UNIVERSIDADE FEDERAL DE SANTA CATARINA
DEPARTAMENTO DE MATEMÁTICA

Nicole Bertoluci Rodrigues

CÓDIGOS CORRETORES DE ERROS

Florianópolis

2017

Nicole Bertoluci Rodrigues

CÓDIGOS CORRETORES DE ERROS

Dissertação submetida ao Programa de Mestrado Profissional de Matemática em rede nacional - PROFMAT para a obtenção do Grau de Mestre em Matemática.
Universidade Federal de Santa Catarina
Orientador: Prof. Dr. Fernando de Lacerda Mortari

Florianópolis

2017

Ficha de identificação da obra elaborada pelo autor,
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Rodrigues, Nicole Bertoluci
Códigos Corretores de Erros / Nicole Bertoluci Rodrigues ; orientador, Fernando de Lacerda Mortari, 2017.
71 p.

Dissertação (mestrado profissional) - Universidade Federal de Santa Catarina, Centro de Ciências Físicas e Matemáticas, Programa de Pós Graduação em Matemática, Florianópolis, 2017.

Inclui referências.

1. Matemática. 2. Códigos Corretores. 3. Código de Hamming. 4. Matriz de Teste de Paridade. I. de Lacerda Mortari, Fernando. II. Universidade Federal de Santa Catarina. Programa de Pós-Graduação em Matemática. III. Título.

Códigos Corretores de Erros

por

Nicole Bertoluci Rodrigues

Esta dissertação foi julgada aprovada para a obtenção do Título de Mestre em Matemática e aprovada em sua forma final pelo Programa de Pós-Graduação em Matemática.

Prof. Dr. Celso Melchiades Dória
Coordenador do Curso

Banca Examinadora

Prof. Dr. Fernando de Lacerda Mortari
Universidade Federal de Santa Catarina -
Orientador

Prof. Dr. Eliezer Batista
Universidade Federal de Santa Catarina

Prof. Dra. Maria Inez Cardoso Gonçalves
Universidade Federal de Santa Catarina

Prof. Dra. Silvia Martini de Holanda
Instituto Federal de Santa Catarina

Florianópolis, 18 de setembro de 2017.

Este trabalho é dedicado a todos aqueles que direta ou indiretamente contribuíram nesta minha jornada, em especial a meu marido Cesar Augusto Cardoso e a minha filha Valentina Rodrigues Cardoso.

AGRADECIMENTOS

Gostaria de agradecer a todos os colegas que me acompanharam durante esta jornada de estudos, em especial a Daine Debortoli e seu marido Ezequiel Debortoli, pelo acolhimento e ajuda para que o sonho da conclusão deste curso fosse possível.

A todos os professores que buscaram nos ajudara e transmitiram conhecimentos para que obtivéssemos o grau de mestre: Daniel Gonçalves, Abdelmoubine Amar Henni, Eliezer Batista, Celso Melchiades Doria, Ruy Coimbra Charão e Gilles Gonçalves de Castro, em especial ao professor Fernando de Lacerda Mortari por ter disposto de seu tempo e me orientado durante esse trabalho.

A minha mãe Divanir Bertoluci que sonhou comigo e me apoiou quando precisei. Ao meu pai Lourival Detânico Rodrigues (in Memoriam) por sempre me mostrou que a educação e o conhecimento são os maiores e melhores bens que podemos ter.

Ao meu marido Cesar Augusto Cardoso, e a minha filha Valentina Rodrigues Cardoso pela paciência, ajuda e compreensão pelas minhas faltas e ausência durante esses dois anos de estudo.

A CAPES, pela ajuda financeira durante o curso.

Tu te tornas eternamente responsável por aquilo que cativas.

(Antoine de Saint-Exupéry, 1943)

RESUMO

Neste trabalho mostraremos a capacidade de códigos lineares detectarem e corrigirem erros através da utilização de matrizes de teste de paridade. Para isso, precisaremos de conceitos de álgebra linear como as operações entre matrizes, em especial a multiplicação.

Palavras-chave: Códigos corretores. Matrizes de teste de paridade. Código de Hamming.

ABSTRACT

In this work we will show the ability of linear codes to detect and correct errors through the use of parity test matrices. For this, we will need concepts of linear algebra such as operations between matrices, especially multiplication.

Keywords: Correcting codes. Parity test matrices. Code of Hamming.

LISTA DE FIGURAS

SUMÁRIO

1 INTRODUÇÃO

A tecnologia está presente em nosso dia-a-dia e junto com ela a matemática, por mais que muitas vezes nem percebamos sua presença.

Hoje em dia, é muito difícil encontrar alguma pessoa que não assista televisão, que não ouça música, que não tenha celular e e-mail, e para que todas essas tecnologias consigam transmitir informações a matemática aparece. Para que consigamos concluir simples tarefas, como ler um e-mail, os códigos corretores de erros (C.C.E.) entram em ação. São eles que permitem que erros nas transmissões de dados sejam detectados e corrigidos.

Mas não é somente na transmissão de informações por meio de equipamentos eletrônicos que identificamos estes códigos. Por exemplo quando nos encontramos em uma ambiente muito ruidoso, com muitas pessoas conversando, o nosso cérebro automaticamente utiliza um código corretor de erro, que nos permite compreender o que uma pessoa esta nos falando, mesmo com as diversas interferências que o ambiente está fazendo.

Outra aplicação de códigos corretores de erros pelo nosso cérebro é no caso de frases escritas com letras trocadas que ainda assim podem ser compreendidas. Um exemplo é a frase "É F4C1L L3R 357A M3N5AG3M S3M P3NS4R MU170", presente em [D'ORNELAS, 2012] que apesar de conter algarismos nos lugares de letras nosso cérebro ainda tem capacidade de decifrar a mensagem.

Com o crescimento das tecnologias da informação foi crescendo também a necessidade de que estas informações chegassem ao seu destino com o mesmo significado que saíram de sua origem, mesmo que na transmissão ocorram ruídos (interferências no processo de transmissão de dados).

Na década de 40, Richard W. Hamming, matemático, Ph.D pela Universidade de Illinois em Urbana-Champaign trabalhou de 1946 a 1976 nos Laboratórios da Bell Telephone, onde tinha acesso a computadores. Estes gravavam seus dados em cartões perfurados e durante sua leitura se um erro fosse detectado ela era imediatamente interrompida.

Não satisfeito com isso, Hamming pensou que se as máquinas conseguiam detectar o erro, por que não conseguir localizá-lo e corrigi-lo? E com essa motivação ele começou seu estudo, que levou em torno de três anos para ser publicado no "The Bell System Technical Journal".

Claude E. Shannon, matemático, conhecido como o pai da Teoria da Informação, que também trabalhava nos Laboratórios da Bell Telephone, era parceiro de pesquisa de Hamming e teve um artigo publicado em 1948 (antes mesmo do artigo de Hamming) no qual propôs um código capaz de detectar e corrigir erros, que foi chamado de (7,4)-Código de Hamming.

Marcel J.E. Golay, matemático, físico e teórico da informação, foi um profissional muito importante para o desenvolvimento da tecnologia da informação. Ele teve papel bastante importante para o Exército Americano, principalmente por desenvolver sistemas de radares, que identificavam emissões infravermelhas dos aviões. No que diz respeito aos códigos corretores de erros, Golay aprofundou o estudo de Hamming e Shannon, conseguindo determinar um código de comprimento primo p. Golay desenvolveu também, os códigos (23,12) e (11,6) durante seus estudos.

A NASA, com suas missões espaciais precisa transmitir uma grande quantidade de informações que estão muito suscetíveis a interferências. As espaçonaves Voyager 1 e 2, começaram a transmitir imagens em cores graças ao código (24,12,8) de Golay, código este muito usado também nas transmissões de rádio de alta frequência.

No segundo capítulo apresentaremos os códigos corretores de erros do ponto de vista do ensino médio, em particular o Código de Hamming, mostrando como o Diagrama de Venn pode ser utilizado para verificar e corrigir mensagens com até um erro na transmissão.

No terceiro capítulo apresentaremos os códigos corretores de erros do ponto de vista da álgebra linear, definindo o que é alfabeto, palavra, comprimento de palavra, métrica de Hamming, distância entre palavras, código corretor. Exemplificamos os códigos do Teste de Paridade, de Repetição Binária e do Código de Hamming, definimos também a distância mínima de um código linear e matrizes geradoras de um código, para então chegarmos a uma matriz de teste de paridade do Código de Hamming.

Segue, então, que o objetivo do nosso trabalho é mostrar a aplicação do código de Hamming do ponto de vista do Ensino Fundamental e também da Álgebra Linear, e apresentar uma *Matriz Teste de Paridade* especial para o Código de Hamming.

2 C.C.E. NO PONTO DE VISTA DO ENSINO FUNDAMENTAL E MÉDIO

Nós seres humanos estamos constantemente nos comunicando, com tudo aquilo que nos cerca. O simples fato de olharmos uma imagem e identificá-la exemplifica um processo de comunicação, já que comunicação é a ação de transmitir uma mensagem e também de receber mensagens, ou seja, pode ser dita como transmissão e recebimento de informações. Por exemplo, a nossa capacidade de falar e ouvir, nos faz constantemente estar transmitindo e recebendo informações.

O nosso cotidiano está cercado de interações, com o meio ambiente, com outras pessoas, com materiais. O recebimento de correspondências, como contas de luz, água e telefone, via correio, também pode ser entendida como uma interação entre o remetente e o destinatário e uma transmissão de informações. Nossa visão, também sugere uma interação, para que consigamos enxergar uma árvore, o nosso cérebro recebe informações que são captadas por nossos olhos e as traduz na imagem que vemos. Quando sentimos um cheiro gostoso, ou não muito bom, também estamos recebendo um tipo de informação que é processada pelo nosso cérebro.

Mas nem todos esses processos de comunicação ocorrem facilmente. Na transmissão de informações podem ocorrer diversos tipos de interferências. No caso da nossa fala, barulhos externos podem interferir na compreensão de uma conversa, assim como problemas auditivos.

Imagine que o seu carteiro pegue um dia muito chuvoso e que suas correspondências molhem no percurso que ele faz até a sua casa. Você receberá seus documentos molhados e provavelmente com dificuldades de decifrar o que esta escrito neles, isso é mais um exemplo de interferência. Assim como, quando olhamos por uma janela que é feita de um vidro opaco, isso acaba interferindo na imagem que vemos e nem sempre conseguimos decifrar o que esta por de trás do vidro. No caso do olfato, se ao mesmo tempo que alguém cozinha uma comida muito perfumada, alguém estiver utilizando um produto químico muito forte, seu olfato não será capaz de identificar a informação que esta recebendo, pois esta sofrendo interferência.

Mas o que podemos fazer para ajudar nesses processos de transmissão de informações que estão sofrendo interferências? Muitas vezes quando estamos conversando com alguma pessoa e não conseguimos compreender o que ela falou, pedimos que ela repita. Nesse processo, nem sempre a pessoa fala exatamente o que havia dito antes, mas com

as informações que coletamos no primeiro momento, mais as informações que coletamos no segundo momento, normalmente conseguimos decifrar o que a pessoa quis nos falar.

Nesse processo de repetição, estamos utilizando redundância, que consiste em acrescentar partes extras à mensagem, que auxiliam no processo de sua compreensão. Nem sempre a repetição nos ajuda a decifrar a mensagem, por exemplo no caso de olhar através do vidro opaco, repetidamente não irá nos ajudar a enxergar o que esta atrás dele, para resolver essa interferência precisamos de outra abordagem.

A redundância, por se tratar de qualquer informação extra acrescida na mensagem original, pode ser na forma de informações sendo enviadas repetidamente, como também pode ser alguma informação diferente da enviada, mas que junto com a mensagem original, auxilia na sua compreensão. Nosso estudo acontecerá em cima dessas informações extras que surgem a partir da mensagem original.

Com o desenvolvimento da tecnologia e surgimento dos computadores, veio a necessidade de armazenar qualquer tipo de informação com mais confiabilidade e simplicidade. Sendo assim, as informações armazenadas e transmitidas por computadores, CD's, são codificadas em sequências binárias. Mas esse processo não está livre de interferências.

