UFRRJ

INSTITUTO DE CIÊNCIAS EXATAS

MESTRADO PROFISSIONAL EM MATEMÁTICA EM REDE NACIONAL

DISSERTAÇÃO

Códigos Corretores de Erros: Exemplos da Matemática Aplicada em Situações do Cotidiano.

Raphael Bruno Rodrigues da Silveira

2015

UNIVERSIDADE FEDERAL RURAL DO RIO DE JANEIRO
INSTITUTO DE CIÊNCIAS EXATAS
MESTRADO PROFISSIONAL EM MATEMÁTICA EM REDE NACIONAL – PROFMAT

CÓDIGOS CORRETORES DE ERROS: EXEMPLOS DA MATEMÁTICA APLICADA EM SITUAÇÕES DO COTIDIANO.

RAPHAEL BRUNO RODRIGUES DA SILVEIRA

Sob a Orientação do Professor

Dr. André Luiz Martins Pereira

Dissertação de Mestrado apresentada ao Mestrado Profissional em Matemática em Rede Nacional – PROFMAT – da Universidade Federal Rural do Rio de Janeiro, como requisito parcial à obtenção do título de **Mestre em Matemática**.

Seropédica, RJ

Maio de 2015

510.7
S587c
T

Silveira, Raphael Bruno Rodrigues da, 1982-
Códigos corretores de erros: exemplos da matemática aplicada em situações do cotidiano / Raphael Bruno Rodrigues da Silveira. – 2015.
99 f.: il.

Orientador: André Luiz Martins Pereira.
Dissertação (mestrado) – Universidade Federal Rural do Rio de Janeiro, Curso de Mestrado Profissional em Matemática em Rede Nacional, 2015.
Bibliografia: f. 92-95.

1. Matemática – Estudo e ensino – Teses. 2. Códigos corretores de erros (Teoria da informação) – Teses. 3. Sistemas lineares – Teses. 4. Matrizes (Matemática) – Teses. 5. Ensino auxiliado por computador – Teses. I. Pereira, André Luiz Martins, 1980- II. Universidade Federal Rural do Rio de Janeiro. Curso de Mestrado Profissional em Matemática em Rede Nacional. III. Título.

UNIVERSIDADE FEDERAL RURAL DO RIO DE JANEIRO
INSTITUTO DE CIÊNCIAS EXATAS
CURSO DE PÓS-GRADUAÇÃO EM MESTRADO PROFISSIONAL EM MATEMÁTICA EM REDE NACIONAL – PROFMAT

RAPHAEL BRUNO RODRIGUES DA SILVEIRA

Dissertação submetida como requisito parcial para obtenção do grau de **Mestre**, no curso de Pós-Graduação em Mestrado Profissional em Matemática em Rede Nacional – PROFMAT, área de Concentração em Matemática.

DISSERTAÇÃO APROVADA EM 28/05/2015

André Luiz Martins Pereira - Doutor em Matemática - UFRRJ
(Orientador)

Luciano Vianna Félix - Doutor em Matemática - UFRRJ

Albetã Costa Mafra - Doutor em Matemática - UFRJ

DEDICATÓRIA

Ao meu Deus, fonte de vida, conhecimento e inspiração; à minha esposa e à minha filha, meus verdadeiros tesouros nesse mundo; aos meus amigos; e a todos que acreditam na educação como o caminho para desenvolvimento de nosso país.

AGRADECIMENTOS

A Deus, pois sem Ele eu não teria conseguido alcançar essa bênção, que é a conclusão do curso de Mestrado.

Aos meus pais, Damil e Edionea, por terem me ensinado o caminho a seguir e pelas constantes orações.

À minha amada esposa, pela compreensão, orações e palavras de fortalecimento em todos os momentos.

Ao meu orientador, André, que além de ter sido professor de duas disciplinas durante o curso, sempre esteve presente nos momentos em que precisei, aconselhando-me durante a elaboração deste trabalho.

Aos meus queridos amigos e professores da turma do PROFMAT/UFRRJ-2013 pelos momentos que passamos juntos.

À SBM e à CAPES que idealizaram este maravilhoso curso de pós-graduação *stricto sensu*.

“*Melhor é a sabedoria do que a força.*”

Eclesiastes capítulo 9, verso 16 – Bíblia Sagrada

RESUMO

Este trabalho tem como objetivo oferecer uma sequência didática que permita a contextualização do ensino de sistemas de numeração, vetores, matrizes e sistemas lineares por meio dos códigos corretores de erros. Na da mencionada sequência de atividades é realizada a contextualização histórica do surgimento dos códigos corretores, com o uso de situações-problemas para motivar os educandos. Além disso, o computador é utilizado como um colaborador no processo de ensino-aprendizagem. Os problemas são apresentados de forma que os alunos possam perceber o que motivou Hamming a se dedicar à pesquisa sobre os códigos corretores, pois são apresentadas situações em que: não será possível a detecção de erros; será possível a detecção, mas não a correção; será possível a detecção, mas não a correção com confiança; e será possível a detecção e a correção com certo grau de certeza. Assim, por meio da mencionada sequência didática, a qual possui um enfoque construtivista de construção do conhecimento, espera-se que os estudantes, ao final das atividades, compreendam que a Matemática é útil para resolver diversos problemas do dia-a-dia e sejam capazes de verificar que os conteúdos estudados no Ensino Médio possuem mais aplicações no cotidiano do que eles imaginam.

Palavras-chave: Códigos Lineares Corretores de Erros. Ensino de Matemática. Contextualização. Matrizes e Sistemas Lineares.

ABSTRACT

This paper aims to offer a didactic sequence to enable the contextualization of teaching the numbering systems, vectors, matrices and linear systems using error correcting codes. Through the mentioned sequence of activities is carried out at historical background of the emergence of error correcting codes, with the use of application problems that depict real situations of day-to-day to motivate students. In addition, the computer is used as a collaborator in the teaching and learning process. The problems are presented in a way that students can understand what motivated Hamming to research on error correcting codes, because situations are presented in which: no error detection is possible; detection is possible, but not correction; detection is possible, but not the correction with confidence; and it is possible the detection and correction with some degree of certainty. Thus, through the mentioned didactic sequence, which has a constructivist approach to knowledge construction, it is expected that students at the end of the activities, understand how mathematics is useful to solve problems and be able to verify that the contents studied in high school have more applications in everyday life than they realize.

Keywords: Error Correcting Linear Codes. Mathematics Teaching. Contextualization. Matrices and Linear Systems.

SUMÁRIO

INTRODUÇÃO

Códigos capazes de detectar e corrigir erros estão presentes, sem que percebamos, em diversas situações no cotidiano: quando ouvimos um CD, assistimos DVD ou televisão, utilizamos telefones celulares ou rádio, bem como quando há transferência e recebimento de informações via satélite ou por meio de computadores e internet.

Os códigos corretores de erros tem a função de detectar e corrigir erros que surjam na transmissão ou armazenamento de dados. Como as informações, ao serem transmitidas, estão sujeitas a "ruídos" – erros de digitação, interferências eletromagnéticas, dentre outros –, devem ser acrescentados dados adicionais ao codificar a informação que será transmitida ou armazenada – chamadas de redundâncias – e, por meio dessa informação extra, os dados podem ser recuperados.

Embora estejam tão presentes no dia-a-dia, a grande maioria dos alunos termina o Ensino Médio sem ao menos ter ouvido falar do assunto. Entretanto, os educandos estudam conteúdos que são ferramentas para construção de códigos como, por exemplo, a representação numérica em outras bases (mudanças de base), os vetores, as matrizes e os sistemas lineares. Conteúdos esses que, muitas vezes, são apresentados na forma de procedimentos mecânicos para resolução problemas sem qualquer contexto prático. Portanto, considerando a possibilidade de contextualizar os assuntos citados por meio do estudo dos códigos lineares corretores de erros e, ainda, de despertar nos alunos o interesse ao aprofundamento dos estudos no campo científico e tecnológico, é que será apresentada a sequência didática de atividades desta dissertação.

A referida sequência é composta de situações que envolvem a codificação, a transmissão, o recebimento e decodificação de mensagens, analisando-se em que situações é possível a correção delas, bem como o uso de planilhas eletrônicas como auxílio à resolução dos problemas. Ela deve ser aplicada, preferencialmente, a alunos que estejam cursando o segundo ou o terceiro ano do Ensino Médio, haja vista a necessidade de conhecimento dos conceitos relativos a matrizes e a sistemas lineares.

Para que haja uma boa compreensão do que é e de como pode ser aplicado um código corretor de erros, a dissertação será estruturada em seis capítulos, da seguinte forma.

No primeiro capítulo é apresentado o contexto histórico que motivou o início da pesquisa sobre os códigos corretores.

No segundo capítulo é exposta a fundamentação teórica sobre a teoria dos códigos, com os principais conceitos necessários à construção de um código corretor. São apresentados, ainda, diversos exemplos que possibilitam uma verificação numérica dos teoremas e dos mencionados conceitos.

No terceiro capítulo são definidos os códigos corretores lineares, bem como são apresentados teoremas importantes para a construção dos referidos códigos. Ressalta-se que, por meio dos mecanismos apresentados nesse capítulo, será possível a elaboração de códigos corretores mais eficazes.

No quarto capítulo é apresentada a fundamentação pedagógica, por meio da qual é ressaltado que o uso da contextualização histórica, de situações-problema e de computadores em sala de aula são métodos eficazes para a construção do conhecimento. A perspectiva adotada é a abordagem construtivista de ensino-aprendizagem.

No quinto capítulo são apresentadas as propostas de atividades. Inicialmente, é construído um código no qual não há qualquer preocupação com a possibilidade de correção de erros, caso eles ocorram (atividade 1). Em seguida, são analisados alguns códigos que são capazes de indicar – não em todas as situações – que ocorreram erros, mas que não são capazes de corrigi-los (atividade 2). Após, é construído um código capaz de corrigir erros. Todavia, não se sabe a quantidade de erros que ocorrem na transmissão ou armazenamento das informações e, por isso, a correção fica prejudicada. Por último, é construído um código em que é considerado que quantidade de erros que ocorrem na transmissão ou armazenamento é conhecida e, portanto, é possível estabelecer uma situação ideal, em que qualquer palavra pode ser corrigida, o que acontece quando o código linear é perfeito.

No sexto e último capítulo são apresentadas as considerações finais.

1 BREVE HISTÓRICO SOBRE O SURGIMENTO DOS CÓDIGOS CORRETORES DE ERROS

Na década de 1940[1], quando os computadores não eram tão populares e apenas grandes instituições tinham condições de possuí-los e mantê-los, Richard W. Hamming, que trabalhava para o laboratório Bell de Tecnologia, decidiu realizar pesquisas com o intuito de que as máquinas, que na época apenas detectavam erros, também fossem capazes de corrigi-los. A motivação para o estudo foi o fato de ele ter perdido várias semanas sem que seus arquivos fossem descarregados, pois, ao detectar um erro, os computadores interrompiam a leitura dos cartões perfurados e faziam a leitura dos dados do próximo usuário. Assim, como o acesso às máquinas era apenas nos fins de semana, quando chegava ao trabalho, no início da semana seguinte, Hamming verificava que os dados não haviam sido descarregados. Então, após esse problema ter ocorrido alguma vezes, ele decidiu que deveria encontrar uma solução.

Hamming começou sua pesquisa e a cada avanço publicava memorandos internos no Laboratório Bell. Em 1950 ele publicou seu trabalho no "The Bell System Thechnical Journal", no qual ele expôs um código capaz de detectar até dois erros e corrigir apenas um erro, se ele fosse único.

Mas não foi apenas Hamming que se interessou pelos códigos corretores de erros. Claude E. Shannon[2] e Marcel J. E. Golay também se debruçaram sobre a pesquisa dos códigos. Este último, inclusive, desenvolveu um código que foi utilizado para transmitir fotografias coloridas de Jupiter e Saturno pela espaçonave Voyager[3].

As ideias desenvolvidas por Hamming, Shannon e Golay são utilizadas ainda hoje nos celulares, gravadores de CD e DVD, comunicações via satélite, processamento de imagens digitais, internet e rádio, entre outras. Por isso, os códigos corretores de erros são objeto de estudo de várias áreas como matemática, computação, engenharia elétrica, estatística etc.

Atualmente os códigos corretores também são muito utilizados pela National Aeronautics and Space Administration (NASA) e pelo Jet Propulsion Laboratory (JPL). Por

[1] As informações descritas neste capítulo referentes ao início do estudo dos códigos corretores de erros foram obtidas, principalmente, por meio da referência bibliográfica [16].
[2] Shannon publicou um artigo intitulado "A Mathematical Theory of Communication" em outubro de 1948. Sua publicação é anterior ao artigo de Hamming, pois houve um pedido de patente dos códigos pelo laboratório Bell, o que ocasionou um atraso na publicação de Hamming.
[3] As sondas espaciais Voyager 1 e Voyager 2 enviam dados via rádio para a Terra a fim de que os cientistas possam analisa-los. Uma reportagem interessante sobre isso está disponível em http://ultimo segundo.ig.com.br/ciencia/2013-09-12/nasa-afirma-que-sonda-voyager-1-finalmente-saiu-do-sistema-solar.html

exemplo, nas missões Galileo[4], Cassini[5] e Marte foram utilizados códigos corretores que combinam técnicas dos códigos de Reed-Solomon com códigos convolucionais.

[4] Sonda enviada à Júpiter. Para mais informações sobre o assunto, pode ser verificado o registro http://noticias.terra.com.br/ciencia/interna/0,,OI145598-EI301,00.html.

[5] Sonda enviada à Saturno. Mais informações podem ser encontradas em http://www.ccvalg.pt/astronomia/noticias/2014/07/1_cassini_saturno.htm.

2 CONCEITOS IMPORTANTES

2.1 Definição de Código Corretor de Erros

Um bom modelo de um código corretor de erros presente no cotidiano é um idioma. Procederemos a uma caracterização do mencionado código. Inicialmente, vamos tomar um "alfabeto" $\mathbb{F}$ formado pelas 26 letras do nosso alfabeto, pelo espaço em branco (considerado como uma letra), pelo c cedilha e pelas vogais acentuadas. A seguir, procedemos à identificação da maior palavra do "alfabeto". Na língua portuguesa ela é, supostamente, inconstitucionalissimamente, que possui 27 letras. A partir daí, afirmaremos que cada palavra da língua portuguesa pode ser escrita como um elemento de $\mathbb{F}^{27}$, o espaço das 27-uplas com entradas em $\mathbb{F}$, bastando colocar espaços em branco à direita de cada palavra, até atingir 27 "letras", e omitindo esses caracteres em branco na escrita. A partir dessa construção, podemos tomar alguns exemplos de como ocorre a detecção e a correção de erros neste código.

