuição que os profissionais da educação podem dar é mostrar aos alunos que a

algebra pode ser divertida e funcional, apresentando, mesmo que de forma básica e leve, as estruturas envolvidas na criação de sistemas de identificação e validação. Tais algoritmos por sí só já são códigos, e a estratégia utilizada para sua definição pode se tornar corretora de erros. Por fim, a relação entre a álgebra e os códigos corretores é interessante, de grande aplicabilidade na computação moderna, e toda contribuição do profissional na escola básica é bem vinda, pois é possivel estabelecer um paralelo com os outros ramos da matemática e mostrar que as propriedades que essas manipulações possuem podem ser aplicáveis nas diversas áreas, garantindo segurança, eficiência e robustez.

Referências

ALENCAR, M. S. D. A. *Televisão digital*. [S.l.]: Érica, 2007. Citado na página 48.

ANTON, H.; RORRES, C. *Álgebra linear com aplicações*. [S.l.]: Bookman Porto Alegre, 2001. v. 8. Citado na página 29.

BOLDRINI, J. L. et al. *Algebra Linear*. Sao paulo: Editora harper & row do brasil ltda, 1978. Citado na página 29.

BRANCO, E. S. *Como funciona o sistema binário?. Plano de Aula*. 2010. Disponível em: <http://portaldoprofessor.mec.gov.br/fichaTecnicaAula.html?aula=22116>. Accessed: 2019-01-02. Nenhuma citação no texto.

CALLIOLI, C. A.; DOMINGUES, H. H.; COSTA, R. C. F. *Álgebra linear e aplicações*. [S.l.]: Atual, 2000. Citado na página 44.

DIAS, C. M. C. Álgebra booleana e lógica digital uma aplicação da lógica matemática. *Revista Acadêmica: ciências agrárias e ambientais*, Curitiba, 1994. Citado na página 13.

DUTRA, K. *Como a BNCC prevê o uso das tecnologias na sua disciplina?* 2018. Disponível em: <http://redes.moderna.com.br/2018/08/07/bncc-tecnologias-disciplina>. Accessed: 2018-10-08. Citado 2 vezes nas páginas 74 e 76.

EDC, Detecção e Código de erros. Disponível em: <https://epxx.co/artigos/edc.html>. Accessed: 2018-08-30. Citado na página 22.

GONZAGA, Y. *Unidade 1 - O sistema Binário*. Disponível em: <https://youtu.be/_Cw3STPum5w?t=113>. Accessed: 2018-11-08. Citado na página 74.

HEFEZ, A. *Elementos de Aritmética*. Rio de janeiro: Instituto de Matematica Pura e Aplicada, IMPA, 2008. Citado na página 16.

HEFEZ, A. C. d. S. F. *Introduçao a Álgebra Linear*. Rio de janeiro: Instituto de Matematica Pura e Aplicada, IMPA, 2016. Citado 2 vezes nas páginas 29 e 65.

HEFEZ ABRAMO. M.L.T, V. *Código Corretores de erros*. Rio de Janeiro: Instituto de Matemática Pura e Aplicada, IMPA, 2008. Citado 9 vezes nas páginas 50, 52, 53, 56, 61, 64, 67, 68 e 74.

LAY, D. C. *Algebra lineal y sus aplicaciones*. [S.l.]: Pearson educación, 2007. Citado na página 42.

LIRA, E. H. C. d. Códigos corretores de erros no ensino médio: Um estudo sobre o código de hamming. Universidade Federal Rural de Pernambuco,Recife, BR-PE, 2018. Nenhuma citação no texto.

MACHADO, D. A. *Uma abordagem de dígitos verificadores e códigos corretores no ensino fundamental*. Tese (Doutorado) — Universidade de São Paulo. Citado 3 vezes nas páginas 54, 67 e 68.

MALAGUTTI, P. L.; BEZERRA, D. d. J.; RODRIGUES, V. C. d. S. *Aprendendo criptologia de forma divertida*. [S.l.]: Paraıba, 2010. Citado na página 74.

MEC, M. d. E. e. C. Pcn - parametros curriculares nacionais do ensino fundamental. *Secretaria de Educação Fundamental. Brasília: MEC/SEF*, 1998. Citado 2 vezes nas páginas 74 e 76.

MEC, M. d. E. e. C. Pcn - parametros curriculares nacionais do ensino médio. *Disponivel em: http://portal.mec.gov.br/seb/arquivos/pdf/CienciasNatureza.pdf.Secretaria de Educação Fundamental. Brasília: MEC/SEF*, p. 43, 2000. Citado na página 76.

MILIES, C. P. Breve introduçao a teoria dos códigos corretores de erros. *Departamento de Matemática, UFMS*, 2009. Citado na página 48.

NICOLETTI, E. R. Aplicações de ágebra linear aos códigos corretos de erros e ao ensino médio. Universidade Estadual Paulista (UNESP),Mestrado Profissional em Matemática, 2015. Nenhuma citação no texto.

OBMEP, P. d. I. C.-P. *Aula 12- Bases de Numeração*. Disponível em: <https://www.youtube.com/watch?v=4Bu9zJJEIpY>. Accessed: 2018-10-10. Citado na página 74.

PINTO, H. Sistemas de identificação com algarismos de controlo1. 2006. Citado na página 27.

PULINO, P. *Álgebra Linear e Suas Aplicações: Notas de Aula. 2012.* 2012. Disponível em: <http://www.ime.unicamp.br/~pulino/ALESA/Texto/>, note = Accessed: 2018-11-10. Citado na página 29.

ROCHOL, J. Comunicação de dados. *Bookman*, 2012. Citado na página 49.

TAUSK, D. V. *Método prático para extrair uma base de um conjunto*. 2008. Disponível em: <https://www.ime.usp.br/~tausk/texts/MetodoBase.pdf>. Accessed: 2018-11-08. Citado na página 43.

TVESCOLA, M. *História dos Números*. 2018. Cópia do Video Disponível em: <http://www.youtube.com/watch?v=Qh6wS2MWXLU>. Accessed: 2018-11-09. Citado na página 78.

VENTURA, J. *Notas de Combinatória e Teoria de Códigos*. Disponível em: <https://www.math.tecnico.ulisboa.pt/~jventura/CTC/NotasCTC.pdf>. Accessed: 2019-1-10. Nenhuma citação no texto.

WINTERLE, P.; STEINBRUCH, A. Álgebra linear. *São Paulo: 2ª ed. McGraw-Hill*, 1987. Citado 2 vezes nas páginas 29 e 31.