Na transmissão de informação, existem diversas maneiras de codificar e decodificar mensagens em binário. Primeiramente é difícil pensarmos em como é possível escrever um texto com dígitos, mas existem artifícios para isso.

A partir de 1960, foi desenvolvida uma tabela chamada ASCII, capaz de converter caracteres em sequências binárias. Esta tabela consiste em um conjunto de 255 sequências binárias de 8 dígitos, em que cada sequência representa um caracter (uma letra ou um sinal de pontuação, por exemplo); assim, podemos utilizar a tabela ASCII para codificar em uma sequência binária, qualquer mensagem escrita usando os caracteres contemplados pela tabela, trocando cada caractere pela sequência binária de 8 dígitos que a ele corresponde. Esta tabela é apenas uma das maneiras de codificarmos uma mensagem.

Vejamos a frase *"Quando abro a porta de uma nova descoberta já encontro Deus lá dentro."* de Albert Einsten. Utilizando a tabela ASCII, podemos codificar o símbolo " de aspas duplas, como 00100010, a letra Q maiúscula como 01010001, e assim a frase codificada fica (lendo-se em linhas, da esquerda para a direita):

00100010 01010001 01110101 01100001 01101110 01100100
01101111 00100000 01100001 01100010 01110010 01101111
00100000 01100001 00100000 01110000 01101111 01110010
01110100 01100001 00100000 01100100 01100101 00100000
01110101 01101101 01100001 00100000 01101110 01101111
01110110 01100001 00100000 01100100 01100101 01110011
01100011 01101111 01100010 01100101 01110010 01110100
01100001 00100000 01101010 11100001 00100000 01100101
01101110 01100011 01101111 01101110 01110100 01110010
01101111 00100000 01000100 01100101 01110101 01110011
00100000 01101100 11100001 00100000 01100100 01100101
01101110 01110100 01110010 01101111 00101110 00100010

A Tabela ASCII também pode ser usada para a codificação de mensagens numéricas. Por exemplo, o número 6347 em base decimal pode ser codificado como

00110110000110011001101000011 0111,

se o tratarmos como um texto escrito considerando 6, 3, 4 e 7 como caracteres.

Na matemática, existe uma maneira de representar qualquer número natural usando uma sequência binária, usando a representação numérica em base 2. Adiante mostrarei como fazer esta conversão.

Nosso sistema usual de numeração, é o sistema posicional em base 10, que esse que se utiliza de dez algarismos (0,1,2,3,4,5,6,7,8,9) para formar os infinitos números naturais que conhecemos e operamos. Mas esta não é a única base em que podemos escrever um número. É possível escrever qualquer número natural em outra base, como por exemplo a base 8, a base 2, entre outras.

Nossas televisões, celulares e computadores utilizam o sistema binário, ou seja, o sistema de base 2, que é compostos de somente dois algarismos, o 0 e o 1, ou seja, os números são compostos apenas por sequências binárias.

Identificar um número na base 10 é um processo automático para nós; já no sistema binário, requer um pouco de atenção. Observe: O

número

$$3\ 377$$

está representado na base 10. A representação de um número na base 10, significa que cada algarismo corresponde a multiplicação do mesmo, por uma potência de base 10, onde o algarismo das unidades é multiplicado por 10^0, o algarismo das dezenas é multiplicado por 10^2, o algarismo da centena é multiplicado por 10^3, e assim sucessivamente, aumentando uma unidade no expoente por vez, isto significa que sua representação decimal fica

$$3 \times 10^3 + 3 \times 10^2 + 7 \times 10^1 + 7 \times 10^0.$$

Mas e se quiséssemos fazer sua representação no sistema binário, como seria?

O processo consiste em escrever o número utilizando potências de base 2, mas para isso, é preciso que façamos as sucessivas divisões por 2.

$$\begin{aligned}
3\ 377 &= 2 \times 1\ 688 + 1 \\
1\ 688 &= 2 \times 844 + 0 \\
884 &= 2 \times 422 + 0 \\
422 &= 2 \times 211 + 0 \\
211 &= 2 \times 105 + 1 \\
105 &= 2 \times 52 + 1 \\
52 &= 2 \times 26 + 0 \\
26 &= 2 \times 13 + 0 \\
13 &= 2 \times 6 + 1 \\
6 &= 2 \times 3 + 0 \\
3 &= 2 \times 1 + 1
\end{aligned}$$

Para escrever o número na base 2, devemos utilizar os dígitos dos restos das divisões, ou seja, em sequência binária, logo o número pode ser escrito como:

$$110\ 100\ 110\ 001,$$

e como na base decimal, isto significa que cada algarismo representa

uma multiplicação por uma potência de base 2, ou seja:

$$1 \times 2^{11} + 1 \times 2^{10} + 0 \times 2^9 + 1 \times 2^8 + 0 \times 2^7 + 0 \times 2^6 + 1 \times 2^5 + 1 \times 2^4 + 0 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0.$$

2.1 DIAGRAMA DE VENN

No envio de uma mensagem na forma de sequências binárias, queremos incluir informações extras, redundantes, que ajudem o receptor a decidir se houve erros na transmissão da mensagem e, caso um erro ocorra, que o auxiliem a corrigi-lo. Mas como decidir qual informação extra enviar? Abaixo irei apresentar uma maneira de fazer isso, visto que existem diversas.

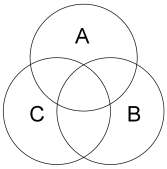
Imagine que a mensagem binária a ser enviada é uma sequência de quatro dígitos:

$$d_1 \ d_2 \ d_3 \ d_4 \ .$$

Cada um desses dígitos pode ser o 0 ou o 1. Por exemplo, uma mensagem poderia ser 1001, outra seria 1110, entre tantas outras possíveis de se formar.

Considere um Diagrama de Venn com três conjuntos, A, B e C, como na Figura 1.

Figura 1 – Diagrama de Venn

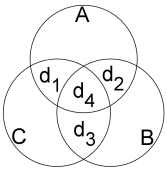


Preencha as intersecções dos conjuntos com os dígitos da mensagem, como apresentado na Figura 2.

Na Figura 2, há três regiões restantes sem dígitos. Usaremos os dígitos já presentes em cada um dos conjuntos para determinar quais dígitos escrever nessas regiões.

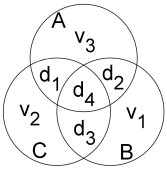
- Se em um dos conjuntos tivermos uma quantidade par de dígitos 1, escreveremos o dígito 0 na região correspondente;
- Se o conjunto tiver uma quantidade ímpar de dígitos 1, escreveremos o dígito 1 na região correspondente.

Figura 2 – Dígitos da Mensagem



Nosso diagrama agora ficará preenchido como na Figura 3.

Figura 3 – Diagrama de Venn



Então, a mensagem final a ser enviada, constituída pelos quatro dígitos da mensagem original, acrescidos dos três dígitos extras encontrados como acima, fica:

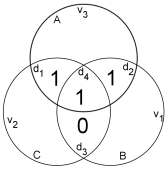
$$d_1 \ d_2 \ d_3 \ d_4 \ v_1 \ v_2 \ v_3.$$

Exemplo 1. *Considere a mensagem*

1101.

Iremos colocar os dígitos no diagrama de Venn, seguindo a ordem de d_1, d_2, d_3, d_4.

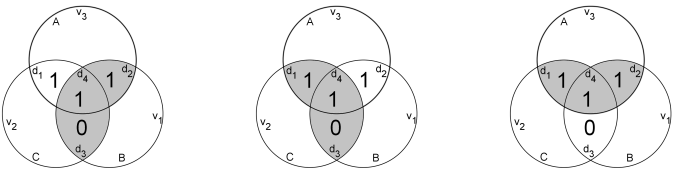
Figura 4 – 1101



Para decidir quais serão os dígitos extras na mensagem final a ser enviada, vamos analisar na Figura 5 cada conjunto, com seus res-

pectivos dígitos já presentes.

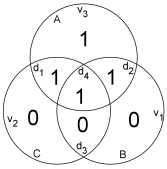
Figura 5 – 1101



Ao analisarmos o conjunto B, notamos que a parte sombreada possui dois dígitos 1, ou seja, uma quantidade par, então o dígito a ser acrescentado é o 0. No conjunto C, notamos que a parte sombreada possui dois dígitos 1, ou seja, uma quantidade par, então o dígito a ser acrescentado é o 0. Já no conjunto A, notamos que a parte sombreada possui três dígitos 1, ou seja, uma quantidade ímpar, então o dígito a ser acrescentado é o 1.

O diagrama da mensagem 1101 a ser enviada, com dígitos extras será como o da Figura 6.

Figura 6 – 1101



Então a mensagem com informações extras a ser enviada será:

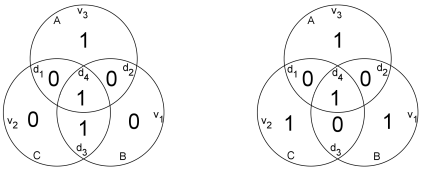
d_1	d_2	d_3	d_4	v_1	v_2	v_3
|	|	|	|	|	|	|
1	1	0	1	0	0	1

Mas o que acontece se a mensagem a ser enviada possuir mais de quatro dígitos? Utilizando o Diagrama de Venn para encontrar as informações adicionais a serem enviadas juntamente com a mensagem, escrevemos quatro dígitos e adquirimos três informações extras, portanto, devemos separar a mensagem a ser enviada em blocos de quatro dígitos, e, para cada bloco de quatro dígitos, encontramos os três dígitos extras como acima.

Observe o número 3 377 que anteriormente escrevemos em base 2 na forma: 110 100 110 001. Perceba que os quatro primeiros dígitos do número, são exatamente o exemplo que acabamos de fazer.

Seguindo a mesma regra aplicada para os quatro primeiros dígitos, os diagramas de Venn correspondente aos próximos oito dígitos (separados em dois blocos de quatro dígitos cada) são os seguintes:

Figura 7 – 0011 0001



Desta maneira, a mensagem

1101 0011 0001

será transmitida em blocos de sete algarismos, onde os quatro primeiros dígitos são a mensagem e os três últimos são os dígitos extras obtidos com o auxílio do diagrama de Venn. Deste modo a mensagem que o destinatário deve receber é:

1101 001 0011 001 0001 111.

2.2 DETECTANDO E CORRIGINDO ERROS

Como já dito anteriormente um processo de transmissão de informações pode sofrer constantemente interferências, e com isso um dígito que foi transmitido como 0 pode chegar a seu destinatário como 1, o que modifica completamente a mensagem.

Hamming na década de 40, já estudava como identificar e corrigir estes problemas de interferências, já que os computadores da época, quando identificavam que houve algum erro simplesmente interrompiam a leitura.

Com base nos estudos de Hamming, em 1948, Shannon criou um código que chamou de (7,4)-Código de Hamming, e que é capaz de identificar e corrigir erros.

No exemplo utilizado anteriormente, acrescentamos informações

extras à mensagem original a ser enviada, com a ideia de identificarmos possíveis erros na transmissão da mensagem. Isso nos mostra uma maneira de utilizar redundância, capaz de nos ajudar a detectar e talvez corrigir possíveis erros causados pelas interferências no processo de transmissão da informação.

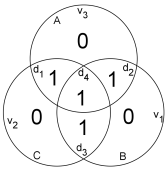
Exemplo 2. *Vamos analisar a mensagem recebida em binário*

1111000.

Vamos assim, exemplificar como o código (7,4)-Hamming possibilita detectar erros nas transmissões de informações. Suponha que na transmissão da mensagem houve erro em apenas um dígito.

Para isso vamos utilizar o diagrama de Venn, da Figura 8 colocando todos os dígitos respeitando a ordem $d_1, d_2, d_3, d_4, v_1, v_2, v_3$.

Figura 8 – 1111 000



Vamos analisar cada um dos conjuntos e seus respectivos elementos na Figura 9. Note que no conjunto B, a parte sombreada possui uma quantidade ímpar de dígitos 1, logo seu dígito extra também deveria ser o 1 ao invés do 0. No conjunto C, notamos que a sua parte sombreada também possui uma quantidade ímpar de dígitos 1, então seu dígito extra deveria ser o 1 e não o zero. Ao analisarmos o conjunto A, a parte sombreada, possui uma quantidade ímpar de dígitos 1, logo, o dígito extra deveria ser o 1 e não o zero.