Tomemos uma exemplificação. Ao ser visualizada a palavra "cathorro" é possível perceber que houve um erro, pois a referida palavra não pertence ao conjunto das palavras da língua portuguesa. Além disso, é admissível a correção por "cachorro", haja vista que esta é a palavra da língua portuguesa que mais se assemelha a "cathorro".

Podemos imaginar, em vista do exemplo analisado, que a língua portuguesa é um excelente código corretor de erros. Entretanto, vamos verificar, a seguir, que isso não é verdade. Para tanto, imaginemos que uma palavra foi grafada como "aato". Neste caso, não seria possível identificar a palavra correta, pois "bato", "tato", "gato", "mato" e "rato" também são palavras próximas àquela que foi erroneamente grafada. Devido a isso, podemos dizer que o código não é muito eficiente, pois existem diversas palavras muito próximas umas das outras.

Apesar de a língua portuguesa não ser um bom código corretor de erros, conforme ilustrado, esse modelo de codificação é muito útil, pois mostra, simplificadamente, como funciona um código. A partir dele podemos refletir sobre que características um bom código deve possuir para ser capaz de detectar e corrigir erros.

A reflexão elencada é justamente o objeto de estudo da chamada de Teoria dos Códigos. Por meio dela são pesquisadas as formas de obtenção de bons códigos corretores de erros, que são aqueles capazes de detectar e corrigir erros com certa precisão. Segundo Villela

e Hefez (2008, p. i), “um código corretor de erros é, em essência, um modo organizado de acrescentar algum dado adicional a cada informação que se queira transmitir ou armazenar, que permita, ao recuperar a informação, detectar e corrigir erros”.

Um exemplo interessante e que ilustra como funciona um código que consegue corrigir erros é o do código robô (Villela e Hefez, 2008, p. 2). Supõe-se que um robô esteja sobre um tabuleiro quadriculado e que seja capaz de se movimentar segundo quatro comandos apenas: Leste, Oeste, Norte, Sul. Assim, ao receber o comando, o robô se desloca do centro da casa que ocupa no tabuleiro para o centro da casa contígua indicada.

Para transmitir a informação ao robô, utiliza-se o sistema binário, criando-se um código de fonte segundo especificado a seguir:

Tabela 1

FONTE	CÓDIGO DA FONTE
Leste	00
Oeste	01
Norte	10
Sul	11

Fonte: Adaptação do exemplo apresentado por Villela e Hefez (2008, p.2).

Porém, ao efetuar a transmissão dos dados ao robô, pode ocorrer do código 00 ser recebido como 10, por exemplo, o que acarretaria na impossibilidade de identificação do erro cometido. É nesse momento que se verifica a existência da necessidade de acrescentar mais informações – chamadas de redundâncias. Assim, ao serem adicionadas as mencionadas redundâncias, é criado um código de canal, o qual permitirá a identificação e correção de erros.

Tabela 2

CÓDIGO DA FONTE	CÓDIGO DE CANAL
00	00000
01	01011
10	10110
11	11101

Fonte: Adaptação do exemplo apresentado por Villela e Hefez (2008, p.2)

Dessa maneira, após a nova codificação, se houver um erro de transmissão de alguma palavra, por exemplo, se 10110 for recebido como 11110, além de ser possível a verificação de que houve um erro na transmissão, pois essa palavra não está nas relacionadas pelo código de canal, também será possível a busca pela palavra correta. Para tanto, será procedida a verificação de qual palavra está mais próxima a que fora transmitida, ou seja, a palavra recebida é comparada com todas as palavras do código e é verificada qual delas possui a menor de quantidade de dígitos diferentes. No caso do exemplo anterior, obter-se-á a palavra 10110 – a quantidade de caracteres diferentes entre a palavra 11110 e as palavras 00000, 01011, 10110, 11101 é, respectivamente, quatro, três, um e dois.

Um esquema do que é efetuado para transmissão da mensagem está na figura a seguir:

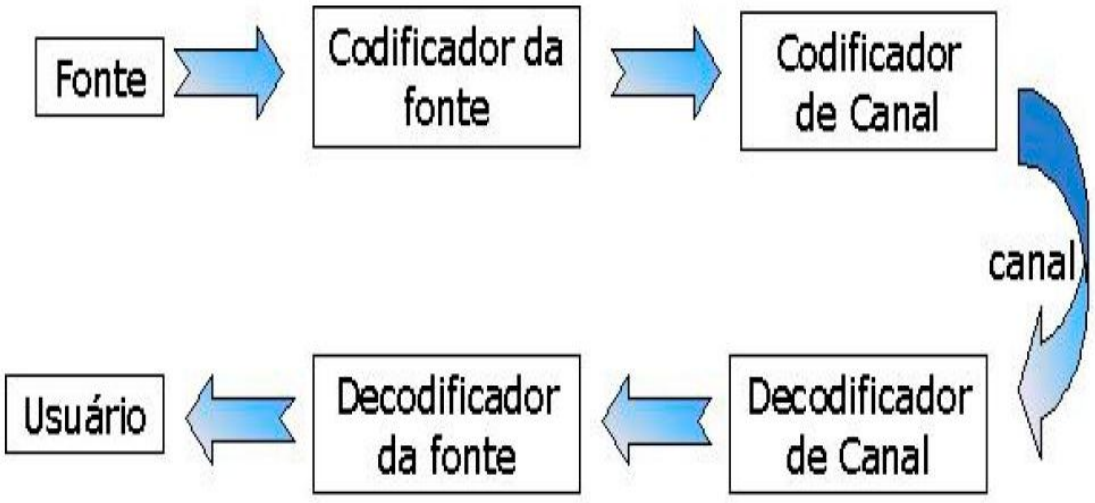


Figura 1
Fonte: Um Primeiro Curso sobre Códigos Corretores de Erros.
Disponível em http://www.ufsj.edu.br/portal2-repositorio/File/i-ermac/anais/minicursos/mc8.pdf.

Por meio do desenho, podemos observar que antes da transmissão da informação cada elemento do código de fonte é transformado em elemento do código de canal. A informação, então, é transmitida – podem ocorrer erros – e, ao ser recebida, o decodificador de canal identifica e corrige erros – quando é possível a identificação e correção. Em seguida, a palavra é transformada na linguagem inicial para que o usuário possa entendê-la.

É importante ressaltar que no exemplo apresentado anteriormente nada foi elencado sobre o aspecto probabilístico presente na teoria dos códigos. Da mesma forma procederemos em todos os demais exemplos que serão apresentados. Ou seja, estaremos considerando que, para ocorrer a transmissão e a recepção de mensagens com certo grau de certeza, devem ser

consideradas conhecidas a quantidade de erros e também não deve haver diferença probabilística de erro entre os símbolos transmitidos. Isto significa que todos os caracteres têm a mesma probabilidade de serem recebidos errados (por exemplo, a probabilidade de o elemento “0” ser recebido com erro é a mesma de o elemento “1” ser recebido com erro) e que a probabilidade de um símbolo transmitido com erro ser qualquer um dos demais caracteres do código é a mesma (por exemplo, se na palavra o caractere “0” foi transmitido com erro, a probabilidade de ele ser 1, 2, 3, ou qualquer outro é a mesma).

Procedendo-se em conformidade com os parâmetros acima, podemos citar outra situação em que é apresentado um bom código. Suponhamos que um indivíduo deseja enviar uma mensagem da forma (x,y,z), em que x, y, z $\in$ {0, 1}, e que de alguma forma foi identificado que o canal causa um erro em cada seis dígitos consecutivos. Dessa maneira, haverá erro em uma mensagem a cada duas enviadas. Sabendo disso, para tentar corrigir o erro, o emissor faz algumas tentativas. Na primeira hipótese, ele envia a mesma mensagem duas vezes. Por exemplo, ele envia (x, y, z) (x, y, z) e recebe (x, y, z) (y, y, z). Nesse caso, não há como o receptor saber se a mensagem correta é (x, y, z) ou (y, y, z) e ele apenas tem certeza de que dois dos dígitos estão corretos – o segundo e o terceiro, pois se repetiram. Como o envio de duas mensagens não permitiu a identificação da mensagem correta, o emissor decide, então, fazer outra tentativa. Desta vez ele faz o envio da mesma mensagem três vezes. Nessa situação, o receptor conseguirá identificar o erro e obterá a mensagem correta. Porém, essa última tentativa gerou um custo computacional muito grande – envio de nove caracteres para receber uma mensagem de três. Uma alternativa para enviar a mensagem com custo computacional menor é acrescentar três redundâncias da seguinte maneira: (x, y, z, x + y, x + z, y + z). Como x, y, z $\in$ {0, 1} devemos observar que a soma realizada é a soma módulo 2, ou seja, no sistema binário, onde 0 + 0 = 0, 0 + 1 = 1, 1 + 0 = 1 e 1 + 1 = 0.

Vamos entender porque essa última configuração funciona. Suponhamos que foi recebida a mensagem (**a**, **b**, **c**, **d**, **e**, **f**) na qual já se sabe que há apenas um erro – **a**, **b**, **c**, **d**, **e**, **f** pertencem ao sistema binário. Diante disso, o receptor calcula **a** + **b**. Se **a** + **b** = **d**, tem-se que **a**, **b**, **d** estão corretos e o erro está em **c**, **e** ou **f**. Como o **c** aparece três vezes (em **c**, em **e**, pois **e** = **a** + **c**, em **f**, pois **f** = **b** + **c**), comparando-se esses itens é possível verificar que **c** será o valor que ocorrer duas vezes, sendo possível, portanto, obter a mensagem correta. Por outro lado, se **a** + **b** $\neq$ **d**, um dos três (**a**, **b** ou **d**) está errado. Daí, **c** está correto, **e** = **a** + **c** está correto, e **f** = **b** + **c** está correto. Portanto, como os caracteres **c** e **e** estão corretos, obtêm-se **a**. De modo semelhante, como **f** e **c** estão corretos, obtêm-se **b**. Assim, o receptor consegue corrigir o erro da mensagem.

Agora que já determinamos o que é e entendemos como funciona um código corretor de erro, vamos definir os elementos de um código.

2.2 Principais Conceitos Relacionados aos Códigos

Nesta seção serão apresentados os conceitos básicos sobre os códigos corretores de erros.

Começaremos pelo alfabeto. Ele é o primeiro conjunto necessário para construir um código corretor de erros. No exemplo do código robô, o alfabeto escolhido foi o conjunto $\mathbb{F}_2 = \{0, 1\}$.[6]

Vamos definir este importante conjunto.

Definição 2.1. O **alfabeto** é um conjunto finito com **q** elementos ($\mathbb{F}_q$).

Passemos, então, à próxima etapa. Escolhido o alfabeto, vamos formar sequências de caracteres com os símbolos do mencionado conjunto, a fim de codificarmos os elementos da fonte. Em seguida, acrescentaremos as redundâncias. Ao final desse processo, obteremos as denominadas **palavras**. O número de letras das palavras é chamado de **comprimento**. Para facilitar na codificação e decodificação são convencionadas palavras de mesmo comprimento **n**. Por isso, esses códigos também são chamados de códigos em blocos. No exemplo do código robô, as palavras tinham comprimento **n** = 5.

Por meio do exemplificado, verificamos a importância da existência de um conjunto $\mathbb{F}_q$, a fim de que um elemento da fonte possa ser codificado em elemento de canal e, além disso, a necessidade de que seja estabelecido um comprimento **n** para as palavras de um código. Isso nos leva à próxima definição.

Definição 2.2. (Código) Um código C é um subconjunto de $\mathbb{F}_q^n$, em que

$$\mathbb{F}_q^n = \{(a_1, a_2, \dots, a_n) / a_i \in \mathbb{F}_q, 1 \leq i \leq n\}.$$

O código criado é chamado de **código q-ário** de comprimento **n**.

[6] O conjunto {0, 1} é chamado de Z_2. Ele é utilizado para representar os códigos criados no sistema binário de numeração.

No caso do código robô, o código escolhido foi o conjunto C = {00000, 01011, 10110, 11101} $\subset \mathbb{F}_2^5 = \mathbb{Z}_2^5$ – o conjunto $\mathbb{Z}_2^5$ possui 32 elementos obtidos pela permutação simples dos elementos 0 e 1 em cinco posições diferentes.

Vamos prosseguir com mais definições.

Definição 2.3. (Distância de Hamming) Dados dois elementos **x**, **y** $\in \mathbb{F}_q^n$, a distância de Hamming entre **x** e **y** é definida como o cômputo da quantidade de vezes em que os caracteres de mesma ordem diferem nas palavras. Ou seja, se **x** = $(x_1, x_2, \dots, x_n)$ e **y** = $(y_1, y_2, \dots, y_n)$, então

$$d(\mathbf{x}, \mathbf{y}) = |\{ i;\ x_i \neq y_i,\ 1 \leq i \leq n \}| .$$

Por exemplo, em $\{0, 1\}^4$, verificamos que:

$$d(0001, 1111) = 3;$$
$$d(0000, 1111) = 4;$$
$$d(1001, 1101) = 1.$$

Definição 2.4. (Métrica) Uma função d: A x A $\rightarrow \mathbb{R}$ é uma métrica se satisfaz as seguintes condições:

i) d (**x**, **y**) ≥ 0, $\forall$ **x**, **y** $\in$ A, sendo válida a igualdade se, e somente se, **x** = **y**;

ii) d (**x**, **y**) = d (**y**, **x**), $\forall$ **x**, **y** $\in$ A (propriedade simétrica);

iii) d (**x**, **y**) $\leq$ d (**x**, **p**) + d (**p**, **y**), $\forall$ **x**, **y, p** $\in$ A (desigualdade triangular).

Teorema 2.1. A distância de Hamming é uma métrica.

É imediata a constatação de que a distância de Hamming satisfaz as propriedades (i) e (ii). Vamos demonstrar que ela também satisfaz o item (iii).

Demonstração: A contribuição das i-ésimas coordenadas de **x** e **y** para d (**x**, **y**) é igual a zero se $x_i = y_i$, e igual a 1 se $x_i \neq y_i$. Assim, temos duas situações para verificar.

A primeira é aquela em que a contribuição é zero. Nesse caso, é certo que a contribuição das i-ésimas coordenadas a d (**x**, **y**) é menor ou igual a das i-ésimas coordenadas a d (**x**, **p**) + d (**p**, **y**), que podem ser 0, 1 ou 2 ($0 \leq 0$, $0 \leq 1$ ou $0 \leq 2$).

A segunda hipótese é aquela em que a contribuição é 1. Nessa situação, temos $x_i \neq y_i$ e, então, não podemos ter $x_i = p_i$ e $p_{i=}y_i$. Se esse último resultado acontecesse, teríamos $x_i = y_i$ e a contribuição seria zero, o que contraria nossa hipótese. A consequência do exposto é que a contribuição das i-ésimas coordenadas de d (**x**, **p**) + d (**p**, **y**) é maior ou igual a 1, que é o valor da contribuição das i-ésimas coordenadas de d (**x**, **y**). □

Definição 2.5. (Disco e esfera de centro em v e raio t) Seja **v** um elemento de $\mathbb{F}_q^n$ e **t** ≥ 0 um número inteiro.

Define-se o **disco** de centro **v** e raio **t** como sendo o conjunto D (**v**, **t**) = {**u** ∈ $\mathbb{F}_q^n$/d (**u**, **v**) ≤ **t**}, ou seja, o conjunto dos elementos **u** de $\mathbb{F}_q^n$ cuja distância até **v** seja menor do que ou igual ao número inteiro **t**.

Define-se a **esfera** de centro **v** e raio **t** como sendo o conjunto S (**v**, **t**) = {**u** ϵ $\mathbb{F}_q^n$ /d (**u**, **v**) = **t**}, ou seja, o conjunto dos elementos **u** de $\mathbb{F}_q^n$ cuja distância até **v** é exatamente igual ao número inteiro **t**.

Lema 2.1. O número de elementos do conjunto D (**v**, **t**) = {**u** ϵ $\mathbb{F}_q^n$ / d (**u**, **v**) ≤ **t**} é

$$| D (v, t)| = \sum_{i=0}^{t} \binom{n}{i} (q-1)^i.$$

O número de elementos do conjunto S (**v**, **t**) = {**u** ϵ $\mathbb{F}_q^n$ / d (**u**, **v**) = **t**} é

$$| S (v, t)| = \binom{n}{t} (q-1)^t.$$

Demonstração: Sabemos que ao escolhermos dois pontos **x** e **y** podemos afirmar que **y** está a distância **t** de **x** se, e somente se, **x** for diferente de **y** em exatamente **t** posições (definição 2.3). Então, ao analisarmos a quantidade de elementos do conjunto S (v, t), podemos dizer que para efetuar a contagem devemos selecionar **t** posições fixas para serem alteradas – entre as que compõem **v** –, a fim de que sejam obtidas as palavras cuja distância até **v** seja **t**. Como em cada uma das posições podemos ter **q – 1** caracteres do alfabeto diferentes, existem $(q-1)^t$ palavras de $\mathbb{F}_q^n$ que são diferentes de **v** nas **t** posições fixadas. Além disso, temos $\binom{n}{t}$ maneiras distintas para escolher **t** posições dentre **n** posições possíveis no elemento **v**. Portanto, existem exatamente $\binom{n}{t}(q-1)^t$ pontos na esfera S (**v**, **t**) (este resultado nos permite afirmar que todas as esferas de raio **t** possuem o mesmo número de pontos).

Como as distâncias são sempre números naturais, dentro de um mesmo disco[7] de centro **v** e raio **t** estão contidas todas as esferas do mesmo centro e de raios que são naturais menores ou iguais a **t**.

Logo:

$$D(v,t) = \bigcup_{t=0}^{t} S(v,t)$$

Ante ao exposto, podemos dizer que o número de pontos no disco de centro **v** e raio **t** é dado pela seguinte expressão:

$$|\,D\,(v,t)| = \sum_{i=0}^{t} \binom{n}{i} (q-1)^i$$

□

Tomemos um exemplo. Suponhamos que um conjunto $\mathbb{F}_2 = \{0, 1\}$ foi utilizado como alfabeto para obtenção de um conjunto $\mathbb{F}_2^3$ = {(0,0,0), (0,0,1), (1,0,0), (1,0,1), (1,1,0), (1,1,1), (0,1,1), (0,1,0)}. Em seguida, escolhemos a palavra **v** = (0,0,0) e o número inteiro **t** = 1. Como **q** = 2 (número de letras do alfabeto) e **n** = 3 (comprimento das palavras), o número de elementos de | D (**v**, **t**) | é:

$$|D\,(v,1)| = \sum_{i=0}^{1} \binom{3}{i} (2-1)^i = \binom{3}{0}(2-1)^0 + \binom{3}{1}(2-1)^1 = 1 + 3 = 4,$$

resultado que pode ser verificado por meio do conjunto D ((0,0,0), 1) = {(0,0,0), (0,0,1), (1,0,0), (0,1,0)}, que possui quatro elementos.

Tomando-se a palavra **v** = (0,0,0), novamente, e modificando-se o número inteiro para **t** = 2, obtêm-se o conjunto D ((0, 0, 0), 2) = {(0,0,0), (1,0,0), (0,1,0), (0,0,1), (1,0,1), (1,1,0), (0,1,1)}, que possui sete elementos. O referido valor também é confirmado por meio do cálculo:

$$|D\,(v,2)| = \sum_{i=0}^{2} \binom{3}{i} (2-1)^i = \binom{3}{0}(2-1)^0 + \binom{3}{1}(2-1)^1 + \binom{3}{2}(2-1)^2$$
$$= 1 + 3 + 3 = 7.$$

[7] Milies chama os discos de “bolas” em [15] e [16].

Definição 2.6 (Distância mínima) A distância mínima de um código C é o número **d** = min {d(**x**, **y**); **x**, **y** ∈ C e **x** ≠ **y**}.

Por exemplo, no código robô temos que **d** = 3, pois:

d(00000, 01011) = 3;
d(00000, 10110) = 3;
d(00000, 11101) = 4;
d(01011, 10110) = 4;
d(01011, 11101) = 3;
d(11101, 10110) = 3.

Para calcular **d** deve ser avaliada a distância entre as palavras uma quantidade de vezes igual a $\binom{M}{2}$, onde M é o número de palavras do código (combinação entre todas as palavras, duas a duas). Todavia, existe uma forma mais prática, com custo computacional melhor, que será apresentada posteriormente[8].

Definição 2.7. (Parâmetro k) Em um código C com distância mínima **d** define-se **κ** = $\left[\frac{d-1}{2}\right]$, onde [x] representa a parte inteira do número x.

Teorema 2.2. Utilizando-se o parâmetro **κ**, é possível constatar que se **x** e **y** são palavras distintas de um código C, então D(**x**, **κ**) ∩ D(**y**, **κ**) = ∅.

Demonstração: De fato, se uma palavra **p** pertencesse a D (**x**, **κ**) ∩ D (**y**, **κ**), então teríamos d (**p**, **x**) ≤ **κ** e d (**p**, **y**) ≤ **κ** e, portanto, d (**x**, **y**) ≤ d (**x**, **p**) + d (**p**, **y**) ≤ 2**κ** ≤ **d – 1**, o que é um absurdo já que d (**x**, **y**) ≥ **d**, haja vista que **x** e **y** são palavras do código e **d** é a distância mínima . □

Diante do que foi exposto nas definições e considerações anteriores, podemos apresentar, agora, o resultado principal da seção, que é o teorema que trata da quantidade de erros que podem ser detectados e corrigidos em um código.

[8] Voltaremos a esse assunto no capítulo 3, quando será definido o peso de um código.

Teorema 2.3 Seja C um código com distância mínima **d**. Então C pode corrigir até **k** = $\left[\frac{d-1}{2}\right]$ erros e detectar até **d – 1** erros.

Demonstração: Se na transmissão de uma palavra **x** do código ocorrem **t** erros, com **t** ≤ **κ**, será recebida uma palavra **y** com d (**x**, **y**) = **t** ≤ **κ**. Daí, **y** ∈ D(**x**, **κ**) e, pelo exposto no teorema 2.2, **y** ∉ D(**x'**, **κ**) para **x** ≠ **x'**, podendo-se concluir que d (**x**, **y**) é menor do que a distância de **y** a qualquer outra palavra do código e, dessa forma, obtêm-se **x** a partir de **y**. Quanto à quantidade de erros, como a distância mínima entre duas palavras do código C é **d**, tem-se que se a palavra **x** é recebida como **y**, com até **d – 1** erros, pode-se verificar que **y** ∉ C, pois d (**x**, **y**) < **d**. □

Por exemplo, no código robô verificamos que **d** = 3 (exemplo apresentado na definição 2.6). Dessa forma, pelo teorema acima, podemos dizer que o mencionado código corrige **κ** = $\left[\frac{3-1}{2}\right]$ = 1 erro e detecta até 3 – 1 = 2 erros.

2.3 O Problema Principal da Teoria dos Códigos

O principal problema da teoria dos códigos é estudar a relação entre os parâmetros **n** (comprimento das palavras do código), **M** (quantidade de palavras) e **d** (distância mínima). Os códigos que mais interessam são aqueles em que **M** e **d** sejam grandes relativamente a **n**, ou seja, códigos capazes de transmitir bastante informação (**M** é relativamente grande) e boa capacidade de correção (**d** é relativamente grande). Infelizmente, achar uma boa correlação entre **n**, **M** e **d** é algo complicado, pois quando se aumenta o número de palavras de um código, naturalmente a distância mínima entre elas diminui. Assim, é em busca da melhor forma de correlacionar os três parâmetros que faremos as considerações a seguir.

Inicialmente, fixemos **n** e analisemos a relação que deve existir entre **M** e **d**. Conforme demonstramos por meio do lema 2.1, se **x** ∈ $\mathbb{F}_q^n$, então

$$|D(x,t)| = \sum_{i=0}^{t} \binom{n}{i} (q-1)^i.$$

Além disso, de acordo com o teorema 2.2, todos os discos com centro em pontos diferentes do código C e raio **κ** tem interseção vazia. Assim, verificamos que

$$\bigcup_{x \in C} D(x, \kappa) \subset \mathbb{F}_q^n$$

e, então,

$$\sum_{x \in C} |D(x, \kappa)| \leq q^n.$$

Portanto, se considerarmos que o código possui **M** palavras, podemos dizer que

$$M \left[\sum_{t=0}^{\kappa} \binom{n}{t} (q-1)^t \right] \leq q^n.$$

Ante ao apresentado, obtermos uma limitação para o número de palavras de um código, considerando conhecidos o seu comprimento e a distância mínima.

O resultado acima é conhecido como Cota de Hamming:

$$M \leq \frac{q^n}{\sum_{t=0}^{\kappa} \binom{n}{t} (q-1)^t}$$

Definição 2.8. (Código Perfeito) Um código C $\subset \mathbb{F}_q^n$ com distância mínima **d** é dito um *código perfeito* se

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

onde k = $\left[\frac{d-1}{2}\right]$. Ou seja, se o código C tem **M** palavras, comprimento **n**, e distância mínima **d**, então ele é um código perfeito se, e somente se

$$M\left[\sum_{t=0}^{k} \binom{n}{t} (q-1)^t\right] = q^n$$

A condição citada é chamada de condição de empacotamento[9].

A importância de o código ser perfeito é manifesta na seguinte situação. Suponhamos que uma mensagem seja recebida e que seja constatada a ocorrência de erro na transmissão. Procede-se, então, à verificação de qual disco contém a mensagem e, dessa forma, é possível efetuar a correção do erro, assumindo que a mensagem correta é aquela mais próxima da

[9] Vide referência bibliográfica [16].

recebida, ou seja, o centro do disco. O problema é que algumas vezes ocorre de a mensagem não pertencer a nenhum dos discos, o que impossibilita a sua correção. Todavia, em se tratando de códigos perfeitos, isso não acontece.

Por exemplo, o código robô não é um código perfeito. Vejamos porquê. O código robô foi definido como o conjunto C = {00000, 01011, 10110, 11101} $\subset \mathbb{F}_2^5 = \mathbb{Z}_2^5$. Vamos escrever os conjuntos D (00000, 1), D (01011, 1), D (10110, 1) e D (11101, 1). Temos que:

D (00000, 1) = {00000, 10000, 01000, 00100, 00010, 00001}
D (01011, 1) = {01011, 11011, 00011, 01111, 01001, 01010}
D (10110, 1) = {10110, 00110, 11110, 10010, 10100, 10111}
D (11101, 1) = {11101, 01101, 10101, 11001, 11111, 11100}

Fazendo a união dos conjuntos D (00000, 1), D (01011, 1), D (10110, 1) e D (11101, 1) não obtemos o conjunto $\mathbb{Z}_2^5$. Ou seja, esse código não satisfaz a condição que caracteriza os códigos perfeitos.

2.4 Uma Estratégia para Detectar e Corrigir Erros

O teorema 2.3 permite traçar uma estratégia para detecção e correção de erros. Seja C um código com distância mínima **d**, que corrige até **k** = $\left[\frac{d-1}{2}\right]$ erros. Quando o receptor recebe uma palavra **r**, uma das seguintes situações é verificada:

a) a palavra **r** está em algum disco de raio **κ** em torno de uma palavra **c** do código (essa palavra é única, pela prova do teorema 2.2). Nesse caso, substitui-se **r** por **c**;
b) a palavra **r** não se encontra em nenhum disco de raio **κ** em torno de uma palavra **c** do código. Então, não é possível decodificar **r** com boa margem de segurança[10].

Observemos que em (a) não se pode ter certeza absoluta de que **c** tenha sido a palavra transmitida, pois poderíamos ter cometido mais do que **κ** erros, afastando assim **r** da palavra transmitida e aproximando-a de outra palavra do código. Observemos, também, que a hipótese (b) não ocorre em códigos perfeitos.

[10] A estratégia de correção transcrita foi proposta por Hefez e Villela (2008, p. 7). Devemos ressaltar que o item (b) somente é verdadeiro quando o código for linear.

Exemplificando a estratégia acima por meio do código robô, vamos realizar a verificação da possibilidade de decodificação das palavras 01011, 01101, 11011 e 10001.

Inicialmente, calcula-se o disco de raio 1 em torno de cada palavra do código:

D (00000, 1) = {(00000, 00001, 00010, 00100, 01000, 10000)}
D (01011, 1) = {(01011, 11011, 00011, 01111, 01001, 01010)}
D (10110, 1) = {(10110, 00110, 11110, 10010, 10100, 10111)}
D (11101, 1) = {(11101, 01101, 10101, 11001, 11111, 11100)}

Conforme já foi afirmado, o código robô não é perfeito, pois a união dos conjuntos anteriores não é o conjunto $\mathbb{F}_2^5$. Agora, procede-se à análise de cada palavra.

a) a palavra 01011 ∈ D(01011, 1) e, portanto, a palavra foi enviada sem erros;
b) a palavra 01101 ∈ D(11101, 1) e, portanto, a palavra correta é 11101;
c) a palavra 11011 ∈ D(01011, 1) e, portanto, a palavra correta é 01011;
d) a palavra 10001 não pertence aos conjuntos dos discos de raio 1 em torno de cada palavra do código e, então, não é possível decodificar essa palavra com segurança. Além disso, d(00000, 10001) = 2 e d(11101, 10001) = 2, o que confirma não ser possível estimar qual a palavra código foi transmitida.