Com isso, conseguimos perceber que o erro encontra-se na intersecção presente nos três conjuntos, ou seja, no d_4, e portanto a mensagem correta a ser transmitida é

1110 000

e sua representação correta no Diagrama de Venn fica como na Figura 10.

Figura 9 – 1111 000-1

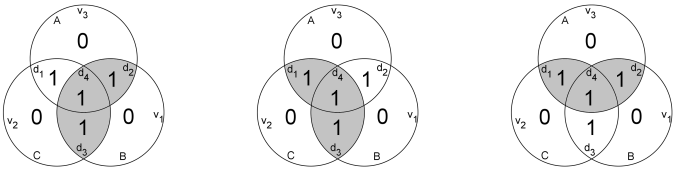
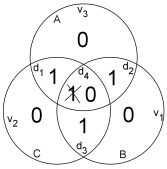


Figura 10 – 1110 000



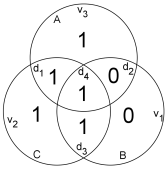
Exemplo 3. *Vamos analisar outro exemplo:*

1011011.

Suponha que na transmissão da mensagem houve erro em apenas um dígito.

Para isso, vamos utilizar o mesmo processo do exemplo anterior, distribuindo os dígitos no diagrama de Venn, na ordem já especificada, como na Figura 11

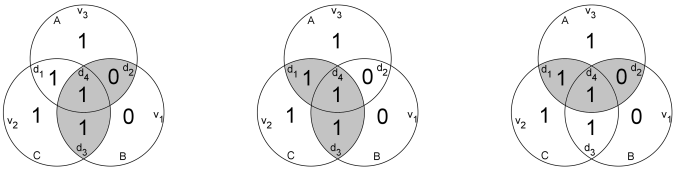
Figura 11 – 1011 011



Seguindo o mesmo processo do exemplo anterior, e analisando as intersecções correspondentes a cada conjunto, temos:

A parte sombreada do conjunto B na Figura 12 possui uma quantidade par de dígitos 1, logo a informação extra é o 0. No conjunto C a parte sombreada, possui uma quantidade ímpar de dígitos 1, logo o dígito extra é o 1. Continuando a nossa análise, olhando agora o conjunto

Figura 12 – 1011 011 - 1



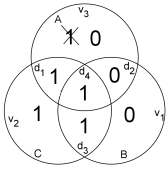
A, vemos que a parte sombreada possui uma quantidade par de dígitos 1, portanto o seu dígito extra deve ser 0, e não o 1. Encontramos um erro nesta transmissão.

Com o auxilio do código (7,4)-Hamming, fomos capazes de perceber um erro no conjunto A e, somente nele, portanto o erro encontra-se na informação extra, pois se estivesse em algum dos dígitos da mensagem original, teríamos detectado erros nos outros dígitos extras também.

Então a mensagem sofreu interferência em sua transmissão, e como esse erro ocorreu na informação extra, somos capazes de corrigila. Portanto a mensagem correta seria

1011 010.

Figura 13 – 1011 010 - 2



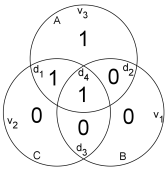
Exemplo 4. *Vamos analisar outro exemplo. Vejamos a mensagem*

1001 001.

Suponha que na transmissão da mensagem houve erros, mas que não sabemos a quantidade.

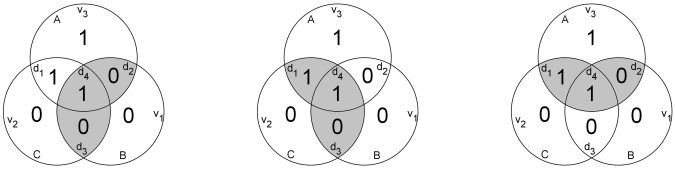
Novamente vamos colocá-la no diagrama de Venn para podermos analisar os possíveis erros na transmissão.

Figura 14 – 1001 001



Vamos observar agora as intersecções com seus respectivos dígitos extras.

Figura 15 – 1001 001 - 1

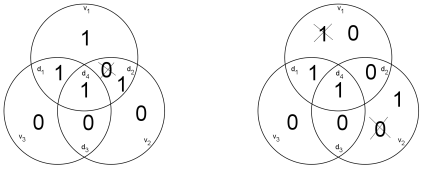


Podemos observar que no conjunto B da Figura 15, a parte sombreada possui uma quantidade ímpar de dígitos 1, logo seu dígito extra deveria ser o 1 e não o 0. No conjunto C, a parte sombreada possui uma quantidade par de dígitos 1, logo seu dígito é o 0. Ao olharmos o conjunto A, notamos que a parte sombreada possui uma quantidade par de dígitos 1, portanto o seu dígito deveria ser o 0 e não o 1.

Analisando o diagrama para fazermos as devidas correções, percebemos que tanto os dígitos extras dos conjuntos A e B podem estar errados quanto o algarismo colocado no espaço de intersecção d_2.

Desta maneira não conseguimos corrigir a mensagem, pois existe mais de uma possibilidade, como podemos ver nos diagramas abaixo:

Figura 16 – 1001 001 - 2



Então existem duas possibilidades para a mensagem que foi en-

viada, e não conseguimos identificar a correta:

$$1101\ 001 \ \textit{ou}\ 1001\ 100.$$

Percebemos com essas análises, que o código (7,4)-Hamming é capaz de identificar até dois erros ocorridos durante a transmissão de dados, mas é capaz de corrigir apenas um erro.

3 CÓDIGOS CORRETORES DE ERROS

Este capítulo trará a formalização da ideia de codificação de mensagens para transmissão e da ideia de detecção e correção de erros de transmissão.

Analisando o que é uma mensagem, percebemos que estas são compostas de palavras, e estas por sua vez, são compostas de letras pertencentes ao alfabeto. Porém, como visto anteriormente, cada letra pode ser representada por uma sequência binária, composta apenas pelos caracteres 0 e 1, e assim as mensagens são transformadas em mensagens binárias.

Definição 3.1. *Por* alfabeto *entendemos um conjunto não vazio A fixado. Os elementos de A serão chamados de* letras *deste alfabeto. Uma* palavra *no alfabeto A consiste de uma sequência de n letras deste alfabeto, para algum $n \in \mathbb{N}$; neste caso, n é chamado de* comprimento *da palavra. O conjunto de todas as palavras no alfabeto A de comprimento n é denotado por A^n.*

Se p é uma palavra de comprimento n no alfabeto A dada pela sequência de letras $a_1, a_2, \ldots, a_n$, denotaremos este fato por

$$p = a_1 a_2 \ldots a_n.$$

Esta notação é a que utilizamos normalmente no caso de palavras no alfabeto usual da língua portuguesa: a palavra dada pela sequência de letras f,a,m,í,l,i,a é escrita simplesmente como *família*, com as letras concatenadas uma ao lado da outra em ordem.

Exemplo 5. *No alfabeto $A = \{0, 1\}$, as letras do alfabeto são* 0 *e* 1*; tem-se que* 0011011 *é uma palavra de comprimento 7, e* 101 *é uma palavra de comprimento 3. O conjunto A^3 de todas as palavras de comprimento 3 neste alfabeto é*

$$A^3 = \{000, 001, 010, 011, 100, 101, 110, 111\}.$$

Definição 3.2. *Para um alfabeto A e $n \in \mathbb{N}^*$, um* código corretor de erros *no alfabeto A consiste de um subconjunto não vazio $C \subseteq A^n$. Neste caso, n é chamado de* comprimento *do código C. No caso particular em que $A = 0, 1$, um código corretor de erros C no alfabeto A é também chamado de um* código binário.

Exemplo 6. *Os conjuntos $C_1 = \{000, 100, 001, 010\}$ e $C_2 = \{111, 011\}$*

são códigos binários de comprimento 3.

Como vimos anteriormente, em um processo de transmissão de mensagens, temos as palavras que queremos transmitir e as palavras que realmente transmitimos, estas acrescidas de redundâncias, ou seja, informações extras capazes de nos permitir perceber se a mensagem que recebemos realmente é a mensagem que nos foi enviada. As palavras originais da mensagem, formam um conjunto de palavras que é um código corretor de erros, chamado de *código* da fonte. Já, as palavras acrescidas de redundâncias, formam um conjunto de palavras que também é um código corretor de erros, chamado de *código do canal*. O processo utilizado para transformar as palavras originais nas palavras acrescidas de redundância é chamado de *codificação de canal*. Portanto, uma codificação de canal é, formalmente, uma *função* do código da fonte para o código de canal.

Exemplo 7. *No capítulo anterior, fizemos exemplos disso, com o código de Hamming. Naqueles exemplos, as palavras originais são palavras de 4 letras no alfabeto* $A = \{0, 1\}$ *que geram um conjunto de palavras que é o código da fonte, e este por sua vez é* A^4*. As palavras que transmitiremos, são as palavras originais acrescidas de 3 informações redundantes, formando palavras de 7 letras no alfabeto* A*, e este processo já foi descrito anteriormente. Então, o código de canal é um subconjunto de* A^7*. O processo que utilizamos para transformar palavras de 4 letras em palavras de 7 letras, que foi descrito anteriormente através de diagramas de Venn, forma a codificação de canal.*

Exemplo 8. *Utilizando novamente o alfabeto* $A = \{0, 1\}$*, vamos considerar* A^6 *como código da fonte. Isso indica que as mensagens originais serão palavras de 6 letras no alfabeto* A*. Para transmitirmos a mensagem, modificaremos cada palavra acrescentando uma letra ao final. A codificação de canal é feita da seguinte maneira:*

- *Se a soma das letras da palavra original for par, acrescentamos 0;*
- *Se a soma das letras da palavra original for ímpar, acrescentamos 1.*

Por exemplo, a palavra 100100 *do código da fonte será modificada para a palavra* 1001000*, pois* $1+0+0+1+0+0 = 2$*, que é par. Note que essa letra extra é uma redundância, que foi obtida através da informação presente nas letras da palavra original. Esta letra acrescida ao final da palavra original* $p \in A^6$ *é chamada de* bit de paridade *da palavra*

p. Realizando esse processo em todas as palavras de A^6*, obteremos um subconjunto* C *de* A^7*, dado por*

$$C = \{p = a_1 \dots a_7 \in A^7 | a_7 \text{ é o bit de paridade de } a_1 \dots a_6\}.$$

Note que C *é o código de canal neste caso, o qual chamamos de* código do teste de paridade. *O processo descrito acima, que leva cada palavra do código da fonte na nova palavra do código de canal, é a codificação de canal, que acrescenta um bit de paridade ao final da palavra.*

Até o presente momento, tratamos da formalização do processo de codificação para o envio de mensagens. Precisamos agora, pensar a respeito do problema em detectar erros de transmissão e corrigi-los se possível.

O processo de detectar erros é simples, apesar de nem sempre ser possível fazê-lo: tanto o emissor quanto o receptor da mensagem conhecem o código de canal, portanto sabem quais palavras *poderiam* em princípio ter sido enviadas. Por exemplo, se o código de canal usado é

$$C = \{00000, 10010, 01100, 10101\}$$

e o receptor recebeu a palavra $p = 10110$, então consultando o código de canal percebe-se que a palavra p não pertence ao código de canal. Isto significa que o emissor não enviou a palavra p, e detectou-se portanto um erro na transmissão.

Para corrigir os erros da transmissão, o problema é mais delicado, pois não sabemos quantos erros ocorreram na transmissão. Por exemplo, a palavra enviada poderia ter sido 00000, e na transmissão ocorreram erros na primeira, terceira e quarta letras, uma vez que a palavra recebida foi $p = 10110$; ou a palavra enviada poderia ter sido 10010, e neste caso teria havido erro na transmissão apenas na terceira letra.

Para que o processo de correção dos erros na transmissão seja possível, precisamos saber algo a respeito da confiabilidade do meio de transmissão das mensagens. É preciso saber alguma informação sobre a média de erros que podemos esperar na transmissão de um certa quantidade de letras pelo meio de transmissão escolhido.