É importante destacar que não há como ter certeza absoluta de que a decodificação acima está correta, pois podem ter sido cometidos mais de **κ** erros – no caso da questão, foi utilizado **κ** =1 nas primeiras três mensagens.

3 CÓDIGOS LINEARES

A classe de códigos mais utilizada é a dos lineares. Para entendê-la será necessária a observação prévia de alguns conceitos da Álgebra Linear, o que será feito a seguir.

3.1 Conceitos Auxiliares

Definição 3.1. (Corpo) Um conjunto K é chamado de corpo se ele for munido de duas operações e satisfizer as seguintes condições:

1) A adição é associativa: $(a + b) + c = a + (b + c)$, $\forall$ a, b, c $\in$ K;

2) A adição é comutativa: $a + b = b + a$, $\forall$ a, b $\in$ K;

3) A adição possui elemento neutro: $\exists$ 0 $\in$ K, tal que $a + 0 = a$, $\forall$ a $\in$ K;

4) A adição possui simétrico: $\forall$ a $\in$ K, $\exists$ $-a \in$ K tal que $a + (-a) = 0$;

5) A multiplicação é associativa: $(a \times b) \times c = a \times (b \times c)$, $\forall$ a, b, c $\in$ K;

6) A multiplicação é comutativa: $a \times b = b \times a$, $\forall$ a, b $\in$ K;

7) A multiplicação possui elemento neutro: $\exists$ 1 $\in$ K\{0}, tal que $a \times 1 = a$, $\forall$ a $\in$ K;

8) A multiplicação possui inversos: $\forall$ a $\in$ K\{0}, $\exists$ $a^{-1} \in$ K tal que $a \times a^{-1} = 1$;

9) A multiplicação é distributiva com relação à adição: $a \times (b + c) = a \times b + a \times c$, $\forall$ a, b, c $\in$ K.

Os conjuntos numéricos $\mathbb{Q}, \mathbb{R}$ *e* $\mathbb{C}$ satisfazem as condições acima e por isso são classificados como corpos.

Definição 3.2. (Espaço Vetorial) Um conjunto V é dito um espaço vetorial sobre um corpo K, se possui uma adição (+) com as seguintes propriedades:

1) A adição é associativa: $(u + v) + w = u + (v + w)$, $\forall$ u, v, w $\in$ V ;

2) A adição é comutativa: $u + v = v + u$, $\forall$ u, v $\in$ V;

3) A adição possui elemento neutro (elemento zero): $\exists$ 0 $\in$ V , tal que $v + 0 = v$, $\forall v \in$ V;

4) A adição possui simétricos: $\forall$ v $\in$ V , $\exists -$ v $\in$ V tal que $v + (-v) = 0$;

e existe, ainda, uma operação chamada de multiplicação por escalar, que associa a um elemento $a \in K$ e a um elemento $v \in V$, um elemento $av \in V$, tal que

5) $a(u + v) = au + av, \ \forall \ a \in K$ e $u, v \in V$;

6) $(a_1 + a_2)\, v = a_1\, v + a_2\, v, \ \forall \ a_1, a_2 \in K$ e $v \in V$;

7) $(a_1\, a_2)\, v = a_1\, (a_2\, v), \ \forall\, a_1, a_2 \in K$ e $v \in V$;

8) $1\, v = v, \ \forall\, v \in V$.

Os elementos do conjunto V são chamados de vetores e os elementos do corpo K de escalares.

Definição 3.3. (Subespaço Vetorial) Sejam V um espaço vetorial e W um subconjunto não vazio de V. Dizemos que W é um subespaço vetorial de V, ou simplesmente um subespaço de V, se W, com as operações de adição em V e de multiplicação por escalares, é um espaço vetorial.

Simplificando a verificação, W é um subespaço de V se, e somente se, as seguintes condições são satisfeitas:

(i) $0 \in W$

(ii) se $u, v \in W$, então $u + v \in W$;

(iii) se $a \in K$ e $u \in W$, então $a.u \in W$.

Por exemplo, o conjunto $W = \{(x, y);\ x + y = 0\}$ é um subespaço vetorial de $\mathbb{R}^2$, pois:

(i) $(0, 0) \in W$, se fizermos $x = y = 0$, temos $0 + 0 = 0$;

(ii) se (a, b) e $(c, d) \in W$, então $a + b = 0$ e $c + d = 0$. Daí, ao efetuar a soma $(a, b) + (c, d)$, obtemos o par $(a + c, b + d)$ no qual $a + c + b + d = a + b + c + d = 0 + 0 = 0$ e, portanto, $(a, b) + (c, d) \in W$;

(iii) se $k \in \mathbb{R}$ e $(a, b) \in W$, então $k\,(a, b) = (ka, kb) \in W$, pois $ka + kb = k\,(a + b) = k\,0 = 0$ (já que $a + b = 0$, pois $(a, b) \in W$).

Já o conjunto $W = \{(x, y);\ x + y = 1\}$ não é um subespaço vetorial de $\mathbb{R}^2$, pois $0 = (0, 0) \notin W$.

Verifiquemos, agora, uma maneira de descobrir o subespaço vetorial gerado por um conjunto de vetores.

Para isso, vamos apresentar, inicialmente, o que é uma combinação linear. Se V é um espaço vetorial e $v_1, v_2, \dots, v_r$ são vetores de V, então dizemos que um vetor v de V é a combinação linear de $v_1, v_2, \dots, v_r$ se existirem escalares $a_1, a_2, \dots, a_r$ pertencentes a K tais que

$$v = a_1 v_1 + a_2 v_2 + \dots + a_r v_r$$

Por exemplo, o vetor (1, 6, 0) $\in \mathbb{R}^3$ é combinação linear dos vetores (1, 2, 0) $\in \mathbb{R}^3$ e (-1, 2, 0) $\in \mathbb{R}^3$, pois (1, 6, 0) = 2 (1, 2, 0) + 1 (-1, 2, 0). Por outro lado, o vetor (2, -2, 6) $\in \mathbb{R}^3$ não é combinação linear dos vetores (1, 2, 0) e (-1, 2, 0), pois não é possível encontrar a_1 e a_2 reais tais que (2, -2, 6) = a_1 (1, 2, 0) + a_2 (-1, 2, 0). Isso ocorre porque o sistema de equações lineares

$$\begin{cases} a_1 - a_2 = 2 \\ 2.a_1 + 2.a_2 = -2 \\ 0.a_1 + 0.a_2 = 6 \end{cases}$$

é impossível.

De posse dos mesmos vetores do $\mathbb{R}^3$ do parágrafo anterior, (1, 2, 0) e $(-1, 2, 0)$, podemos identificar qual o conjunto gerado por eles. Para isso, basta encontrarmos um elemento genérico (x, y, z) $\in \mathbb{R}^3$ como combinação linear dos vetores (1, 2, 0) e $(-1, 2, 0)$. Então, se tomarmos os números $a, b \in \mathbb{R}$, podemos escrever a combinação dos vetores como

$$(x, y, z) = a\,(1, 2, 0) + b\,(-1, 2, 0).$$

O sistema de equações referente ao descrito acima é

$$\begin{cases} a - b = x \\ 2.a + 2.b = y\,, \\ 0.a + 0.b = z \end{cases}$$

o qual só possui solução se z = 0. Ou seja, os vetores (1, 2, 0) e $(-1, 2, 0)$ geram o plano z = 0.

Dessa forma, exemplificamos que ao considerarmos o conjunto de todas as combinações lineares dos vetores $v_1, v_2, \dots, v_r$ de V, obteremos um subespaço vetorial W de V, cujo conjunto gerador é $\{v_1, v_2, \dots, v_r\}$.

Todavia, em geral pode haver mais de uma forma de escrever um vetor de W como a combinação linear de vetores de um conjunto gerador. De fato, o $\mathbb{R}^3$, por exemplo, pode ser gerado pelos vetores (1,1,1), (1,1,0), (0,1,1) e (1,0,1) ou pelos vetores (1, 0, 0), (0, 1, 0) e

(0, 0, 1). Então, ao observamos o vetor (4, 2, 1) do $\mathbb{R}^3$, verificamos que ele pode ser escrito como:

$$(4,2,1) = 1\,(1,1,1) + 2\,(1,1,0) - 1\,(0,1,1) + 1\,(1,0,1) \text{ ou}$$
$$(4,2,1) = -1\,(1,1,1) + 2\,(1,1,0) + 0\,(0,1,1) + 2\,(1,0,1).$$

Ante o apresentado, a fim de que a maneira de obtenção de um vetor de um subespaço vetorial seja única, é necessário utilizar um conjunto gerador que tenha uma característica específica, ou seja, é preciso encontrar um conjunto linearmente independente.

Podemos dizer que um conjunto de vetores $v_1, v_2, \dots, v_r$ de um espaço Vetorial V é linearmente independente, se a equação

$$a_1v_1 + a_2v_2 + \dots + a_rv_r = 0\,,$$

é satisfeita se, e somente se, $a_1 = a_2 = \dots = a_r = 0$. Caso exista algum $a_i \neq 0$, $1 \leq i \leq n$, dizemos que os vetores são linearmente dependentes.

Os resultados acima nos levam à próxima definição.

Definição 3.4. (Base e Dimensão) Um conjunto $\{v_1, v_2, \dots, v_r\}$ de vetores de um espaço vetorial V é uma base de V se esse conjunto é linearmente independente e se ele gera o espaço V. Quando isso acontece, cada vetor de V pode ser escrito de modo único como a combinação linear dos vetores da base.

O número de elementos de uma base de um espaço vetorial V é chamado de *dimensão de V*. Se V for o espaço vetorial nulo, então sua dimensão é 0.

Por exemplo, o conjunto {(1,0,0), (0,1,0), (0,0,1)} é linearmente independente e gera o $\mathbb{R}^3$. Portanto, ele é uma base do $\mathbb{R}^3$. A dimensão desse espaço vetorial é 3.

Definição 3.5. (Transformação linear) As funções de interesse da Álgebra Linear são as funções cujos domínios e contradomínios são espaços vetoriais e que, além disso, preservam as operações de adição de vetores e de multiplicação de um vetor por um escalar.

Assim, tomando-se os espaços vetoriais V e W sobre um corpo K, dizemos que uma *transformação linear de V em W* é uma função $T : V \to W$ que possui as seguintes propriedades:

(i) $T(v_1 + v_2) = T(v_1) + T(v_2)$, para quaisquer v_1 e v_2 em V;

(ii) $T(av) = aT(v)$, para quaisquer v em V e a em K.

As propriedades (i) e (ii) são equivalentes à seguinte propriedade:

$T(v_1 + av_2) = T(v_1) + aT(v_2)$; para quaisquer v_1 e v_2 em V e para qualquer a em K.

Por exemplo, a função T: $\mathbb{R}^3 \rightarrow \mathbb{R}^2$ dada por T (x, y, z) = (x – y, y – z) é uma transformação linear, pois se tomarmos $v_1 = (x_1, y_1, z_1)$, $v_2 = (x_2, y_2, z_2) \in \mathbb{R}^3$ e a $\in \mathbb{R}$ teremos:

$$T(v_1 + av_2) = T(x_1 + ax_2, y_1 + ay_2, z_1 + az_2)$$
$$= (x_1 + ax_2 – (y_1 + ay_2), y_1 + ay_2 – (z_1 + az_2)$$
$$= ((x_1 – y_1) + a(x_2 – y_2), (y_1 – z_1) + a(y_2 – z_2))$$
$$= (x_1 – y_1, y_1 – z_1) + a(x_2 – y_2, y_2 – z_2)$$
$$= T(v_1) + a\ T\ (v_2)$$

Definição 3.6. (Núcleo e imagem de uma transformação linear) Seja T : V → W uma transformação linear. O núcleo de T é o conjunto de vetores de V que são levados por T no vetor nulo de W, ou seja, $\{v \in V; T(v) = 0\}$. O núcleo é um subespaço de V.

Se o núcleo de uma transformação linear for apenas o vetor nulo, a transformação é chamada de injetiva.

A imagem de T é o conjunto Im T = T(V) e também é um subespaço vetorial.

Por exemplo, se tomarmos T : $\mathbb{R}^4 \rightarrow \mathbb{R}^3$ definida por T(x, y, s, t) = (x – y + s + t, x + 2s – t, x + y + 3s – 3t), para calcular o núcleo devemos obter o conjunto de vetores (x, y, s, t) em $\mathbb{R}^4$ tais que T(x, y, s, t) = (x – y + s + t, x + 2s – t, x + y + 3s – 3t) = (0, 0, 0), ou seja, devemos obter a solução do sistema linear homogêneo

$$\begin{cases} x - y + s + t = 0 \\ x + 2s - t = 0 \\ x + y + 3s - 3t = 0 \end{cases}$$

Resolvendo o sistema obtemos como núcleo o conjunto $\{(–2s + t, –s + 2t, s, t); s, t \in R\}$, que é um subespaço vetorial do $\mathbb{R}^4$ de dimensão 2.

Para determinar o espaço gerado pela imagem, devemos tomar um conjunto de geradores do $\mathbb{R}^4$ (pode ser a base canônica) e obter a imagem desses geradores para, em seguida, após escalonar a matriz formada pelas imagens, tomar as linhas não nulas da matriz equivalente, as quais formam uma base para o espaço linha procurado. Portanto:

T(1,0,0,0) = (1, 1, 1), T(0,1,0,0) = (–1, 0, 1), T(0,0,1,0) = (1, 2, 3) e T(0,0,0,1) = (1, –1, –3)

$$\begin{bmatrix} 1 & 1 & 1 \\ -1 & 0 & 1 \\ 1 & 2 & 3 \\ 1 & -1 & -3 \end{bmatrix}$$

Escalonando-se:

$$\begin{bmatrix} 1 & 1 & 1 \\ -1 & 0 & 1 \\ 1 & 2 & 3 \\ 1 & -1 & -3 \end{bmatrix} \begin{matrix} L_2 \to L_2 + L_1 \\ L_3 \to L_3 - L_1 \\ L_4 \to L_4 - L_1 \end{matrix} \to \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 2 \\ 0 & 1 & 2 \\ 0 & -2 & -4 \end{bmatrix} \begin{matrix} L_3 \to L_3 - L_2 \\ L_4 \to L_4 + 2L_2 \end{matrix} \to \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 2 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Então, uma base da imagem é {(1, 1, 1), (0, 1, 2)}.

3.2 Entendendo o Funcionamento de um Código Linear

Tomemos o conjunto $\mathbb{F} = \mathbb{Z}_2 = \{0, 1\}$ como alfabeto, no qual a operação de soma é a soma módulo 2 – sistema binário –, a fim de construir um código de comprimento **n** = 6, em que as três primeiras componentes de cada uma das palavras sejam a informação que desejamos transmitir.

Se a palavra a ser transmitida é x, y, z, podemos definir os dígitos de redundância como t = x + y, w = x + z, v = y + z, por exemplo.

Adotando-se a notação vetorial, uma palavra **c** do código pode ser escrita como **c** = (x, y, z, x + y, x + z, y + z), que na notação matricial é:

$$(x, y, z) \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix} = (x, y, z, x + y, x + z, y + z)$$

Desse modo, quando (x, y, z) percorrer todos os elementos de $\mathbb{F}_2^3 = \mathbb{Z}_2^3$, serão produzidas todas as palavras do código. Por isso a matriz acima é chamada de matriz geradora do código.

O exemplo ilustrou o que será apresentado a seguir.

Definição 3.7. (Códigos Lineares) Tomemos um corpo finito $\mathbb{F}_q$ de **q** elementos para ser um alfabeto e, a partir dele, construamos um código $C \subset \mathbb{F}_q^n$. Esse código será chamado código linear se for um subespaço vetorial de $\mathbb{F}_q^n$.

Se a dimensão do referido subespaço for **m**, onde **m** < **n**, então o número de palavras do código será $M = q^m$.

Um código nas condições estabelecidas acima é chamado de $(\boldsymbol{n}, \boldsymbol{m})_q$-código linear sobre $\mathbb{F}$ e, caso também seja conhecida a distância mínima d, ele é chamado de $(\boldsymbol{n}, \boldsymbol{m}, \boldsymbol{d})_q$-código linear.

Note que os códigos lineares são subespaços vetoriais. Portanto, o vetor **0**, elemento neutro da soma, sempre estará em um código linear.

Definição 3.8. (Peso de um elemento) Define-se o *peso de Hamming* de um elemento **c** de um código linear como o número w (**c**) = d (**c**, **0**), onde **d** é a métrica de Hamming.

Definição 3.9. (Peso do código) O *peso de um código C* é o menor valor obtido dentre o peso de cada um dos elementos do código, ou seja,

$$w(C) := \min \{w(\mathbf{c}); \mathbf{c} \in C \setminus \{0\}\}$$

Notemos que se $\mathbf{x} = (a_1, a_2, ..., a_n)$ e $\mathbf{y} = (b_1, b_2, ..., b_n)$ são elementos de um código linear C, então $d(\mathbf{x}, \mathbf{y}) = |\{i / a_i \neq b_i, 1 \leq i \leq n\}| = |\{i / a_i - b_i \neq 0, 1 \leq i \leq n\}| = d(\mathbf{x} - \mathbf{y}, 0) = w(\mathbf{x} - \mathbf{y})$. Portanto, toda distância entre elementos do código C é também o peso de algum elemento, o que acarreta em

$$d = w(C).$$

Assim, como o peso do código é a distância mínima, não é mais necessário fazer a análise de $\binom{M}{2} = \frac{M(M-1)}{2}$ distâncias (conforme descrito na definição 2.6), mas apenas de **M – 1** (a distância de cada um dos M – 1 elementos não nulos ao elemento neutro). Apesar do avanço, se o número de elementos do código for muito grande, o cálculo de **d** por meio do peso das palavras do código também fica inviável, por representar um custo computacional muito elevado, sendo necessário o desenvolvimento de outros métodos para determinação da distância mínima.

Tomemos um exemplo. Seja C = {0000, 1011, 0110, 1101} ⊂ $\mathbb{Z}_2^4$ um código linear. Podemos observar que uma base para o código pode ser o conjunto {1011, 1101}, pois todos os elementos do código são combinação linear de 1011 e 1101:

$$0000 = 0\,(1011) + 0\,(1101)$$
$$1011 = 1\,(1011) + 0\,(1101)$$

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

Ou seja, cada vetor foi escrito como a_1 (1011) + a_2 (1101), onde $a_1, a_2 \in \mathbb{Z}_2$. Além disso, a_1 (1011) + a_2 (1101) = (0000) se, e somente se, $a_1 = a_2 = 0$ (os vetores são linearmente independentes).

Para obter a distância mínima do código C basta calcular m – 1 = 4 – 1 = 3 pesos: w (1011) = 3, w (0110) = 2 e w (1101) = 3. Portanto, a distância mínima de C é 2 e o código pode ser classificado como um $(4,2,2)_2$-código linear.

3.3 Formalizando o Código

Por meio da Álgebra Linear sabemos que é possível construir subespaços vetoriais C de $\mathbb{F}_q^n$ de duas maneiras: obtendo-se o núcleo ou a imagem de uma transformação linear. Inicialmente vamos utilizar a imagem.

3.3.1 Código obtido por meio da imagem de uma transformação linear

Suponhamos que será construído um código para enviar **m** dígitos de informação e **n – m** dígitos de redundância. Desse modo, o vetor de informação é um elemento do espaço vetorial $\mathbb{F}_q^m$, o vetor codificado é um elemento do $\mathbb{F}_q^n$ e o código C é um subespaço $C \subset \mathbb{F}_q^n$ de dimensão **m** e base $v_1, v_2, \ldots, v_m$.

Obteremos a representação do código C por meio da seguinte transformação linear:

$$\begin{array}{ccccc} T: & \mathbb{F}_q^m & \rightarrow & \mathbb{F}_q^n \\ & x = (x_1, x_2, \ldots, x_m) & & x_1v_1 + x_2v_2 + \ldots + x_mv_m \end{array}$$

Como a aplicação linear T é injetora (demonstração abaixo), então a imagem de T é o código C, ou seja, Im (T) = C.

Demonstração: Tomemos $x, y \in \mathbb{F}_q^m$ tais que T(x) = T(y). Portanto,

$$x_1v_1 + x_2v_2 + \ldots + x_mv_m = y_1v_1 + y_2v_2 + \ldots + y_mv_m$$
$$(x_1 - y_1)v_1 + (x_2 - y_2)v_2 + \ldots + (x_m - y_m)\, v_m = 0$$

e, como $\{v_1, v_2, \ldots, v_m\}$ é um conjunto linearmente independente (pois é uma base do código C) temos que $x_i - y_i = 0$ para todo $i \in \{1, 2, \ldots, m\}$, o que acarreta em $x = y$ e demonstra que a aplicação é injetora. □

Por exemplo, utilizemos um código $C \subset \mathbb{F}_2^7$ sobre o corpo $\mathbb{F}_2 = \{0, 1\}$ com base {(1,0,0,0,1,1,0), (0,1,0,0,0,1,1), (0,0,1,0,1,0,1), (0,0,0,1,1,1,1)} e uma palavra **x** = (1,0,1,1) do espaço vetorial $\mathbb{F}_2^4$. Logo, para codificar x basta fazermos:

$$T(x) = T(1,0,1,1) = 1.\ (1,0,0,0,1,1,0) + 0.\ (0,1,0,0,0,1,1) + 1.(0,0,1,0,1,0,1) + 1.(0,0,0,1,1,1,1)$$

$$T(x) = (1,0,1,1,1,0,0) = c.$$

Assim, a palavra foi codificada para o código e pode ser transmitida.

Façamos outro exemplo. Tomemos a transformação linear $T: \mathbb{F}_2^3 \to \mathbb{F}_2^5$ tal que $T(x_1, x_2, x_3) = (x_1, x_3, x_1 + x_2, x_2 + x_3, x_2)$. No caso, $T(x_1, x_2, x_3) = (x_1, x_3, x_1 + x_2, x_2 + x_3, x_2) = (0,0,0,0,0)$ se, e só se, $x_1 = x_2 = x_3 = 0$. Então, a aplicação é injetiva e sua imagem é o código C: Im (T) = C.

Escrevendo a imagem dos elementos da base canônica de $\mathbb{R}^3$ em termos dos elementos da base canônica de $\mathbb{R}^5$, temos que:

$$T(1,0,0) = (1,0,1,0,0) = 1(1,0,0,0,0) + 0(0,1,0,0,0) + 1(0,0,1,0,0) + 0(0,0,0,1,0) + 0(0,0,0,0,1)$$

$$T(0,1,0) = (0,0,1,1,1) = 0(1,0,0,0,0) + 0(0,1,0,0,0) + 1(0,0,1,0,0) + 1(0,0,0,1,0) + 1(0,0,0,0,1)$$

$$T(0,0,1) = (0,1,0,1,0) = 0(1,0,0,0,0) + 1(0,1,0,0,0) + 0(0,0,1,0,0) + 1(0,0,0,1,0) + 0(0,0,0,0,1)$$

e, então, a matriz de codificação do código é

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

Ante ao apresentado, é possível observar que há duas formas de codificar uma palavra. Por exemplo, se x = (1,0,1), pode ser calculado T(1,0,1) = (1, 1, 1, 1, 0) ou

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

E, em vista do referido resultado, segue-se a definição abaixo.

Definição 3.10. (Matriz geradora de um código) Para descrever a matriz geradora de um código $C \subset \mathbb{F}_q^n$ (transmite elementos de um espaço $\mathbb{F}_q^m$ com **n – m** dígitos de redundância), devemos tomar uma base $B = \{v_1, v_2, \ldots, v_m\}$ do código C, em que cada vetor é da forma $v_i = (v_{i1}, v_{i2}, \ldots, v_{in})$ e escrever a matriz na qual as linhas são os vetores da base de C. Essa matriz será chamada de matriz geradora do código C associada à base B.

$$G = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_m \end{bmatrix} = \begin{bmatrix} v_{11} & v_{12} & \cdots & v_{1n} \\ v_{21} & v_{22} & \cdots & v_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ v_{m1} & v_{m2} & \cdots & v_{mn} \end{bmatrix}$$

Como em cada uma das linhas da matriz há um vetor do código, o código C é um subespaço de $\mathbb{F}_q^n$ gerado pelas linhas da matriz G. Os elementos de C são vetores $\mathbf{y} \in \mathbb{F}_q^n$ da foma **x**.G = **y**, para todo $\mathbf{x} \in \mathbb{F}_q^m$. Isso é equivalente a escrever uma transformação linear

$$\begin{array}{ccccc} \text{T:} & \mathbb{F}_q^m & \rightarrow & \mathbb{F}_q^n \\ & \text{x} & \rightarrow & \text{xG} \end{array}$$

em que **y** = T (**x**) = **x**.G = $x_1v_1 + x_2v_2 + ... + x_mv_m$ e **x** = (x_1, x_2, ... , x_m). Logo, T ($\mathbb{F}_q^m$) = C. Daí, podemos afirmar que $\mathbb{F}_q^m$ é o código da fonte e C o código de canal, sendo a transformação T uma forma de codificação.

Façamos um exemplo. Tomemos uma transformação linear T: $\mathbb{F}_2^2 \rightarrow \mathbb{F}_2^5$ tal que T (x_1, x_2) = (x_1, x_2, x_1, $x_1 + x_2$, x_2), que transmite dois dígitos de informação e três dígitos de redundância, na qual o alfabeto é o corpo $\mathbb{F}_2 = \{0, 1\}$. Inicialmente, é realizada a verificação de que a aplicação linear é injetiva.

Note que T(x_1, x_2) = (x_1, x_2, x_1, $x_1 + x_2$, x_2) = (0,0,0,0,0) se, e só se, $x_1 = x_2 = 0$. Então, a aplicação é injetiva e a imagem dela é o código linear C.

Como os resultados de T(1,0) e T (0, 1) são, respectivamente, T(1,0) = (1, 0, 1, 1, 0) e T (0, 1) = (0, 1, 0, 1, 1), e os vetores (1, 0, 1, 1, 0) e (0, 1, 0, 1, 1) são linearmente independentes, podemos concluir que {(1, 0, 1, 1, 0), (0, 1, 0, 1, 1)} é uma base para a imagem e, consequentemente, para o código. Como a base possui dois vetores, então temos que sua dimensão **m** = 2. Além disso, verificamos que o código possui a seguinte quantidade de palavras: $M = q^m = 2^2 = 4$.

Obtida uma base para o código, podemos escrever a matriz geradora, que é

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

Vamos efetuar mais algumas verificações. Podemos escrever todas as palavras do código: T(0,0) = (0, 0, 0, 0, 0); T(1,0) = (1, 0, 1, 1, 0); T(0,1) = (0, 1, 0, 1, 1); T(1,1) = (1, 1, 1, 0, 1). Em seguida, verifiquemos o valor do peso de cada uma delas, com exceção do vetor nulo: w(1, 0, 1, 1, 0) = 3; w(0, 1, 0, 1, 1) = 3; w(1, 1, 1, 0, 1) = 4. Então, podemos concluir

que o peso do código é 3. Portanto, a distância mínima é **d** = 3. Donde podemos afirmar que o código é capaz de corrigir $\boldsymbol{\kappa} = \left[\frac{d-1}{2}\right] = 1$ erro e detectar $d - 1 = 3 - 1 = 2$ erros.

Façamos outro exemplo. Tomemos o código que é a imagem da transformação linear

$$\begin{array}{cccc} T: & \mathbb{F}_2^3 & \rightarrow & \mathbb{F}_2^5 \\ & x & \rightarrow & xG \end{array}$$

onde $\mathbb{F}_2 = \mathbb{Z}_2 = \{0, 1\}$ e G é a seguinte matriz geradora:

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

A palavra 101 do código pode ser codificada como 01010, pois

$$[1 \quad 0 \quad 1]\begin{bmatrix} 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} = [0 \quad 1 \quad 0 \quad 1 \quad 0].$$

Suponhamos agora que desejamos decodificar a palavra 10101. Para isso, devemos resolver o sistema $[x_1 \quad x_2 \quad x_3]G = [1 \quad 0 \quad 1 \quad 0 \quad 1]$, ou seja, resolver o sistema de equações abaixo:

$$\begin{cases} x_1 + x_2 + x_3 = 1 \\ x_2 + x_3 = 0 \\ x_1 + x_3 = 1 \\ x_2 + x_3 = 0 \\ x_1 + x_3 = 1 \end{cases}$$

cuja solução é $x_1 = 1$, $x_2 = 0$ e $x_3 = 0$.