Por exemplo, se optarmos por um meio de transmissão em que a média de erros na transmissão é de uma em cada um milhão de letras transmitidas, é provável que se o receptor detectou um erro na transmissão de uma palavra esse erro tenha ocorrida em apenas *uma letra*. Assim, pode-se supor que houve apenas um erro (ele pode estar errado, mas muito provavelmente está correto), e tentar corrigir a palavra re-

cebida com base nesta informação.

Comparando a palavra recebida $p = 10110$ com as palavras possíveis do código da fonte C, vemos que a única palavra de C que difere de p em uma única letra é a palavra 10010. Deste modo, o receptor conclui que a palavra enviada foi 10010, e o processo de correção está concluído.

Nosso trabalho não tem o intuito de estudar as particularidades dos meios de transmissão nem suas confiabilidades; temos o interesse em saber se o código de canal que apresentamos é capaz de detectar que houve erros na transmissão e, se houverem erros, se ele nos fornece informações suficiente para corrigi-los, supondo que seja de nosso conhecimento que tenha havido no máximo t erros durante a transmissão.

Tratando formalmente o problema desse ponto de vista, somos capazes de criar códigos de canal com capacidade de detecção e correção de erros determinados; deste modo, quando o usuário conhecer a confiabilidade do meio de transmissão a ser usado, ele poderá optar por um código de canal que melhor se adeque às particularidades do meio de transmissão escolhido, para maximizar as possibilidades de correção de eventuais erros de transmissão.

Usaremos agora, uma abordagem geométrica e essa envolve noções de *distância*: se temos palavras p, q, r, com p e q diferindo em uma letra e p e r diferindo em duas letras, podemos imaginar que a palavra q está *mais próxima* da palavra p do que a palavra r, ou que a palavra r está *mais distante* da palavra p do que a palavra q.

A noção de *distância* entre palavras, surge a partir desta comparação entre palavras e as letras em que elas diferem e é conhecida como *distância de Hamming*. Formalmente, a distância de Hamming é exemplo do que chamaremos de *métrica* sobre um conjunto de palavras, e portanto é também conhecida por *métrica de Hamming*; veremos isto com mais detalhes, a partir de agora.

3.1 A MÉTRICA DE HAMMING

Como vimos anteriormente, em matemática, métrica é um conceito que diz respeito a generalização da ideia geométrica de distância.

Definição 3.3. *Dado um conjunto não vazio M, uma métrica sobre M consiste de uma função $d : M \times M \to \mathbb{R}_+$ satisfazendo três axiomas:*

(M1) Para quaisquer $x, y \in M$, $d(x, y) = 0$ se e somente se $x = y$.

(M2) Para quaisquer $x, y \in M$ tem-se que $d(x, y) = d(y, x)$.

(M3) Para quaisquer $x, y, z \in M$ tem-se que $d(x, z) \leq d(x, y) + d(y, z)$.

Dada uma métrica d sobre um conjunto M e elementos $x, y \in M$ quaisquer, dizemos que $d(x, y)$ é a *distância de x até y*, ou simplesmente a *distância entre x e y.*

Definição 3.4. *Dados $n \in \mathbb{N}^*$, A um alfabeto e para palavras $u, v \in A^n$ quaisquer,a* distância de Hamming entre u e v *é definida como o número de letras em que u e v diferem. A função $d : A^n \times A^n \to \mathbb{R}_+$ que leva cada par ordenado (x, y) de palavras de comprimento n no alfabeto A na distância de Hamming entre elas é chamada de* distância de Hamming.

Exemplo 9. *Seja o* alfabeto $A = \{0, 1\}$ *e todas as palavras de 4 letras pertencentes a A, ou seja A^4, temos as seguintes palavras:*

$$0000, 0001, 0010, 0100, 1000, 0011, 0110, 1100,$$
$$0101, 1001, 1010, 0111, 1011, 1101, 1110, 1111.$$

Vamos analisar a distância de algumas palavras, com 1111.
$d(0000, 1111) = 4$, pois as palavras diferem em quatro posições.
$d(0010, 1111) = 3$, pois as palavras diferem em três posições.
$d(1100, 1111) = 2$, pois as palavras diferem em duas posições.
$d(0111, 1111) = 1$, pois as palavras diferem em uma posição.

Definição 3.5. *Dados $n \in \mathbb{N}^*$, A um alfabeto, $x, y \in A^n$ e $i \in 1, \ldots, n$ quaisquer, denotando por x_i e y_i as i-ésimas letras de x e y, respectivamente, definimos o número $d(x, y)_i$ como*

$$d(x, y)_i = \begin{cases} 0 & \text{se } x_i = y_i \\ 1 & \text{se } x_i \neq y_i \end{cases} .$$

Observação 3.1. *Segue das definições que para quaisquer $x, y \in A^n$ tem-se que*

$$d(x, y) = d(x, y)_1 + d(x, y)_2 + \ldots + d(x, y)_n = \sum_{i=1}^{n} d(x, y)_i$$

Iremos agora provar que a *distância de Hamming* é uma *métrica.*

Proposição 3.1.1. *Para quaisquer $n \in \mathbb{N}^*$ e alfabeto A, a* distância de Hamming é uma métrica sobre A^n.

Demonstração. Já sabemos que a *distância de Hamming* é uma função $d : A^n \times A^n \to \mathbb{R}_+$. Logo, basta mostrarmos que d satisfaz os três axiomas da métrica.

Para (M1): Sejam $x, y \in A^n$ quaisquer e suponha que $d(x, y) = 0$, temos que $x_i = y_i \ \ \forall i \in 1, \ldots, n$, logo $x = y$. Por outro lado, se $x = y$ então $x_i = y_i \ \ \forall i \in 1, \ldots, n$, portanto $d(x, y) = 0$.

Para (M2): Sejam $x, y \in A^n$ quaisquer. Sabemos que $d(x, y)$ é, por definição, o número de letras em que x e y diferem. Também, $d(y, x)$ é o número de letras em que y e x diferem. Claramente, são a mesma coisa, isto é, $d(x, y) = d(y, x)$.

Para (M3): Sejam $x, y, z \in A^n$ quaisquer. Seja $i \in 1, \ldots, n$ qualquer. Temos dois casos: ou $x_i = y_i$, ou $x_i \neq y_i$.

- Se $x_i = y_i$, então $d(x, y)_i = 0 \leq d(x, z)_i + d(z, y)_i$.

- Se $x_i \neq y_i$, então $d(x, y)_i = 1$.

Além disso, neste caso z_i não pode ser simultaneamente igual a x_i e a y_i, logo, temos que ou $x_i \neq z_i$ ou $y_i \neq z_i$. Em outras palavras, ou $d(x, z)_i = 1$ ou $d(y, z)_i = 1$, e portanto $d(x, z)_i + d(y, z)_i \geq 1$. Nos dois casos, chegamos à mesma conclusão, portanto tem-se que para todo $i \in 1, \ldots, n$ vale que

$$d(x, y)_i \leq d(x, z)_i + d(z, y)_i.$$

Usando a Observação 3.1,

$$\begin{aligned} d(x, y) = \sum_{i=1}^{n} d(x, y)_i &\leq \sum_{i=1}^{n} (d(x, z)_i + d(z, y)_i) \\ &= \sum_{i=1}^{n} d(x, z)_i + \sum_{i=1}^{n} d(z, y)_i \\ &= d(x, z) + d(z, y) \end{aligned}$$

□

Por esta razão, a distância de Hamming também é chamada de *métrica de Hamming*.

Definição 3.6. *Dados $n \in \mathbb{N}^*$, A um* alfabeto *e C um código corretor de erros de comprimento n sobre o alfabeto A, a* distância mínima *de C é a menor distância de Hamming entre pares de palavras distintas de C, isto é,*

$$\min\{d(u,v) | u, v \in C \text{ com } u \neq v\}.$$

Denotaremos a distância mínima do código C por d.

Exemplo 10. *Sejam C um código corretor de comprimento 4 e $u, v, w \in C$ e $u \neq v, u \neq w$ e $v \neq w$, tais que $u = 0010, v = 0011$ e $w = 1001$. Vamos comparar as palavras de C duas a duas.*

- *(u e v) Note que a primeira entrada de u e v são iguais, então $d(u_1, v_1) = 0$. A segunda entrada de u e v são iguais, então $d(u_2, v_2) = 0$. A terceira entrada de u e v são iguais, então $d(u_3, v_3) = 0$. E na quarta entrada de u e v, notamos que há diferença, então $d(u_4, v_4) = 1$.*
  Segue da Observação 3.1,

$$d(u,v) = 0 + 0 + 0 + 1 = 1.$$

- *(u e w) Note que a primeira entrada de u e w são diferentes, então $d(u_1, w_1) = 1$. A segunda entrada de u e w são iguais, então $d(u_2, w_2) = 0$. A terceira entrada de u e w são diferentes, então $d(u_3, w_3) = 1$. E na quarta entrada de u e v, notamos que há diferença, então $d(u_4, w_4) = 1$.*
  Segue da Observação 3.1,

$$d(u,w) = 1 + 0 + 1 + 1 = 3.$$

- *(v e w) Note que a primeira entrada de v e w são diferentes, então $d(v_1, w_1) = 1$. A segunda entrada de v e w são iguais, então $d(v_2, w_2) = 0$. A terceira entrada de v e w são diferentes, então $d(v_3, w_3) = 1$. E na quarta entrada de v e w, notamos que não há, então $d(v_4, w_4) = 0$.*
  Segue da Observação 3.1,

$$d(v,w) = 1 + 0 + 1 + 0 = 2.$$

Portanto, pela Definição 3.6, a distância mínima do código C é $d = 1$.

Definição 3.7. *Dados $n, r \in \mathbb{N}^*$, A um alfabeto e $a \in A^n$, o disco de centro a e raio r é o subconjunto de A^n denotado por $D(a, r)$ formado*

por todas as palavras de A^n que diferem de a em no máximo r posições, isto é,

$$D(a,r) = \{u \in A^n | d(u,a) \leq r\}.$$

Quando tratamos de discos, a ideia é de que se um código corretor de erros C possuir distância mínima grande, então os discos centrados em palavras deste código C com raios menores que esta distância mínima não terão palavras em comum; o tamanho deste raio depende da distância mínima do código C. Pensando geometricamente, podemos supor que se esses raios não ultrapassarem metade da distância d, conseguimos garantir que os discos centrados nas palavras do código C com estes raios não terão intersecção, ou seja, serão disjuntos. Mas precisamente, temos a seguinte definição.

Definição 3.8. *Dados $n \in \mathbb{N}^*$, A um alfabeto e C um código corretor de erros de comprimento n sobre o alfabeto A com distância mínima d, denotaremos por k o número natural dado pela parte inteira de $\frac{d-1}{2}$.*

Se a parte inteira de um número real t é denotada por $[t]$, então temos que

$$k = \left[\frac{d-1}{2}\right].$$

Proposição 3.1. *Sejam d e k como acima.*

1. *Se d é par, então $d = 2k + 2$;*

2. *Se d é ímpar, então $d = 2k + 1$.*

Demonstração. Se d é par, existe $l \in \mathbb{N}^*$ tal que $d = 2l$. Nesse caso, temos que

$$k = \left[\frac{d-1}{2}\right] = \left[\frac{2l-1}{2}\right] = l - 1 \Rightarrow l = k + 1.$$

Então $d = 2(k+1) = 2k + 2$, como queríamos provar.

Se d é ímpar, existe um $l \in \mathbb{N}^*$ tal que $d = 2l - 1$. Nesse caso, temos que

$$k = \left[\frac{d-1}{2}\right] = \left[\frac{2l-1-1}{1}\right] = \left[\frac{2l-2}{2}\right] = l - 1 \Rightarrow l = k + 1.$$

Então $d = 2(k+1) - 1 = 2k + 2 - 1 = 2k + 1$, como queríamos provar.

□

Nosso próximo resultado, mostrará precisamente que k é o maior raio de modo que todos os discos centrados em palavras do código C com esse raio k são disjuntos aos pares.