Esse sistema foi de fácil resolução, mas se a matriz G for complexa, a resolução pode ficar muito difícil. Ou seja, o custo computacional da referida resolução pode ser elevado.

Por outro lado, é possível efetuar operações[11] com as linhas da matriz G, a fim de que ela assuma outra forma (G'):

$$\begin{bmatrix} 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{matrix} \overrightarrow{} \\ L_2 \rightarrow L_2 - L_1 \\ L_3 \rightarrow L_3 - L_1 \end{matrix} \begin{bmatrix} 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 \end{bmatrix} L_2 \overrightarrow{\leftrightarrow} L_3 \begin{bmatrix} 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 \end{bmatrix} L_3 \rightarrow \overrightarrow{L_3} - L_2$$

$$\begin{bmatrix} 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \end{bmatrix} L_1 \rightarrow \overrightarrow{L_1} - L_3 \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \end{bmatrix}$$

[11] São as mesmas operações utilizadas para obter sistemas de equações equivalentes. São elas:
L1 – permutação de duas linhas
L2 – multiplicação de uma linha por um escalar não nulo
L3 – adição de um múltiplo escalar de uma linha a outra.

Dessa forma, ao fazermos xG' obtemos (x_1 x_2 x_3 x_2 x_3) o que torna mais fácil a decodificação, pois o vetor é imediatamente identificado por meio das três primeiras componentes. Logo, a palavra 10101 é facilmente decodificada como 101.

O método utilizado nos leva aos resultado a seguir.

3.3.2 Matriz geradora na forma padrão

Dizemos que uma matriz de codificação G de um código C está na forma padrão se tivermos G = [Id_m | A], onde Id_m é a matriz identidade **m** x **m** e A uma matriz **m** x (**n – m**).

Verificamos que a matriz geradora do tópico anterior pode ser colocada na forma padrão. Todavia, nem sempre isso é possível com o uso das operações entre linhas. Por exemplo, se tomarmos um código em $\mathbb{F}_2^5$, $\mathbb{F}_2 = \{0, 1\}$, de matriz geradora

$$\begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix},$$

isso não será possível, pois as operações padrões entre linhas não permitem que tal aconteça. Entretanto, se forem efetuadas operações de permutações de colunas, é possível obter uma matriz na forma padrão

$$\begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix} \begin{matrix} c_1 \leftrightarrow c_3 \\ c_2 \leftrightarrow c_4 \end{matrix} \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{bmatrix}.$$

Portanto, se acrescentarmos às operações padrões entre linhas as operações entre colunas:

C1 – permutação de duas colunas, e

C2 – multiplicação de uma coluna por um escalar não nulo

também poderemos encontrar matrizes G' de um código C' equivalente a C, o que nos leva ao teorema a seguir.

Teorema 3.1. (Código equivalente) Dado um código C, existe um código equivalente C' com matriz geradora na forma padrão.

Demonstração: Seja G uma matriz geradora de um código C. Podemos colocar G na forma padrão com uma sequência das operações descritas no item 3.3.3 (L1, L2, L3, C1 e C2).

Tomemos a matriz

$$G = \begin{bmatrix} v_{11} & v_{12} & \cdots & v_{1n} \\ v_{21} & v_{22} & \cdots & v_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ v_{m1} & v_{m2} & \cdots & v_{mn} \end{bmatrix}$$

Como a primeira linha de G não é nula (os vetores linhas de G são linearmente independentes), por meio de C1 podemos supor de $v_{11} \neq 0$. Em seguida, multiplicamos a primeira linha por v_{11}^{-1} e, então, podemos colocar 1 no lugar de v_{11} (nesse caso foi utilizada a operação L2).

Somando-se à i-ésima linha a primeira linha por (–1) v_{i1}, obteremos a matriz abaixo (nesse caso foi utilizada uma sequência de operações do tipo L3):

$$\begin{bmatrix} 1 & a_{12} & \cdots & a_{1n} \\ 0 & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & a_{m2} & \cdots & a_{mn} \end{bmatrix}$$

Agora, considerando que na segunda linha da matriz há algum elemento não nulo e que por meio da operação C1 podemos colocá-lo na segunda linha e segunda coluna, vamos proceder de forma semelhante ao que foi realizado no parágrafo anterior: multiplicamos a segunda linha pelo inverso do elemento, transformando a matriz, conforme apresentado abaixo.

$$\begin{bmatrix} 1 & b_{12} & \cdots & b_{1n} \\ 0 & 1 & \cdots & b_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & b_{m2} & \cdots & b_{mn} \end{bmatrix}$$

Assim, por meio de operações do tipo L3, de forma análoga ao que já foi realizado anteriormente, obtemos a matriz

$$\begin{bmatrix} 1 & 0 & c_{13} & \cdots & c_{1n} \\ 0 & 1 & c_{23} & \cdots & c_{2n} \\ 0 & 0 & c_{33} & \cdots & c_{3n} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & c_{m3} & \cdots & c_{mn} \end{bmatrix}$$

Fazendo-se de forma sucessiva o procedimento acima, coluna por coluna, em determinado momento encontraremos uma matriz na forma padrão

$$G' = [Id_m \mid A]. \square$$

Façamos um exemplo. Tomemos um código C definido sobre $\mathbb{F}_2 = \{0, 1\}$ pela matriz G abaixo. Podemos obter uma matriz G' na forma padrão, por meio dos seguintes procedimentos:

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

$$\begin{bmatrix}1&1&1&0&0&0&0\\1&0&0&1&1&0&0\\1&0&0&0&0&1&1\\0&1&0&1&0&1&0\end{bmatrix}\begin{matrix}\\L_2\to L_2-L_1\\L_3\to L_3-L_1\\\\\end{matrix}\begin{bmatrix}1&1&1&0&0&0&0\\0&1&1&1&1&0&0\\0&1&1&0&0&1&1\\0&1&0&1&0&1&0\end{bmatrix}\begin{matrix}\\\\L_3\to L_3-L_2\\L_4\to L_4-L_2\end{matrix}$$

$$\begin{bmatrix}1&1&1&0&0&0&0\\0&1&1&1&1&0&0\\0&0&0&1&1&1&1\\0&0&1&0&1&1&0\end{bmatrix}\quad c_3\leftrightarrow c_4\quad\begin{bmatrix}1&1&0&1&0&0&0\\0&1&1&1&1&0&0\\0&0&1&0&1&1&1\\0&0&0&1&1&1&0\end{bmatrix}L_1\to L_1-L_2$$

$$\begin{bmatrix}1&0&1&0&1&0&0\\0&1&1&1&1&0&0\\0&0&1&0&1&1&1\\0&0&0&1&1&1&0\end{bmatrix}\begin{matrix}L_1\to L_1-L_3\\L_2\to L_2-L_3\\\\\end{matrix}\begin{bmatrix}1&0&0&0&0&1&1\\0&1&0&1&0&1&1\\0&0&1&0&1&1&1\\0&0&0&1&1&1&0\end{bmatrix}L_2\to L_2-L_4$$

$$G' = \begin{bmatrix}1&0&0&0&0&1&1\\0&1&0&0&1&0&1\\0&0&1&0&1&1&1\\0&0&0&1&1&1&0\end{bmatrix} = [\mathrm{Id}_4 \mid \mathrm{A}_{4x3}].$$

3.3.3 Código dual

Antes de iniciar o estudo dos códigos duais, é importante sabermos a definição de produto interno de vetores.

Definição 3.11. (Produto interno) Sejam **u** = (u_1, u_2, ... , u_n) e **v** = (v_1, v_2, ... , v_n) elementos de $\mathbb{F}^n$. Definimos o produto interno de **u** e **v** como

$$< \mathbf{u}, \mathbf{v} > = u_1v_1 + u_2v_2 + \ldots + u_nv_n.$$

Observemos que a operação < **u**, **v** > possui as propriedades usuais de um produto interno, ou seja, é positiva e definida, simétrica e bilinear[12].

Dizemos que **u** é ortogonal a **v** se, e somente se, < **u**, **v** > = 0.

Definição 3.12 (Código dual) Seja C $\subset \mathbb{F}_q^n$ um código linear com matriz geradora G. Definimos um conjunto $C^{\perp}$ tal que

$$C^{\perp} = \{v \in \mathbb{F}_q^n : < \mathbf{v}, \mathbf{u} > = 0, \ \forall\, u \in C\}.$$

[12] i) < **u**, **u** > ≥ 0, para todo **u** $\in \mathbb{F}_q^n$
ii) < **u**, **u** > = 0 se, e somente se, u = 0
iii) < **u**, **v** > = < **v**, **u** >, para todo **u**, **v** $\in \mathbb{F}_q^n$
iv) < **u** + α**w**, **v** > = < **u**, **v** > + α < **w**, **v** >, para todo **u**, **v**, **w** $\in \mathbb{F}_q^n$ e α $\in \mathbb{F}$.

O conjunto $C^{\perp}$ satisfaz as seguintes propriedades:

$i)$ $C^{\perp}$ é um subespaço vetorial de $\mathbb{F}_q^n$

Demonstração: De fato, se tomarmos **u**, **v** ∈ $C^{\perp}$, **x** ∈ C e **α** ∈ $\mathbb{F}$, verificamos para todo x ∈ C que < **u** + α**v**, **x** > = < **u**, **x** > + α< **v**, **x** > = 0 e, portanto, **u** + α**v** ∈ $C^{\perp}$. □

ii) **x** ∈ $C^{\perp}$ se, e somente se, G $\mathbf{x}^t = 0$

Demonstração: De fato, **x** ∈ $C^{\perp}$se, e somente se, **x** é ortogonal a todos os elementos de C. E **x** é ortogonal a todos os elementos de C se, e somente se, **x** é ortogonal a todos os elementos de uma base de C, o que é o mesmo que dizer G $x^t = 0$, pois as linhas de G são uma base de C. □

Assim, ante ao apresentado, provamos que $C^{\perp}$ é um subespaço vetorial de $\mathbb{F}_q^n$, ortogonal a C, o qual chamaremos de código dual de C.

A seguir serão apresentados alguns resultados importantes.

Teorema 3.2. (Dimensão e matriz geradora de $C^{\perp}$) Seja C ⊂ $\mathbb{F}_q^n$ um código de dimensão **m** e com matriz geradora G = [Id_m | A], na forma padrão. Então

i) dim $C^{\perp}$= **n – m**;

ii) H = [– A^t | Id_{n-m}] é uma matriz geradora de $C^{\perp}$.

Demonstração:

(i) Um vetor **v** = (v_1, v_2, ... , v_n) pertence a $C^{\perp}$ se, e somente se, G $\mathbf{v}^t = 0$. Como G está na forma padrão, isso equivale ao sistema:

$$\left[\begin{array}{ccccc|ccc} 1 & 0 & 0 & 0 & 0 & a_{1(m+1)} & \cdots & a_{1(n)} \\ 0 & 1 & 0 & 0 & 0 & a_{2(m+1)} & \cdots & a_{2(n)} \\ 0 & 0 & \ddots & \vdots & \vdots & \vdots & \cdots & \vdots \\ \vdots & \vdots & \vdots & 1 & 0 & \vdots & \cdots & \vdots \\ 0 & 0 & \cdots & 0 & 1 & a_{m(m+1)} & \cdots & a_{m(n)} \end{array}\right]_{m\,x\,n} \begin{bmatrix} v_1 \\ \vdots \\ \vdots \\ \vdots \\ v_n \end{bmatrix}_{n\,x\,1} = \begin{bmatrix} 0 \\ \vdots \\ \vdots \\ \vdots \\ 0 \end{bmatrix}_{m\,x\,1}$$

Daí,

$$\begin{cases} v_1 & & & + a_{1(m+1)}.v_{m+1} + \cdots + a_{1(n)}.v_n = 0 \\ & v_2 & & + a_{2(m+1)}.v_{m+1} + \cdots + a_{2(n)}.v_n = 0 \\ & & \ddots & \\ & & & v_m + a_{m(m+1)}.v_{m+1} + \cdots + a_{m(n)}.v_n = 0 \end{cases} \Rightarrow$$

$$\begin{cases} v_1 = -(a_{1(m+1)}.v_{m+1} + \cdots + a_{1(n)}.v_n) \\ v_2 = -(a_{2(m+1)}.v_{m+1} + \cdots + a_{2(n)}.v_n) \\ \vdots \\ \vdots \\ v_m = -(a_{m(m+1)}.v_{m+1} + \cdots + a_{m(n)}.v_n) \end{cases} \Rightarrow \begin{bmatrix} v_1 \\ \vdots \\ \vdots \\ \vdots \\ v_m \end{bmatrix} = -A. \begin{bmatrix} v_{m+1} \\ v_{m+2} \\ \vdots \\ \vdots \\ v_n \end{bmatrix}$$

Portanto, temos que $C^{\perp}$ possui **n – m** elementos (É a quantidade de elementos da última coluna: n-(m+1)+1). Logo, $C^{\perp}$ tem dimensão **n – m**. □

(ii) As linhas de H são linearmente independentes, devido ao bloco Id_{n-m}. Portanto, elas geram um subespaço vetorial de dimensão **n – m**. Como as linhas de H são ortogonais às linhas de G, temos que o espaço gerado pelas linhas de H está contido em $C^{\perp}$. E, como esses subespaços possuem a mesma dimensão, eles coincidem. Isso prova que $H = [- A^t \mid Id_{n-m}]$ é uma matriz geradora de $C^{\perp}$. □

Proposição 3.1. (Relação entre as matrizes geradoras dos códigos C e $C^{\perp}$) Suponhamos que $C \subset \mathbb{F}_q^n$ é um código de dimensão **m** e com matriz geradora G. Uma matriz H de ordem (n – m) x n, com coeficientes em $\mathbb{F}_2$ e com linhas linearmente independentes, é uma matriz geradora de $C^{\perp}$ se, e somente se, $G . H^t = 0$.

Demonstração: As linhas de H geram um subespaço vetorial de $\mathbb{F}_q^n$ que possui dimensão **n – m**, idêntica à dimensão de $C^{\perp}$. Além disso, se chamarmos de $h_1, \ldots h_{n-m}$ e $g_1, \ldots g_m$, respectivamente, as linhas de H e de G, então

$$(G.H^t)_{i,j} = < g_i, h_j >$$

Portanto, dizer que $G.H^t = 0$ é equivalente a dizer que todos os vetores do subespaço gerado pelas linhas de H estão em $C^{\perp}$. E, como esse subespaço tem a mesma dimensão de $C^{\perp}$, então

$$G.H^t = 0 \Leftrightarrow C^{\perp} \text{ é gerado pelas linhas de H. } \square$$

Proposição 3.2. $(C^{\perp})^{\perp} = C$

Demonstração: Sejam G e H, respectivamente, matrizes geradoras de C e $C^{\perp}$. Logo, $G.H^t = 0$. Tomando transpostas nessa última igualdade, temos que $H.G^t = 0$. Portanto, G é a matriz geradora de $(C^{\perp})^{\perp}$. Daí, segue-se o resultado. □

Proposição 3.3. Seja C um código linear e suponhamos que H seja uma matriz geradora de $C^{\perp}$. Então, temos que $\mathbf{v} \in C$ se, e somente se, $H\,\mathbf{v}^t = 0$.