Proposição 3.2. *Para quaisquer $n \in \mathbb{N}^*$, A um alfabeto, C um código corretor de erros com distância mínima d e $r \in \mathbb{N}$, são equivalentes:*

1. *$r \leq k$;*

2. *para quaisquer $c, c' \in C$ com $c \neq c'$ temos que*

$$D(c, r) \cap D(c', r) = \emptyset.$$

Demonstração. (1 $\Rightarrow$ 2) Seja $r \in \mathbb{N}$ qualquer e suponha que $r \leq k$. Tome $c, c' \in C$ quaisquer tais que $c \neq c'$. Suponha $x \in D(c, r) \cap D(c', r)$. Logo pelo item (iii) da Definição 3.3 temos que

$$d(c, c') \leq d(c, r) + d(c', r) \leq r + r \leq k + k = 2k \leq d - 1.$$

Observe que $2k \leq d - 1$ pois $k = \left[\frac{d-1}{2}\right]$, logo

$$d(c, c') \leq d - 1$$

o que é absurdo, pois a menor distância entre duas palavras do código C é d.

(2 $\Rightarrow$ 1) Seja $r \in \mathbb{N}$ e suponha $r > k$. A distância mínima de C é d, logo existem $c, c' \in C$ que $d(c, c') = c$ Suponha que $r > k$. Desta maneira encontraremos palavra $a \in A^n$, tal que $d(c, a) = k + 1$ e $d(c', a) = d - (k + 1)$. Visto que $k < r$ por hipótese, temos $k+1 \leq r$ e assim $d(c, a) = k+1 \leq r$, de onde obtemos $a \in D(c, r)$. $a \in D(c', r)$: de fato, temos dois casos, d é par ou d é ímpar.

- Se d é par, temos $d = 2k + 2$ pela Proposição 3.1 , logo

$$d(c', a) = d - (k + 1) = (2k + 2) - (k + 1) = k + 1 \leq r.$$

- Se d é ímpar temos $d = 2k + 1$, logo

$$d(c', a) = d - (k + 1) = (2k + 1) - (k + 1) = k < r.$$

Assim, tem-se que $d(c', a) \leq r$ nos dois casos, e portanto $a \in D(c', r)$, como afirmamos. Portanto, $a \in D(c, r) \cap D(c', r)$, de onde segue que $D(c, r) \cap D(c', r) \neq \emptyset$, e isto conclui a demonstração, por contra positiva.

□

O resultado anterior nos mostra que k dita a capacidade de correção de erros do código C. Por exemplo, se em uma transmissão, soubermos que ocorreram 4 erros e a palavra recebida foi p, então podemos afirmar que a distância entre a palavra original c e a palavra recebida é 4, ou seja, $d(p, c) = 4$

Agora, se c' é outra palavra deste código C e a distância entre c' e p for 4, ou seja, $d(c', p) = 4$, o receptor não saberá decidir qual das duas palavras do código foi enviada, já que as palavras c e c' diferem da palavra recebida em quatro posições.

Pensando geometricamente, temos que, se $d(p, c) = 4$, então $p \in D(c, 4)$ e similarmente se $d(p, c') = 4$, então $p \in D(c', 4)$, portanto, p pertence aos dois discos, o seja $p \in D(c, 3) \cap D(c', 3)$ e com isso $D(c, 3) \cap D(c', 3) \neq \emptyset$. O que nos mostra que se o número de erros for muito grande, então haverá pelo menos dois discos centrados em palavras do código C com raio igual ao número de erros, com alguma palavra p em comum, e neste caso o receptor não saberá como corrigir a palavra p, visto que pelo menos duas palavras diferem em quantidades iguais de letras da palavra p recebida.

Se o número de erros na transmissão de uma palavra ficar abaixo do número k acima, garantimos que os discos centrados nas palavras de C com raios k serão disjuntos, e portanto a palavra recebida pertencerá a apenas um destes discos, logo o erro poderá ser corrigido!

Vamos agora a formalização destas ideias.

Proposição 3.3. *Para quaisquer $n \in \mathbb{N}^*$, A um alfabeto, C um* código corretor de erros *de comprimento n sobre A e $t \in \mathbb{N}$, são equivalentes:*

1. *C pode corrigir todas as transmissões com no máximo t erros;*

2. *os discos de raio t centrados nas palavras de C são disjuntos aos pares.*

Demonstração. (1 $\Rightarrow$ 2) Por contrapositiva: Seja $t \in \mathbb{N}$ qualquer e suponha que existam $c, c' \in C$ tais que $D(c,t) \cap D(c',t) \neq \emptyset$. Então, existe uma palavra $a \in A^n$ nesta intersecção.
Usaremos a para mostrar que existe uma possível transmissão com no máximo t erros que não pode ser corrigida: de fato, suponha que a palavra c foi transmitida, mas a palavra a foi recebida. Então, como $d(c,a) \leq t$, pois $a \in D(c,t)$, temos que nesta transmissão ocorreram no máximo t erros.
Para corrigir os erros, procuramos as palavras de C que distam de a no máximo t unidades. O problema aqui é que há pelo menos duas tais palavras: c e c'.
Assim, o receptor não tem informações suficientes para decidir se a palavra enviada foi c ou c', e deste modo o erro não pode ser corrigido.

(2 $\Rightarrow$ 1) Seja $t \in \mathbb{N}$ qualquer e suponha que os discos de raio t centrados nas palavras de C são disjuntos aos pares.
Afirmamos que C pode corrigir todos os erros de até t símbolos: com efeito, suponha que ao transmitirmos uma palavra $c \in C$, a palavra recebida foi $a \in A^n$, e sabemos que ocorreram no máximo t erros na transmissão. Então, $a \in D(c,t)$, pois $d(c,a) \leq t$.
Por hipótese, os discos de raio t centrados nas palavras de C são disjuntos aos pares; logo, a palavra não está em nenhum outro disco de raio t centrado em alguma palavra de a. Isto garante ao receptor que a palavra transmitida foi de fato a palavra C, visto que todas as outras palavras $c' \in C$ são tais que $d(c',a) > t$ e sabemos que houve no máximo t erros na transmissão.

Deste modo, a transmissão pode ser corrigida, e o resultado está provado. □

Agora, com o próximo resultado, mostraremos que k no mostra precisamente a capacidade de correção de erros de um código C.

Teorema 3.1. *Para quaisquer $n \in \mathbb{N}^*$, A um alfabeto, C um código corretor de erros de comprimento n sobre A e $d \in \mathbb{N}^*$, se C tem distância mínima d, então $k = \left[\frac{d-1}{2}\right]$ é o maior t tal que C pode corrigir todas as transmissões com no máximo t erros.*

Demonstração. Suponha que C tem distância mínima d. Pela Proposição 3.2, os discos de raio k centrados nas palavras de C são disjuntos aos pares. Pela Proposição 3.3, C pode corrigir todas as transmissões

com no máximo k erros. Provaremos que k é o **maior** número natural positivo tal que C pode corrigir todas as transmissões com no máximo k erros. Seja $t \in \mathbb{N}$ qualquer e suponha que $t > k$. Pela Proposição 3.2, existem palavras $c, c' \in C$ tais que $D(c,t) \cap D(c',t) \neq \emptyset$. Pela Proposição 3.3, isto garante que C não consegue corrigir todas as transmissões com no máximo t erros. Assim, k é de fato o **maior** número natural positivo tal que C pode corrigir todas as transmissões com no máximo k erros. □

3.2 CÓDIGOS CORRETORES

Ao longo desta seção e das seguintes, denotaremos $\mathbb{F}$ como um corpo finito com q elementos, que será o alfabeto de todos os códigos apresentados, a menos que dito o contrário.

Dado um $n \in \mathbb{N}^*$, considere o espaço vetorial $M_{n\times 1}(\mathbb{F})$ sobre o corpo $\mathbb{F}$, que por simplicidade denotaremos por V_n. Note que cada vetor v de V_n é uma matriz coluna com n entradas, todas estas, preenchidas com elementos de $\mathbb{F}$, unicamente determinados por v. É possível criar, portanto, uma identificação entre vetores de V_n e palavras de comprimento n no alfabeto $\mathbb{F}$: identificamos cada vetor $v \in V_n$, digamos

$$v = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix},$$

com a palavra $a_1 a_2 \dots a_n \in \mathbb{F}$; por esta razão, também chamamos os vetores de V_n de *palavras de comprimento n no alfabeto* $\mathbb{F}$.

Para cada vetor $v \in V_n$, denotaremos a i-ésima entrada de v por v_i, de modo que

$$v = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix}.$$

Um corpo particularmente importante para o nosso estudo é $\mathbb{Z}_2$, que consiste no conjunto $\{0,1\}$ com adição e multiplicação definidas como:

- Adição:

$+$	0	1
0	0	1
1	1	0

- Multiplicação:

$\cdot$	0	1
0	0	0
1	0	1

Este será o corpo usado em todos os códigos lineares binários.

Definição 3.2.1. *Dado $n \in \mathbb{N}^*$, um* código linear de comprimento n *consiste de um subespaço vetorial W de V_n. No caso particular em que $\mathbb{F} = \mathbb{Z}_2$, dizemos que W é um* código linear binário de comprimento n.

Exemplo 11. *(Código do Teste de Paridade)*

Sejam $\mathbb{F} = \mathbb{Z}_2$, $n \in \mathbb{N}^*$ e considere o seguinte subconjunto W_n de V_n:

$$W_n = \{v \in V_n | v_n = v_1 + v_2 + \cdots + v_{n-1}\}.$$

Afirmação: W_n é um subespaço de V_n.

- $\overrightarrow{0}$ é tal que $\overrightarrow{0}_i = 0 \; \forall \; i \in \{1, \ldots, n\}$, logo

$$\overrightarrow{0}_1 + \overrightarrow{0}_2 + \ldots \overrightarrow{0}_{n-1} = 0 + \cdots + 0 = 0 = \overrightarrow{0}_n,$$

logo $\overrightarrow{0} \in W$.

- Sejam $u, v \in W_n$ quaisquer. Então

$$v_n = v_1 + \ldots v_{n-1} \text{ e } u_n = u_1 + \cdots + u_{n-1}.$$

Assim,

$$v_n + u_n = (v_1 + u_1) + \cdots + (v_{n-1} + u_{n-1}).$$

Como $(v + u)_i = v_i + u_i \; \forall \; i \in \{1, \ldots n\}$, temos que $v + u \in W_n$.

- Sejam $v \in W_n$ e $\alpha \in \mathbb{Z}_2$ qualquer. Então

$$v_n = v_1 + \dots v_{n-1}$$

e

$$\alpha \cdot v_n = \alpha(v_1 + \dots v_{n-1}) = \alpha \cdot v_1 + \cdots + \alpha \cdot v_{n-1},$$

portanto $\alpha v \in W_n$.

Pelo exposto, W_n é subespaço vetorial de V_n.

Afirmação: este código é capaz de detectar no máximo um erro na transmissão de uma palavra qualquer.

Demonstração. Seja $v \in W_{n+1}$ qualquer e suponha que ao transmiti-la ocorreu um erro. Se a palavra recebida foi v' então $d(v, v') = 1$.

- Se o erro foi em v_{n+1}, então

$$v'_1 + \cdots + v'_n = v_1 + \cdots + v_n = v_{n+1} \neq v'_{n+1},$$

logo $v' \notin W_{n+1}$ e conclui-se que houve erro.

- Se o erro foi em v_i para algum $i \in \{1, \dots, n\}$, temos que $v'_i \neq v_i$ e $v'_j = v_j \ \forall \ j \neq i$. Assim,

$$\begin{aligned} v'_{n+1} = v_{n+1} = v_1 + \dots v_n = v'_1 + \cdots + v'_{i-1} + v_i + v'_{i+1} + v'_n \\ = v'_1 + \cdots + v'_{i-1} + v_i + v'_{i+1} + v'_n \neq v'_1 + \cdots + v'_n \end{aligned}$$

e portanto $v' \notin W_{n+1}$, logo houve erro.

Porém, não podemos corrigir o erro, por exemplo: seja

$$v = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \in W_{3+1}.$$

Suponha que a palavra v' recebida seja

$$v' = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 1 \end{bmatrix} \in W_{3+1},$$

percebemos que houve erro, pois o bit de paridade é o dígito 1 e a soma dos algarismos da palavra é $1+0+1=2$ que é par, porém não conseguimos determinar em qual das três entradas de v' ocorreu o erro. □

Exemplo 12. *(Código de Repetição Binária)*

Este código consiste em utilizarmos uma repetição das letras da palavra em uma ordem n.

Seja $\mathbb{F}=\mathbb{Z}_2$, $n,k\in\mathbb{N}^*$, *considere o subconjunto* $R\subseteq V_{nk}$, *onde* $V_{nk}=M_{nk\times 1}(\mathbb{R})$, *dado por*

$$R=\{v\in V_{nk}|\ \forall\ i\in\{0,\ldots,n-1\}\text{ temos } v_{ik+1}=v_{ik+2}=\cdots=v_{ik+k}\}.$$

Afirmação: R é subespaço de V_{nk}.

- $\overrightarrow{0}$ é tal que $\overrightarrow{0}_{ik}=0\ \forall\ i\in\{1,\ldots,n\}$, logo

$$\overrightarrow{0}_{0k+1}=\overrightarrow{0}_{1k+2}=\cdots=\overrightarrow{0}_{(n-1)k+k}=\overrightarrow{0}_{nk},$$

logo $\overrightarrow{0}\in V_{nk}$.

- Sejam $u,v\in V_{nk}$ quaisquer. Então $\forall i\in\{0,\ldots,n-1\}$,

$$v_{ik+1}=v_{ik+2}=\cdots=v_{ik+k}\text{ e }u_{ik+1}=u_{ik+2}=\ldots v_{ik+k}.$$

Assim,

$$v_{ik}+u_{ik}=v_{ik+1}+u_{ik+1}=\cdots=v_{ik+k}+u_{ik+k}.$$

Como $(v+u)_{ik}=v_{ik}+u_{ik}\forall i$, temos que $v+u\in V_{nk}$.

- Sejam $v\in V_{nk}$ e $\alpha\in\mathbb{Z}_2$ qualquer. Então $\forall i\in\{0,\ldots,n-1\}$,

$$\alpha v_{ik+1}=\ldots=\alpha v_{ik+k}.$$

Portanto, $\alpha v\in V_{nk}$.

Para codificação: V_n é o código de fonte e R é o código de canal. Portanto temos que, para $v\in V_n$, $v=\begin{bmatrix}v_1\\ \vdots\\ v_n\end{bmatrix}$, monta-se a palavra

$v' \in V_{nk}$ repetindo-se cada letra de v k vezes.

Note que $v' \in R$, logo temos codificação de canal $T : V_n \to R$ tal que $T(v) = v' \; \forall \; v \in V_n$.

Exemplo 13. *(Código de Hamming)*
Em $\mathbb{F} = \mathbb{Z}_2$*, considere o subconjunto* W *de* V_7 *dado por*

$$W = \{v \in V_7 |\; v_5 = v_2 + v_3 + v_4 \text{ e } v_6 = v_1 + v_3 + v_4 \text{ e } v_7 = v_1 + v_2 + v_3\}.$$

- $\overrightarrow{0}$ é tal que $\overrightarrow{0}_i = 0 \; \forall \; i \in \{1, \ldots, n\}$, logo

  $$\overrightarrow{0}_5 = \overrightarrow{0}_2 + \overrightarrow{0}_3 + \overrightarrow{0}_4 = \overrightarrow{0}$$

  e

  $$\overrightarrow{0}_6 = \overrightarrow{0}_1 + \overrightarrow{0}_3 + \overrightarrow{0}_4 = \overrightarrow{0}$$

  e

  $$\overrightarrow{0}_7 = \overrightarrow{0}_1 + \overrightarrow{0_2} + \overrightarrow{0}_4 = \overrightarrow{0}$$

  logo $\overrightarrow{0} \in W$.

- Sejam $u, v \in W$ quaisquer. Então

  $$v_5 = v_2 + v_3 + v_4 \text{ e } v_6 = v_1 + v_3 + v_4 \text{ e } v_7 = v_1 + v_2 + v_4$$

  e

  $$u_5 = u_2 + u_3 + u_4 \text{ e } u_6 = u_1 + u_3 + u_4 \text{ e } u_7 = u_1 + u_2 + u_4.$$

  Assim,

  $$v_5 + u_5 = v_2 + u_2 + v_3 + u_3 + v_4 + u_4$$

  e

  $$v_6 + u_6 = v_1 + u_1 + v_3 + u_3 + v_4 + u_4$$

  e

  $$v_7 + u_7 = v_1 + u_1 + v_2 + u_2 + v_4 + u_4.$$

  Como $(v + u)_i = v_i + u_i \; \forall \; i \in \{1, \ldots n\}$, temos que $v + u \in W$.

- Sejam $v \in W$ e $\alpha \in \mathbb{Z}_2$ qualquer. Então

  $$v_5 = v_2 + v_3 + v_4 \text{ e } v_6 = v_1 + v_3 + v_4 \text{ e } v_7 = v_1 + v_2 + v_4.$$

  Assim,

  $$\alpha v_5 = \alpha(v_2 + v_3 + v_4) = \alpha v_2 + \alpha v_3 + \alpha v_4,$$

,

$$\alpha v_6 = \alpha(v_1 + v_3 + v_4) = \alpha v_1 + \alpha v_3 + \alpha v_4$$

e

$$\alpha v_7 = \alpha(v_1 + v_2 + v_4) = \alpha v_1 + \alpha v_2 + \alpha v_4.$$

Portanto, como $(\alpha v)_i = \alpha v_i$, temos que $\alpha v \in W$.

Pelo exposto, W é um subespaço vetorial de V_7.

Para codificação: V_4 é o código da fonte, W é o código do canal e T é a *codificação de canal*. Portanto temos que

$$v = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{bmatrix} \in V_4 \xrightarrow{T} v' = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_2 + v_3 + v_4 \\ v_1 + v_3 + v_4 \\ v_1 + v_2 + v_4 \end{bmatrix} \in W.$$

3.2.1 Distância Mínima de um Código Linear

Para um código linear W de comprimento n, dadas duas palavras $v, w \in W$ podemos fazer a diferença entre elas,

$$v - w = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix} - \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix} = \begin{bmatrix} v_1 - w_1 \\ v_2 - w_2 \\ \vdots \\ v_n - w_n \end{bmatrix}.$$

perceba que as palavras v e w diferem na letra i se e somente se as entradas v_i e w_i forem diferentes, ou seja $v_i \neq w_i$, o que ocorre se e somente se $v_i - w_i \neq 0$. Assim, podemos perceber que para determinar a distância de Hamming entre v e w basta olharmos para o número de entradas de $v - w$. Disso segue a seguinte definição.

Definição 3.9. *Dados um código linear W de comprimento n e $v \in W$, o* peso de v *é o número denotado por $w(v)$ e dado pelo número de entradas não nulas de v.*

Note que, em particular, como $w(v)$ é o número de entradas não nulas de v, a distância de v ao vetor nulo $\vec{0}$ será exatamente $w(v)$, ou

seja $w(v) = d(v, \vec{0})$. Como a distância mínima de um código é a menor das distâncias possível entre palavras distintas do código, temos de imediato o seguinte resultado.

Proposição 3.4. *Para um código linear W, a distância mínima de W é*

$$d = \min\{w(v) | v \in W \setminus \{\vec{0}\}\} = \min\{d(v, \vec{0}) | v \in W \setminus \{\vec{0}\}\}.$$

Logo, temos que para determinar a distância mínima de um código linear W com M vetores, basta realizarmos o cálculo de $M - 1$ distâncias de Hamming. Perceba que quando este M for grande, isto é muito melhor do que calcular todas as $\frac{M(M-1)}{2}$ distâncias entre os pares de elementos distintos do código linear W para determinar qual delas é a menor.

3.3 MATRIZES GERADORAS DE UM CÓDIGO LINEAR

Definição 3.10. *Dados $n, k \in \mathbb{N}^*$ tais que $k \leq n$. Se $k \leq n$, dizemos que uma matriz $R \in M_{n\times k}(\mathbb{F})$ tem* forma padrão *se as primeiras k linhas de R formam matriz identidade $k \times k$.*

Note que uma matriz na forma *padrão* é como a que segue:

$$\begin{bmatrix} I_{k\times k} \\ A_{(n-k)\times k} \end{bmatrix}.$$

Definição 3.11. *Dado um código linear W de comprimento n tal que* $\dim(W) = k$, *uma* matriz geradora *para W consiste de uma matriz $G \in M_{n\times k}(\mathbb{F})$ cujas k colunas formam uma base de W.*

Segundo a Definição 3.11, temos que, se

$$G = \begin{bmatrix} | & | & | & | \\ w_1 & w_2 & \dots & w_k \\ | & | & | & | \end{bmatrix}_{n\times k} \quad e \ v = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_k \end{bmatrix} \in V_k, \ \textit{então}$$

$$G \cdot v = \begin{bmatrix} | \\ \alpha_1 w_1 + \alpha_2 w_2 + \dots + \alpha_k w_k \\ | \end{bmatrix} = \alpha_1 w_1 + \alpha_2 w_2 + \dots + \alpha_k w_k \in W.$$

Uma matriz geradora G para um código linear W pode ser usada para fazer uma codificação de canal, veja: seja V_k o código da fonte, e W o código de canal. Para cada palavra $v \in V_k$, podemos produzir a palavra $v' \in W$ dada por

$$v' = G \cdot v.$$

Observe que v' é combinação linear das colunas de G, que por sua vez formam uma base de W, logo de fato $v' \in W$.

Se a matriz G está na forma padrão, então esta codificação de canal se dá de uma maneira bastante simples: o vetor v' será obtido a partir de v apenas adicionando algumas redundâncias nas últimas entradas: Por exemplo, se $\mathbb{F} = \mathbb{R}$, $k = 3$ e $n = 5$, e se tivermos a matriz geradora G para um código linear W de comprimento 5 na forma padrão dada por

$$G = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 7 & 2 & 5 \\ 3 & 8 & 4 \end{bmatrix}$$

então para cada vetor v do código da fonte V_3, digamos

$$v = \begin{bmatrix} a \\ b \\ c \end{bmatrix}$$

tem-se que v' é dado por

$$v' = G \cdot v = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 7 & 2 & 5 \\ 3 & 8 & 4 \end{bmatrix} \cdot \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} a \\ b \\ c \\ 7a+2b+5c \\ 3a+8b+4c \end{bmatrix} = \begin{bmatrix} | \\ v \\ | \\ 7a+2b+5c \\ 3a+8b+4c \end{bmatrix}.$$

Conseguimos com isso perceber que as três primeiras entradas de v' correspondem ao vetor v, e as demais entradas são obtidas a partir das três primeiras, caracterizando-se assiim, como redundâncias.

Com o próximo exemplo, mostraremos uma matriz geradora do código de Hamming.

Exemplo 14. *Seja G a matriz dada por*

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

Mostraremos que G é matriz geradora do código de Hamming, segundo a Definição 3.11. Pela Definição 3.11, temos que

$$v' = G \cdot v,$$

então

$$v' = G = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = \begin{bmatrix} a \\ b \\ c \\ d \\ b+c+d \\ a+c+d \\ a+b+d \end{bmatrix} = \begin{bmatrix} | \\ v \\ | \\ b+c+d \\ a+c+d \\ a+b+d \end{bmatrix}.$$

Conseguimos perceber que as quatro primeiras entradas do vetor v' são correspondentes ao vetor v e as demais entradas são formadas a partir das quatro primeiras, caracterizando assim de G como uma matriz geradora do código de Hamming.

Se transportarmos este exemplo para os exemplos que trabalhamos com diagramas de Venn, percebemos que as quatro primeiras entradas correspondem às entradas que no diagrama de Venn chamávamos de d_1, d_2, d_3, d_4 e que as demais três entradas correspondem às entradas que no diagrama chamávamos de v_1, v_2, v_3.

Para um dado código linear W de comprimento n e dimensão k, podemos construir diversas matrizes geradoras para esse código W, no caso, uma para cada base ordenada de W, mas isso não garante que teremos matriz geradora para W na forma padrão.