Demonstração: Conforme já demonstrado, temos que $(C^{\perp})^{\perp} = C$ e $\mathbf{x} \in C^{\perp}$ se, e somente se, $G\,\mathbf{x}^t = 0$. Decorre dessas duas afirmativas que $v \in C$ se, e somente se, $v \in (C^{\perp})^{\perp}$ se, e somente se, $H\,\mathbf{v}^t = 0$. □

A matriz H de $C^{\perp}$ é chamada de matriz teste de paridade de C.

Por exemplo, se for dado um código C sobre $\mathbb{F}_2 = \{0, 1\}$ com matriz geradora

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

como G está na forma padrão ($G = [\ Id_m \mid A\ ]$), é fácil encontrar a matriz teste de paridade H, pois ela é $H = [-A^t \mid Id_{n-m}\ ]$. Portanto,

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

Assim, se quisermos verificar se as palavras $v_1 = 100111$ e $v_2 = 010101$ pertencem ao código, basta calcular

$$H\,v_1^t = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

e

$$H\,v_2^t = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}.$$

Assim, podemos concluir que $v_1 \in C$ e $v_2 \notin C$, pois $H\, v_1^t$ é o vetor nulo e $H\, v_2^t$ não é o vetor nulo.

A verificação de que uma palavra pertence ao código por meio da matriz de paridade requer um custo computacional muito menor do que o método apresentado anteriormente, por meio do qual era preciso resolver o sistema $x_1G = v_1$ e $x_2G=v_2$, onde $x_1, x_2 \in \mathbb{F}_q^m$, para verificar se os vetores pertenciam ao código.

A matriz teste de paridade também contém informações sobre o peso do código, conforme veremos no próximo item.

Proposição 3.4. Seja H a matriz teste de paridade de um código C. Temos que o peso de C é maior ou igual a *s* se, e somente se, quaisquer *s – 1* colunas de H são linearmente independentes.

Demonstração: Vamos supor, inicialmente, que cada conjunto de **s – 1** colunas de H seja linearmente independente. Tomemos uma palavra não nula **c** = (c_1, c_2, ..., c_n) do código C e digamos que h^1, h^2, ..., h^n sejam as colunas de H. Como $Hc^t = 0$, então

$$0 = H.c^t = \sum c_i h^i.$$

Se w (**c**) ≤ s – 1, como w (**c**) é o número de componentes não nulas da palavra **c** e a equação $0 = H.c^t = \sum c_i h^i$, então haveria uma combinação nula de um número t de colunas de H, 1 ≤ t ≤ s – 1, o que é uma contradição. Portanto, w (**c**) ≥ s e w (C) ≥ s.

Reciprocamente, suponhamos que w (C) ≥ s. Suponhamos ainda, por absurdo, que H tenha s – 1 colunas linearmente dependentes. Suponhamos que sejam as colunas $h^{i_1}, h^{i_2}, \dots, h^{i_{s-1}}$. Dessa forma, haveria $c_{i_1}, c_{i_2}, \dots, c_{i_{s-1}}$, nem todos nulos, tais que

$$c_{i_1} h^{i_1} + c_{i_2} h^{i_2} + \dots + c_{i_{s-1}} h^{i_{s-1}} = 0.$$

Logo, **c** = (0, ... , c_{i_1}, 0, ... , $c_{i_{s-1}}$, 0, ..., 0) ∈ C e, consequentemente, w (**c**) ≤ s – 1 < s, o que seria um absurdo. □

Teorema 3.3. Seja H a matriz teste de paridade de um código C. Temos que o peso de C é igual a *s* se, e somente se, quaisquer *s – 1* colunas de H são linearmente independentes e existem *s* colunas de H linearmente dependentes.

Demonstração: Vamos supor que w (C) = s. Logo, todo conjunto de **s – 1** colunas de H é linearmente independente. Portanto, existem **s** colunas de H linearmente dependentes, pois se

isso não acontecesse, teríamos w (C) ≥ s – 1, em observância ao apresentado na proposição anterior.

Suponhamos, agora, que todo conjunto de **s – 1** colunas de H é linearmente independente e existem s colunas linearmente dependentes. Daí, pela proposição anterior, temos que w (C) ≥ s. Todavia, w (C) não pode ser maior que **s**, pois se isso ocorresse, pela proposição anterior, teríamos que todo conjunto com **s** colunas de H é linearmente independente, o que é uma contradição. □

Teorema 3.4. Cota de Singleton

Os parâmetros (**n**, **m**, **d**) de um código linear satisfazem à desigualdade

$$d \leq n - m + 1.$$

Demonstração: A matriz teste de paridade H tem posto **n – m**. O Teorema 3.3 nos permite afirmar que d – 1 é menor ou igual ao posto de H. Portanto, daí segue a desigualdade. □

3.4 Um Exemplo de Código

Definição 3.13. (Código de Hamming) Um código de Hamming de ordem t sobre $\mathbb{F}_2$ é um código com matriz teste de paridade H_t de ordem **t** x **n**, em que **t = n – m**, cujas colunas são os elementos de F_2^t (exceto o vetor nulo de F_2^t) em uma ordem qualquer. A definição de H_t determina o código C a menos de equivalência.

Observa-se que o comprimento do código de Hamming de ordem t é $n = 2^t - 1 = 2^{n-m} - 1$. A dimensão do código é **m**.

Verifica-se que d = 3, pois em H_t é fácil achar três colunas linearmente dependentes.

Como exemplo, consideremos a matriz

$$H_3 = \begin{bmatrix} 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}.$$

Está é a matriz do código de Hamming correspondente a t = 3.

Teorema 3.5. Todo código de Hamming é perfeito.

Demonstração: No caso, como d =3, então **κ** = $\left[\frac{d-1}{2}\right]$= 1. Dado **c** em F_2^t, temos que | D (**c**, **1**) | = 1 + n.

Portanto,

$$|\bigcup_{c \in C} D(c,1)| = [1+n].2^m = [1+2^{n-m}-1].2^m = 2^n,$$

e, consequentemente,

$$\bigcup_{c \in C} D(c,1) = F_2^n. \square$$

3.5 Decodificação pela Síndrome

A decodificação é o procedimento de detecção e correção de erros em um código. O método que será apresentado aqui para decodificação é, segundo Hefez e Villela (2008, p. 104) um aperfeiçoamento do método inventado por D. Slepian que era utilizado no laboratório Bell na década de 60.

Inicialmente, vamos definir o vetor erro $\mathbf{e} \in \mathbb{F}_q^n$ como a diferença entre o vetor recebido $\mathbf{r} \in \mathbb{F}_q^n$ e o vetor transmitido $\mathbf{c} \in \mathbb{F}_q^n$:

$$\mathbf{e = r - c.}$$

Por exemplo, se em um código sobre $\mathbb{F}_2 = \{0, 1\}$ foi transmitida a palavra 010011 e foi recebida a palavra 101011, então o vetor erro é **e** = 101011 – 010011 = 111000.

O peso do vetor erro indica a quantidade de erros cometidos na transmissão. No exemplo anterior, o valor do peso do vetor erro é 3, o que indica que ocorreram 3 erros na transmissão.

Tomemos, agora, a matriz H que faz o teste de paridade do código e calculemos He^t. Como a palavra **c** pertence ao código, temos $Hc^t = 0$ e, então, verificamos que $He^t = H(r^t - c^t) = Hr^t$. Isso nos direciona ao próximo resultado.

Definição 3.14. (Síndrome de um vetor) Dados um código $C \subset \mathbb{F}_q^n$ e um vetor $v \in \mathbb{F}_q^n$, chamamos de síndrome de v o vetor Hv^t, onde H é a matriz teste de paridade do código C.

Como $He^t = Hr^t$, então podemos dizer que o vetor erro **e** e o vetor recebido **r** tem a mesma síndrome.

De fato, se chamarmos de h^i a i-ésima coluna da matriz teste de paridade H e tivermos um vetor erro $\mathbf{e} = (x_1, \ldots, x_n)$, então

$$\sum_{i=1}^{n} x_i h^i = x_1 \begin{pmatrix} h_{11} \\ \vdots \\ h_{(n-m)1} \end{pmatrix} + \cdots + x_n \begin{pmatrix} h_{1n} \\ \vdots \\ h_{(n-m)n} \end{pmatrix} =$$

$$= \begin{pmatrix} x_1 h_{11} + \cdots + x_n h_{1n} \\ \vdots \\ x_1 h_{(n-m)1} + \cdots + x_n h_{(n-m)n} \end{pmatrix} = He^t = Hr^t. \square$$

O resultado a seguir é de extrema importância, pois é por meio dele que é possível efetuar a correção de uma palavra transmitida com erro.

Teorema 3.6. Seja C um código linear em $\mathbb{F}_q^n$ com capacidade de correção **κ**. Se **r** ∈ $\mathbb{F}_q^n$ e **c** ∈ C ⊂ $\mathbb{F}_q^n$ são tais que d (**c**, **r**) ≤ **κ**, então existe um único vetor **e** com w(**e**) ≤ κ cuja síndrome é igual à síndrome de **r** e tal que **c** = **r** – **e**.

Demonstração: A demonstração deste teorema será dividida em três partes:

Primeira Parte: Provando a existência do vetor **e**

O problema que deveremos resolver é como determinar o vetor **e** a partir de Hr^t.

Suponhamos que um código C tenha distância mínima **d** ≥ 3 (pois caso contrário **κ** = 0 e nada teríamos para demonstrar) e que o vetor erro **e**, introduzido entre a palavra transmitida **c** e a palavra recebida **r**, seja tal que w(**e**) ≤ 1, ou seja, o canal introduziu no máximo um erro.

Se $He^t = 0$, então r ∈ C e tomamos **r** = **c**.

Se $He^t \neq 0$, então w(**e**) = 1 e, então, **e** tem apenas uma coordenada não nula. Nesse caso, consideremos que **e** = (0, ... , x, ..., 0) com x ≠ 0 na i-ésima posição.

Verificamos que

$He^t = xh^i$, onde h^i é a i-ésima coluna de H.

Portanto, não conhecendo **e**, mas sabendo que:

$He^t = Hr^t = xh^i$,

Podemos determinar **e** como sendo o vetor com todas as componentes nulas, exceto a i-ésima componentes que é x.

Para w(**e**) > 1, podemos fazer essa mesma construção, anexando mais coordenadas não nulas para o vetor **e**.

Além disso w(**e**) = w (**r** – **c**) = d(**c**, **r**) e d(**c**, **r**) ≤ κ, então existe um vetor **e** tal que w(**e**) ≤ κ.

Segunda Parte: Provando que a síndrome de **e** é igual a síndrome de **r**

De fato, **c** = **r** – **e** e $H\mathbf{e}^t = H(\mathbf{r}^t - \mathbf{c}^t) = H\mathbf{r}^t - H\mathbf{c}^t = H\mathbf{r}^t - 0 = H\mathbf{r}^t$.

Terceira Parte: Provando a unicidade de **e**

Para provar a unicidade, suponhamos que existam os vetores **e** = (x_1, ... , x_n) e **e'** = (x'_1, ... , x'_n), **e** ≠ **e'**, que satisfaçam w(**e**) ≤ κ e w(**e'**) ≤ κ e que tenham síndromes iguais. Então, se H é uma matriz teste de paridade de C, temos

$$He^t = H(e')^t \Rightarrow \sum_{i=1}^{n} x_i h^i = \sum_{i=1}^{n} x'_i h^i \Rightarrow x_1 h^1 + \cdots + x_n h^n - x'_1 h^1 - \cdots - x'_n h^n = 0$$

$$\Rightarrow (x_1 - x'_1)\, h^1 + \cdots + (x_n - x'_n)\, h^n = 0, \text{ ou seja,}$$

$$\sum_{i=1}^{n} (x_i - x'_i) h^i = 0$$

Como w(**e**) ≤ κ e w(**e'**) ≤ κ, ou seja, há no máximo κ entradas não nulas em cada vetor, consequentemente no máximo **2κ** coeficientes $(x_i - x'_i) \neq 0$, com i ∈ {1, 2, ... , n}, na combinação linear . Mas

$$\sum_{i=1}^{n} (x_i - x'_i) h^i = 0,$$

o que nos fornece uma relação de dependência linear entre m colunas de H com m ≤ 2κ ≤ d–1.

Como quaisquer d – 1 colunas de H são linearmente independentes, conforme o teorema 3.3, temos que $x_i = x'_i$ para todo i e, portanto, **e** = **e'**. □

3.5.1 O algoritmo para códigos que corrigem um erro

Seja C ⊂ $\mathbb{Z}_2^5$ o código gerado pela matriz

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

Suponhamos que o vetor erro **e**, introduzido entre a palavra transmitida **c** e a palavra recebida **r**, seja tal que w(**e**) ≤ 1. Suponhamos, ainda, que tenha sido enviada a palavra **c** = 10110 e recebida a palavra **r** = 11110. Estudaremos a decodificação de **r**.

Inicialmente, verificamos que a matriz G está na forma padrão G = [Id_m | A]; que a dimensão do código é **m** = 2, pois a imagem do código é gerada por dois vetores; e que o tamanho das palavras é **n** = 5. Com essa informações, podemos construir a matriz teste de paridade H = [– A^t | Id_{n-m}]:

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

Assim, após obter H, podemos calcular Hr^t:

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

e podemos concluir que a palavra foi transmitida com erro, pois o resultado Hr^t não foi o vetor nulo. Dessa forma, como $w(\mathbf{e}) \leq 1$, podemos concluir que há um erro na transmissão e que o vetor erro terá a forma $\mathbf{e} = (0, \ldots, x, \ldots, 0)$ com $x \neq 0$ na i-ésima posição.