Por exemplo, se W é o subespaço vetorial de V_3 dos vetores com primeira entrada igual a zero, então W não possui matriz geradora na forma padrão, pois não existe vetor em W com primeira entrada igual a 1, e portanto não poderíamos fazer a primeira coluna da matriz

identidade na matriz geradora.

Entretanto, podemos encontrar uma matriz geradora G tal que, após uma *permutação* de suas linhas, consigamos produzir uma matriz G' na forma padrão. Esta nova matriz não será geradora do código W, mas também poderá ser usada para fazer codificação de canal.

As colunas de G' geram um outro código linear W' também de comprimento n e dimensão k, que por sua vez pode ser dito *equivalente* ao código W. A ideia é então usar W' como código de canal e G' para efetuar a codificação de canal, já que está na forma padrão.

Para fazermos isso, é preciso relembrar as operações elementares nas colunas de uma matriz, são elas:

(C1) Permutação de duas colunas;

(C2) Multiplicação de uma coluna por um escalar não nulo;

(C3) Adição de um múltiplo escalar de uma coluna a outra.

O importante para nós é lembrar que usar operações elementares nas colunas de uma matriz $R \in M_{n\times k}(\mathbb{F})$ não altera o subespaço vetorial W de V_n gerado pelas colunas desta matriz; então, se as colunas de R formam uma base de W, as operações elementares sobre as colunas de R sempre resultarão em novas colunas que também serão base de W.

Teorema 3.2. *Seja W um código linear de comprimento n e dimensão k. Então existe uma matriz geradora G para W e uma permutação de suas linhas que produz uma matriz G' na forma padrão.*

Demonstração. Suponha que W é um subespaço vetorial de V_n, com $\dim(W) = k$. Logo, existe base para W com k vetores, digamos $B = \{v_1^1, \dots v_k^1\}$. Considere a matriz

$$A = \begin{bmatrix} | & | & | \\ v_1^1 & v_2^1 & \dots v_k^1 \\ | & | & | \end{bmatrix}_{n\times k}.$$

Como B é uma base de W, os vetores $v_1^1, \dots, v_k^1$ são todos não nulos e, em particular, $v_1^1 \neq \overrightarrow{0}$. Assim, existe um $i_1 \in \{1, \dots n\}$ tal que $(v_1^1)_{i_1} \neq 0$.

Usaremos a operação (C2) na primeira coluna desta matriz, multiplicando-a por $(v_1^1)_{i_1}^{-1}$(que existe pois $(v_1^1)_{i_1} \neq 0$ e $\mathbb{F}$ é corpo), obtendo

uma nova matriz A_1' cuja primeira coluna v_1^2 tem entrada i_1 igual a $(v_1^1)_{i_1} \cdot (v_1^1)_{i_1}^{-1} = 1$, a unidade do corpo $\mathbb{F}$.

Em seguida, usaremos a operação (C3) sucessivamente, de modo a zerar a entrada i_1 de todas as colunas, da segunda em diante: mais especificamente, para cada $j \in \{2, \ldots, k\}$, trocamos a j-ésima coluna v_j^1 da matriz A_1' pelo vetor v_j^2 dado por

$$v_j^2 = v_j^1 - (v_j^1)_{i_j} \cdot v_1^2.$$

Obtemos deste modo uma nova matriz $A_2 \in M_{n\times k}(\mathbb{F})$ cujas colunas $v_1^2 \ldots . v_k^2$ ainda são uma base para W, e de modo que para todo $j \in \{1, \ldots, k\}$ tem-se que

$$(v_j^2)_{i_1} = \begin{cases} 1 \text{ se } j = 1 \\ 0 \text{ se } j \neq 1. \end{cases}$$

Sabemos que $v_2^2 \neq \vec{0}$, pois a base de W, logo existe $i_2 \in \{1, \ldots, n\}$ tal que $(v_2^2)_{i_2} \neq 0$; além disso $i_2 \neq i_1$, pois $(v_2^2)_{i_1} = 0$ por construção.

Prosseguindo como feito no caso da matriz A_1, usamos a operação (C2) na segunda coluna de A_2, multiplicando-a por $(v_2^2)_{i_2}^{-1}$ para chegar em uma nova matriz A_2' cuja segunda coluna v_2^3 tem entrada $i_2 = 1$ (e entrada i_1 ainda igual a zero). Agora, usamos (C3) sucessivamente para que, para cada $j \in \{1, \ldots, k\}$ com $j \neq 2$, troquemos a j-ésima coluna v_j^2 de A_2' pelo vetor v_j^3 dado por

$$v_j^3 = v_j^2 - (v_j^2)_{i_2} \cdot v_2^3. \tag{3.1}$$

Assim, chegamos a uma matriz $A_3 \in M_{n\times k}(\mathbb{F})$ cujas colunas $v_1^3, \ldots, v_k^3$ ainda são uma base para W, e tal que para qualquer $j \in \{1, \ldots, k\}$ temos que

$$(v_j^3)_{i_2} = \begin{cases} 1 \text{ se } j = 1 \\ 0 \text{ se } j \neq 1 \end{cases} .$$

Mais ainda, uma vez que $(v_2^3)_{i_1} = 0$ por construção, temos da Equação 4.1 que $(v_j^3)_{i_j} = (v_j^2)_{i_1}$ para todo j, e assim, segue que

$$(v_j^3)_{i_1} = \begin{cases} (v_j^2)_{i_1} = 1 \text{ se } j = 1 \\ (v_j^2)_{i_1} = 0 \text{ se } j \neq 1 \end{cases} .$$

Em outras palavras, a primeira entrada da i_1-ésima linha de A_3 é igual a 1, e todas as outras zero; a segunda da i_2-ésima linha de A_3 é igual a 1, e todas as demais zero. Prosseguindo desta forma, chegaremos a uma matriz $A_k \in M_{n\times k}(\mathbb{F})$ cujas colunas $v_1^k, \ldots, v_k^k$ são base de W, e em inteiros distintos $i_1, \ldots, i_k \in \{1, \ldots, n\}$ de modo que para todo $r \in \{1, \ldots, k\}$, a r-ésima entrada da linha i_r de A_k é igual a 1, e todas as demais entradas desta linha são iguais a zero; resumindo, para cada $r \in \{1, \ldots, k\}$ tem-se que a i_r-ésima linha de A_k é igual a e_r^t, onde e_r é o vetor da base canônica de A_k e e_r^t é a transposta deste vetor.

Seja $G = A_k$. Então, pelo que vimos G é matriz geradora para W. Além disso, se permutarmos as linhas de G, trocando para cada $r \in \{1, \ldots, k\}$ a linha r com a linha i_r, obteremos uma nova matriz G' que, por construção, tem as k primeiras linhas formando a matriz identidade $k \times k$, e portanto está na forma padrão. □

3.4 MATRIZES DE TESTE DE PARIDADE

Para darmos continuidade, é preciso fixarmos algumas notações e relembrar resultados de Álgebra Linear.

Teorema 3.3. *Sejam $n, m \in \mathbb{N}^*$ e $T : V_n \longrightarrow V_m$ uma transformação linear. Então existe uma matriz $A \in M_{m\times n}(\mathbb{F})$ tal que para qualquer $v \in V_n$ tem-se que*

$$T(v) = A \cdot v.$$

Demonstração. Sendo $B = \{e_1, \ldots, e_n\}$ e $C = \{g_1, \ldots, g_m\}$ as bases canônicas de V_n e V_m, respectivamente, para cada $j \in \{1, \ldots, n\}$ existem escalares $\alpha_{1j}, \ldots, \alpha_{mj}$ tais que

$$T(e_1) = \alpha_{11}g_1 + \alpha_{21}g_2 + \ldots + \alpha_{m1}g_m = \alpha_{11} \cdot \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} + \ldots + \alpha_{m1} \cdot \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha_{11} \\ \alpha_{21} \\ \vdots \\ \alpha_{m1} \end{bmatrix}$$

$$\vdots$$

$$T(e_j) = \alpha_{1j}g_1 + \alpha_{2j}g_2 + \ldots + \alpha_{mj}g_m = \alpha_{1j} \cdot \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} + \ldots + \alpha_{mj} \cdot \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha_{1j} \\ \alpha_{2j} \\ \vdots \\ \alpha_{mj} \end{bmatrix} =$$

$$T(e_j) = \Sigma_{i=1}^{m} \alpha_{1j} g_i = \begin{bmatrix} \alpha_{1j} \\ \alpha_{2j} \\ \vdots \\ \alpha_{mj} \end{bmatrix}.$$

Seja A a matriz $m \times n$ dada por $A = (\alpha_{ij})_{ij}$.

$$A = \begin{bmatrix} | & | & & | \\ T(e_1) & T(e_2) & \ldots & T(e_n) \\ | & | & & | \end{bmatrix} = \begin{bmatrix} \alpha_{11} & \ldots & \alpha_{1n} \\ \alpha_{21} & \ldots & \alpha_{2n} \\ \vdots & \ldots & \vdots \\ \alpha_{m1} & \ldots & \alpha_{mn} \end{bmatrix} = (\alpha_{ij})_{ij}.$$

Note que A é a matriz cujas colunas são os vetores $T(e_j)$. Para cada $j \in \{1, \ldots, n\}$, tem-se que

$$A \cdot e_j = \begin{bmatrix} \alpha_{1j} \\ \alpha_{2j} \\ \vdots \\ \alpha_{mj} \end{bmatrix} = T(e_j).$$

Segue disso, da linearidade de T e da distributividade da multiplicação de matrizes pela adição que, para qualquer $v \in V_n$, $v = \beta_1 e_1 + \ldots + \beta_n e_n$. Então

$$\begin{aligned} T(v) &= \beta_1 T(e_1) + \ldots + \beta_n T(e_n) \\ &= \beta_1 A \cdot e_1 + \ldots + \beta_n A \cdot e_n \\ &= A(\beta_1 e_1 + \ldots + \beta_n e_n) = A \cdot v. \end{aligned}$$

□

Mostraremos a seguir que todo subespaço vetorial de V_n é núcleo de alguma transformação linear.

Para um espaço vetorial V sobre um corpo $\mathbb{F}, k \in \mathbb{N}^*$ e vetores $v_1, \ldots, v_k \in V$, denotaremos o subespaço vetorial de V gerado por estes vetores por

$$[v_1, \ldots, v_k].$$

De Álgebra Linear, temo que para $n, m \in \mathbb{N}^*$, espaços vetoriais V, U sobre um corpo $\mathbb{F}$ com dimensões n e m respectivamente $B = \{v_1, \ldots, v_n\}$ base de V e $T : V \longrightarrow U$ uma transformação linear, a imagem de T é o subespaço de U dado por

$$[T(v_1), \ldots, T(v_n)]\,.$$

Em particular, se A é a matriz associada a transformação T como mostrado no resultado acima, então temos que $Im(T)$ é o subespaço de V_n gerado pelas colunas de A.

Teorema 3.4. *Sejam V, U espaços vetoriais sobre um corpo $\mathbb{F}$, $n \in \mathbb{N}^*$, $B = \{v_1, \ldots, v_n\}$ base de V e $C = \{u_1, \ldots, u_n\}$ um subconjunto de U. Então existe uma única transformação linear $T : V \to U$ de modo que para qualquer $i \in \{1, \ldots, n\}$ tem-se que $T(v_i) = u_i$.*

Teorema 3.5. *Sejam $n, k \in \mathbb{N}^*$ e W um subespaço vetorial de V_n com $\dim(W) = k$. Então, existe uma transformação linear $T : V_n \to V_{n-k}$ de modo que $\ker(T) = W$.*

Demonstração. Por hipótese $\dim(W) = k$. Seja $\{w_1, \ldots, w_k\}$ base de W, e complete-a para obter uma base B' de V_n, digamos

$$B' = \{w_1, \ldots, w_k, v_1, \ldots, v_{n-k}\}.$$

Agora, considerando $C = \{g_1, \ldots, g_{n-k}\}$ uma base qualquer de V_{n-k}, seja $T : V_n \to V_{n-k}$ a única transformação linear tal que $T(w_i) = \vec{0}$ para todo $i \in \{1, \ldots, k\}$ e $T(v_i) = g_i$ para todo $i \in \{1, \ldots, n-k\}$ garantida pelo resultado anterior.
Por construção $w_i \in \ker(T)$ para todo $i \in \{1, \ldots, k\}$, logo $W \subseteq \ker(T)$.

Além disso,

$$\begin{aligned} Im(T) &= [T(w_1), \ldots, T(w_k), T(v_i), \ldots, T(v_{n-k})] \\ &= [T(v_1), \ldots, T(v_{n-k})] \\ &= [g_1, \ldots, g_{n-k}] \\ &= V_{n-k}, \end{aligned}$$

logo $\dim(Im(T)) = n - k$. Pelo teorema da dimensão,

$$\dim(\ker(T)) = n - \dim(Im(T)) = n - (n-k) = k.$$

Finalmente, como $\dim(W) = k$ e $W \subseteq \ker(T)$, segue que $W = \ker(T)$, como desejado. □

Corolário 3.1. *Sejam $n, k \in \mathbb{N}^*$ e W um subespaço vetorial de V_n com $\dim(W) = k$. Então, existe uma matriz $H \in M_{(n-k)\times n}(\mathbb{F})$ tal que*

$$W = \{v \in V_n | H \cdot v = \vec{0} \in V_{n-k}\}.$$

Demonstração. Seja W um subespaço vetorial de V_n, com $\dim W = k$. Pelo Teorema 3.5, W é núcleo de uma transformação linear.

$$T : V_n \to V_{n-k}.$$

Pelo Teorema 3.3, existe uma matriz $H \in M_{(n-k)\times n}(\mathbb{F})$ tal que

$$T(v) = H \cdot v, \forall v \in V_n.$$

$$\begin{aligned} W = \ker T &= \{v \in V_n | T(v) = \vec{0} \in V_{n-k}\} \\ &= \{v \in V_n | H \cdot v = \vec{0} \in V_{n-k}\}. \end{aligned}$$

□

Definição 3.12. *Dados $n, k \in \mathbb{N}^*$ e um código linear W de comprimento n tal que $\dim(W) = k$, uma* matriz de teste de paridade *para W consiste de uma matriz $H \in M_{(n-k)\times n}(\mathbb{F})$ tal que*

$$W = \{v \in V_n | H \cdot v = \vec{0} \in V_{n-k}\}.$$

Observação 3.2. *Com os resultados mostrados acima, conseguimos afirmar que um código linear W sempre admite matrizes de teste de paridade.*

Teorema 3.6. *Sejam $n, k \in \mathbb{N}^*$, W um código linear de comprimento n tal que $\dim(W) = k$ com matriz geradora $G \in M_{n\times k}(\mathbb{F})$, e seja $H \in M_{(n-k)\times n}(\mathbb{F})$ uma matriz com $n-k$ colunas linearmente independentes. Então, H é matriz de teste de paridade para W se e somente se $H \cdot G$ é a matriz nula em $M_{(n-k)\times n}(\mathbb{F})$.*

Demonstração. ($\Rightarrow$) Suponha que H é matriz de teste de paridade para W. Por hipótese G é matriz geradora para W, logo W é gerado pelas colunas de G. Em outras palavras,

$$W = [G \cdot e_1, \ldots, G \cdot e_k].$$

em que $\{e_1, \ldots, e_k\}$ é base canônica de V_k. Por outro lado, $W = \{v \in V_n | H \cdot v = \vec{0} \in V_{n-k}\}$; em particular, para $i \in 1, \ldots, k$,

$$(H \cdot G) \cdot e_i = H \cdot (G \cdot e_i) = \vec{0} \in V_{n-k},$$

garantindo que todas as colunas da matriz $H \cdot G$ são nulas, e portanto que $H \cdot G$ é a matriz nula em $M_{(n-k)\times k}(\mathbb{F})$.
($\Leftarrow$) Suponha agora que $H \cdot G$ é a matriz nula em $M_{(n-k)\times k}(\mathbb{F})$, e seja

$$W' = \{v \in V_n | H \cdot v = \vec{0} \in V_{n-k}\}.$$

Mostraremos que $W' = W$: dado $w \in W$ qualquer, w é combinação linear das colunas de G, logo existe $v \in V_k$ tal que

$$w = G \cdot v,$$

e portanto

$$H \cdot w = H \cdot (G \cdot v) = (H \cdot G) \cdot v = \vec{0} \in V_{n-k}.$$

Isto mostra que $w \in W'$, e assim que $W \subseteq W'$. Agora, considere a transformação linear $T_H : V_n \to V_{n-k}$ de multiplicação por H. Assim,

$$\begin{aligned} W' &= \{v \in V_n | H \cdot v = \vec{0} \in V_{n-k}\} \\ &= \{v \in V_n | T_H(v) = \vec{0} \in V_{n-k}\} \\ &= \ker(T_H). \end{aligned}$$

A imagem de T_H é o subespaço vetorial de V_{n-k} gerado pelas colunas de H; como H possui $n-k$ colunas linearmente independentes, segue que $\dim(Im(T_H)) = n-k$, e do teorema da dimensão podemos concluir que $\dim(W') = \dim(\ker(T_H)) = k$. Uma vez que $\dim(W) = k$ e $W \subseteq W'$,

segue disso que $W = W'$, como queríamos provar. □

Quando a matriz geradora de um código W estiver na forma padrão, fica fácil determinar a matriz de teste de paridade.

Corolário 3.2. *Sejam $n, k \in \mathbb{N}^*$ e W um código linear de comprimento n tal que $\dim(W) = k$ com matriz geradora $G \in M_{n\times k}(\mathbb{F})$ na forma padrão,*

$$G = \begin{bmatrix} I_{k\times k} \\ A_{(n-k)\times k} \end{bmatrix}.$$

Então, a matriz $H \in M_{(3)\times 7}(\mathbb{F})$ dada por

$$\begin{bmatrix} -A & I_{(n-k)\times(n-k)} \end{bmatrix}$$

é uma matriz de teste de paridade para W.

Demonstração. Uma simples verificação direta por multiplicação de matrizes mostra que $H \cdot G$ resulta na matriz nula de tamanho $(n-k)\times k$. Como as últimas $n-k$ colunas de H são linearmente independentes, o resultado segue pelo teorema anterior. □

Exemplo 15. *Seja o código Hamming W, que possui comprimento $n = 7$ e $\dim W = 4$, com matriz geradora G na forma padrão*

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

Então a matriz $H \in M_{(n-k)\times n}(\mathbb{F})$ é tal que

$$H = \begin{bmatrix} 0 & -1 & -1 & -1 & 1 & 0 & 0 \\ -1 & 0 & -1 & -1 & 0 & 1 & 0 \\ -1 & -1 & 0 & -1 & 0 & 0 & 1 \end{bmatrix}$$

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

Exemplo 16. *Com os exemplos que já trabalhamos, mostramos que um mesmo código linear W pode admitir diversas matrizes de teste de paridade. Assim, uma outra matriz de teste de paridade para o código de Hamming é dada por*

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

Esta matriz é muito interessante pois suas colunas são compostas pelos números naturais d 1 a 7 escritos no sistema binário.

Matrizes de teste de paridade são uteis na detecção e correção de possíveis erros de transmissão, mais exemplos a quem possa interessar estão presentes no livro [HEFEZ, 2008]. Neste momento, mostraremos como usar a matriz de teste de paridade H, mostrada anteriormente, para o Código de Hamming, para detectar e corrigir erros no processo de transmissão.

Seja W o código de Hamming e $w \in W$ uma palavra, que possui comprimento 7,

$$\begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_7 \end{bmatrix}.$$

Suponha que na transmissão desta palavra sabemos que ocorreu exatamente um erro, ou seja, que a palavra recebida possui uma letra errada em uma posição $i \in \{1, \ldots, 7\}$ que não sabemos qual é, mas precisamos determinar.

Observe que se o erro foi na transmissão da i-ésima letra de w, então a palavra recebida foi

$$v = w + e_i,$$

em que e_i é o i-ésimo vetor da base canônica de V_7; de fato, e_i tem 1 na entrada i e zero nas demais, logo $w + e_i$ tem as mesmas entradas que w, exceto a i-ésima que é igual a $w_i + 1$ e portanto será igual a 0 se $w_i = 1$, e igual a 1 se $w_i = 0$. Agora, como H é matriz de teste de paridade para o código de Hamming e $w \in W$, temos $H \cdot w = \vec{0} \in V_3$. Assim,

$$H \cdot v = H \cdot (w + e_i) = H \cdot w + H \cdot e_i = \vec{0} + H \cdot e_i = H \cdot e_i,$$

que é a i-ésima coluna de H; mas a i-ésima coluna de H é formada pela representação binária do número i, logo o cálculo de $H \cdot v$ nos indica exatamente onde está o erro na transmissão!

Vamos aplicar as matrizes de teste de paridade para verificar e corrigir erros, nos exemplos que resolvemos utilizando o Diagrama de Venn.

Exemplo 17. *Considere que em uma transmissão a mensagem recebida seja* 1111000 *como no Exemplo 2. A mensagem recebida gera a matriz*

$$v = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

Efetuando a multiplicação da matriz H pela matriz v teremos:

$$H \cdot v = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}.$$

Esse resultado nos mostra que o erro encontra-se na quarta entrada da matriz pois 100 é o número 4 em binário, logo a mensagem correta é 1110000.

Exemplo 18. *Considere agora a outra mensagem recebida* 1011011 *como no Exemplo 3. A mensagem recebida gera a matriz*

$$v = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 1 \\ 0 \\ 1 \\ 1 \end{bmatrix}.$$

Efetuando a multiplicação da matriz H pela matriz v teremos:

$$H \cdot v = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 0 \\ 1 \\ 1 \\ 0 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}.$$

Esse resultado nos mostra que o erro encontra-se na sétima entrada da matriz pois o número 111 representa o número 7 em binário, logo a mensagem correta é 1011010.

4 CONCLUSÃO

Ao término deste trabalho, conseguimos encontrar uma matriz de teste de paridade, capaz de detectar e corrigir um erro, dizendo exatamente em qual dígito encontra-se o erro.

Para chegar a este resultado, foi necessário que construíssemos vários resultados como provar que a distância de Hamming é uma métrica, provar a quantidade máxima de erros que um código é capaz de corrigir, também mostrar que um código corretor possui distância mínima.

Mostramos também que um código pode ser representado na forma de um vetor e também de uma matriz, e com isso chegarmos ao nosso resultado. Para isso foi necessário termos conhecimentos sobre como operar com vetores e matrizes.

É importante salientar que com a elaboração deste trabalho apresentamos um exemplo de uma matriz de teste de paridade para o Código de Hamming, mas esta não é a única, e sim um tipo especial que nos mostra exatamente em qual dígito o erro se encontra.

REFERÊNCIAS

[1] D'ORNELAS, Stephanie. Como sue cerbero pedo lre itso. 2012. Disponível em: https://hypescience.com/como-sue-cerbero-pedo-lre-itso. Acesso em: 14 set. 16.

[2] HEFEZ, Abramo; VILLELA, Maria Lúcia T. **Códigos corretores de erros**. Instituto de Matematica Pura e Aplicada, 2008.

[3] ABRANTES, Sílvio A. **Códigos Correctores de Erros em Comunicações Digitais**. FEUP edições, 2010.

4] MILIES, César Polcino. **Breve introdução a Teoria dos Códigos Corretores de Erros**. Departamento de Matemática, UFMS, 2009.

[5] BARBOSA, Lucas Diego Antunes; HOYOS, Mariana Garabini Cornelissen. **Uma aplicaçao ao estudo de Matrizes**. Departamento de Matemática, UFSJ, 2015.

[6] BOLDRINI, José Luiz; COSTA, Sueli I. Rodrigues; FIGUEREDO, Vera Lúcia, WETZLER, Henry G. **Álgebra Linear**. São Paulo: Harper and Row do Brasil, 1980.

[7] LIPSCHUTZ, Seymour; Lipson, Marc Lars. **Álgebra Linear**. Porto Alegre: Bookmann, 2011.

[8] HEFEZ, Abramo; FERNANDEZ, Cecília de Souza. **Introdução à álgebra linear** Rio de Janeiro: SBM, 2012.