Então, devemos determinar qual é a i-ésima posição. Para tanto, devemos calcular $He^t = Hr^t$, mas $He^t = xh^i$, onde h^i é a i-ésima coluna de H. Fazendo a conta, verificamos que $\mathbf{i} = 2$, pois

$$\begin{bmatrix} 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} = 1 \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} = 1.h^2,$$

e, então, escrevemos o vetor erro $\mathbf{e} = (0, 1, 0, 0, 0)$. Agora, falta apenas achar **c**. Como $\mathbf{e} = \mathbf{r} - \mathbf{c}$, ou ainda $\mathbf{c} = \mathbf{r} - \mathbf{e}$, obtemos **c** fazendo: $c = 11110 - 01000 = 10110$, que foi a palavra transmitida.

Diante do exemplo apresentado, podemos descrever um algoritmo que permite a decodificação em códigos corretores de até um erro:

Seja H a matriz teste de paridade de um código C e seja **r** um vetor recebido (suponhamos $\mathbf{d} \geq 3$)

i) Calcule Hr^t.

ii) Se $Hr^t = 0$, aceite **r** como a palavra transmitida.

iii) Se $Hr^t = s^t$, compare s^t com as colunas de H.

iv) Se existirem i e x tais que $s^t = xh^i$, para $x \in \mathbb{F}$, então **e** é a n-upla com x na posição i e zeros nas outras posições. Corrija **r** pondo $\mathbf{c} = \mathbf{r} - \mathbf{e}$.

v) Se o contrário de (iv) ocorrer, então mais de um erro foi cometido.

3.5.2 Classes laterais

Tomemos um vetor $\mathbf{v} \in \mathbb{F}_q^n$. Definimos o conjunto classe lateral de **v** segundo o código $C \subset \mathbb{F}_q^n$ como sendo

$$\mathbf{v} + \mathbf{c} = \{v + c;\ c \in C\}.$$

Esse conjunto possui algumas propriedades, as quais serão elencadas a seguir e cuja demonstração pode ser verificada em Hefez e Villela (2008, p. 107-109).

i) Os vetores **u** e **v** de $\mathbb{F}_q^n$ têm a mesma síndrome se, e somente se, $\mathbf{u} \in \mathbf{v} + C$;

ii) $\mathbf{v} + C = \mathbf{v'} + C \Leftrightarrow \mathbf{v} - \mathbf{v'} \in C$

iii) $(\mathbf{v} + C) \cap (\mathbf{v'} + C) \neq \emptyset \Rightarrow \mathbf{v} + C = \mathbf{v'} + C$

iv) $\bigcup_{v \in \mathbb{F}_q^n} (\boldsymbol{v} + C) = \mathbb{F}_q^n$

v) $|(\mathbf{v} + C)| = |C| = q^m$.

O mais importante das classes laterais é que todos os elementos de uma classe têm a mesma síndrome e que os elementos de classes laterais distintas têm síndromes diferentes. Estas características definem uma partição em $\mathbb{F}_q^n$.

Por exemplo, seja $C \subset \mathbb{F}_2^4$ o código gerado pela matriz

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

Nele, temos que: a dimensão do código é **m** = 2, pois a matriz G tem duas linhas; o tamanho das palavras é **n** = 4, pois a matriz G tem 4 colunas; o valor de **q** = 2, pois o alfabeto $\mathbb{F}$ tem 2 elementos.

Com essas informações, verificamos que **n** – **m** = 2 e, portanto, o número de elementos de cada classe lateral será $4 = 2^2 = q^m$. O número de classes laterais é $\frac{q^n}{q^m} = q^{n-m} = 2^2 = 4$.

Por meio da matriz G também podemos obter as palavras do código. Basta fazer

$$[0\ \ 0]\begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix} = [0\ \ 0\ \ 0\ \ 0];\ [1\ \ 0]\begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix} = [1\ \ 0\ \ 1\ \ 1];$$

$$[0\ \ 1]\begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix} = [0\ \ 1\ \ 0\ \ 1];\ [1\ \ 1]\begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix} = [1\ \ 1\ \ 1\ \ 0].$$

Ante o exposto, verificamos que as classes laterais são as seguintes:

$$0000 + C = \{0000, 1011, 0101, 1110\}$$
$$1000 + C = \{1000, 0011, 1101, 0110\}$$
$$0100 + C = \{0100, 1111, 0001, 1010\}$$
$$0010 + C = \{0010, 1001, 0111, 1100\}.$$

Um vetor de peso mínimo em uma classe lateral é chamado de elemento líder dessa classe. Então, 0000 é o líder de C; 1000 é o líder de 1000 + C; 0100 e 0001 são líderes de 0100 + C; e 0010 é líder de 0010 + C.

O exemplo anterior mostrou que uma classe lateral pode ter mais de um líder. Todavia, o que se busca é a construção de um código em que cada classe tenha um único líder. Verifiquemos, então, os resultados a seguir.

Teorema 3.7. Tomando-se um código linear $C \subset \mathbb{F}_q^n$ com distância mínima **d**, se $\mathbf{u} \in \mathbb{F}_q^n$ é tal que w(**u**) $\leq \left[\frac{d-1}{2}\right]$, então **u** é o único elemento líder de sua classe.

Demonstração: Suponhamos que **u**, **v** $\in \mathbb{F}_q^n$ com w(**u**) $\leq \left[\frac{d-1}{2}\right]$ e w(**v**) $\leq \left[\frac{d-1}{2}\right]$. Se **u** e **v** são elementos da mesma classe, então **u** – **v** $\in$ C, pois

$$\begin{cases} u \in y + C \Rightarrow u = y + c_1 \\ v \in y + C \Rightarrow v = y + c_2. \\ u - v = c_1 - c_2 \in C \end{cases}$$

Nesse caso, w(**u** – **v**) $\leq$ w(**u**) + w(**v**) $\leq \left[\frac{d-1}{2}\right] + \left[\frac{d-1}{2}\right] \leq d - 1$. Como w(**c**) $\geq$ **d** para todo **c** $\in$ C $\neq$ 0, então **u** – **v** = 0 e, portanto, **u** = **v.** □

Observação importante: O código apresentado na seção anterior não possui um único elemento líder em suas classes laterais pois d = 2, logo $\left[\frac{d-1}{2}\right] = 0$.

3.5.3 Decodificação pela síndrome

Após a apresentação das informações anteriores, podemos apresentar um algoritmo de correção de mensagens que tenham sofrido uma quantidade de erros menor ou igual a **κ** = $\left[\frac{d-1}{2}\right]$.

Inicialmente, devemos determinar os elementos $\mathbf{u} \in \mathbb{F}_q^n$ tais que $w(\mathbf{u}) \leq \left[\frac{d-1}{2}\right]$. Em seguida, devemos calcular as síndromes desses elementos e colocá-las em uma tabela.

Então, ao recebermos uma palavra **r**, devemos:

i) Calcular a síndrome $s^t = Hr^t$

ii) Se **s** está na tabela construída, tomamos o elemento **l** líder da classe tal que $H.\mathbf{l}^t = \mathbf{s}^t$. Em seguida, trocamos **r** por **r** – **l**, pois **r – l = c**, onde **c** é a palavra do código transmitida.

iii) Se **s** não está na tabela, então na mensagem recebida foram cometidos mais do que $\boldsymbol{\kappa} = \left[\frac{d-1}{2}\right]$ erros.

Por exemplo, consideremos um código linear $C \subset \mathbb{F}_2^6$ com matriz teste de paridade

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

Como a matriz teste de paridade é uma matriz de ordem (n – m) x n, então temos que **n – m** = 3 e **n** = 6. Daí, temos que a dimensão do código é **m** = 3. Além disso, verificamos que as colunas da matriz H são linearmente independentes 2 a 2 e existem três colunas linearmente dependentes (primeira, segunda e quarta), podemos dizer que a distância mínima do código é **d** = 3 e, portanto, ele pode corrigir até $\kappa = \left[\frac{d-1}{2}\right] = 1$ erro.

Vamos escrever os vetores de $\mathbb{F}_2^6$ com $w(\mathbf{u}) \leq 1$ e suas respetivas síndromes:

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

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

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

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

Então:

Tabela 3

Líder	Síndrome
000000	000
000001	101
000010	011
000100	110
001000	001
010000	010
100000	100

Fonte: FAMAT em Revista.
Disponível em: http://www.portal.famat.ufu.br/sites/famat.ufu.br/files/Anexos/Bookpage/Famat_Revista_06.pdf

Suponhamos que tenha sido recebida a palavra **r** = 100011. Logo, como $Hr^t = (010)^t$, temos que e = 010000. Consequentemente, **c** = **r** – **e** = 100011 – 010000 = 110011.

Suponhamos, agora, que tenha sido recebida a palavra **r** = 111111. Logo, como $Hr^t = (111)^t$ não se encontra na tabela, foi cometido mais que um erro na mensagem.

Tomemos outro exemplo. Seja $C \subset \mathbb{F}_2^9$ um código linear gerado pela matriz

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

e que possui distância mínima **d** = 5.Por meio da matriz geradora, podemos identificar que a dimensão do código é **m** = 2 e o tamanho das palavras é **n** = 9.

A matriz G não está na forma padrão. Então, por meio de operações elementares entre as colunas, podemos deixá-la no formato desejado. Portanto,

$$G = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 \end{bmatrix} c_2 \leftrightarrow c_5$$

$$G' = \begin{bmatrix} 1 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \end{bmatrix} = [Id_2 \mid A].$$

Agora, podemos montar uma matriz H de teste de paridade,

$$H = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} = [-A^t \mid Id_7].$$

No caso, temos que $\left[\frac{d-1}{2}\right] = 2$ e, portanto, a tabela deve ser montada com os vetores de peso ≤ 2 e suas respectivas síndromes:

Tabela 4 - Vetores na forma transposta (v^t)

0	1	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0
0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0
0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	1	0	0	0	1	1	1	1	0	0	0
0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	1	0	0	1	0	0	1	1	1	0
0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	1	0	0	1	0	1	0	1	1
0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	1	0	0	1	0	1	0	1

Tabela 5 - Síndrome na forma transposta (Hv^t)

0	1	0	1	0	0	0	0	0	0	1	0	1	1	1	1	1	1	1	0	0	0	0	0	0	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	1	0	1	0	0	0	0	0	0	1	0	1	1	1	1	1	1	0	1	1	1	1	1	1	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	1	1	1	0	1	1	1	1	0	0	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	1	1	1	1	0	0	0	0	0	0
0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	1	1	0	1	1	1	0	0	1	0	0	0	0	1	0	0	0	1	0	0	1	1	1	0	0	0
0	0	1	0	0	0	0	1	0	0	1	0	0	0	0	1	0	0	1	1	1	1	0	1	1	0	0	0	1	0	0	0	0	1	0	0	0	1	0	0	1	1	0	0	0
0	1	0	0	0	0	0	0	1	0	1	1	1	1	1	1	1	0	1	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	1	0	0	0	1	0	1	0	1	0
0	0	1	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1	1	1	1	1	1	1	0	0	0	0	0	1	0	0	0	0	1	0	0	0	1	0	1	0	1	1	1

Suponhamos que a palavra recebida seja r = 111011000. Verificamos que Hr^t = $(0000111)^t$, valor que não consta na tabela, o que significa que na palavra foram cometidos mais de dois erros.

Agora consideremos a palavra recebida r = 111011100. Verificamos que Hr^t = $(0000011)^t$, palavra que existe na tabela. Tomamos então o líder da classe correspondente, que é o vetor 0000011 e obtermos a palavra transmitida c = 111011100 – 0000011 = 111011111, a qual pertence ao código, pois $Hc^t = 0$.

4 FUNDAMENTAÇÃO PEDAGÓGICA

Segundo consta nas Orientações Educacionais Complementares aos Parâmetros Curriculares Nacionais para o Ensino Médio (PCN+, 2000, p. 8), o referido nível de ensino tem por objetivo "preparar para a vida, qualificar para a cidadania e capacitar para o aprendizado permanente, em eventual prosseguimento dos estudos ou diretamente no mundo do trabalho". É em busca dessa finalidade que a área das Ciências da Natureza, Matemática e suas Tecnologias se apresenta como capaz de desenvolver competências que envolvem a representação e a comunicação; a investigação e compreensão; e a contextualização sócio-cultural.

Abordando-se mais especificamente o conhecimento matemático, é possível verificar que ele desenvolve competências que auxiliam na tomada de decisões e também ampliam as habilidades do pensamento, ou seja, ele é capaz de ajudar na interpretação da realidade e no desenvolvimento de habilidades ao longo da vida social e profissional.

Para que o Ensino Médio alcance os objetivos acima propostos e consiga efetivamente desenvolver as competências descritas, é preciso que o docente esteja preparado para utilizar novas técnicas de ensino-aprendizagem. Sobre o trabalho do professor, afirma Parra e Saiz (*apud* Vilela, 2008, p. 8, 9):

> O trabalho do professor consiste [...] em propor ao aluno uma situação de aprendizagem para que elabore seus conhecimentos como resposta pessoal a uma pergunta, e os faça funcionar ou os modifique como resposta às exigências do meio e não a um desejo do professor. (Parra e Saiz, 1996, p. 49)

Portanto, o docente deve estar pronto para propor atividades que envolvam a contextualização dos conteúdos, a resolução de problemas, a aplicação de jogos e o uso de computadores com a finalidade de construir o conhecimento. Dessa forma, o professor poderá contribuir para que os estudantes estejam preparados para os vários desafios que encontrarão quando finalizarem o Ensino Básico, uma vez que a sociedade está cada vez mais globalizada e informatizada.

4.1 A Importância da Contextualização Histórica da Matemática

A análise do contexto histórico na construção dos conceitos matemáticos é muito importante. Segundo D' Ambrosio:

> "As ideias matemáticas comparecem em toda a evolução da humanidade, definindo estratégias de ação para lidar com o ambiente, criando e desenhando instrumentos para esse fim, e buscando explicações sobre os fatos e fenômenos da natureza e para a própria existência. Em todos os momentos da história e em todas as civilizações, as ideias matemáticas estão presentes em todas as formas de fazer e de saber." (D' Ambrosio, 1999, p. 97)

Assim, podemos afirmar que a análise histórica contribui para a construção do conhecimento pelos alunos, pois permite dar significado aos conteúdos escolares, uma vez que ajuda os educandos a entenderem os motivos que levam pessoas a fazerem Matemática e que as necessidades (sejam elas práticas, sociais, econômicas ou físicas) normalmente estimulam o desenvolvimento matemático.

É claro que a formalização de uma teoria utilizada hoje não é idêntica à realizada no passado, quando foi obtido aquele resultado matemático. Entretanto, é importante que o aluno compreenda em que circunstâncias a teoria foi formulada, pois isso permitirá uma melhor compreensão dos conceitos e a construção do conhecimento. Nesse sentido, Viana e Silva afirmam que

> "[...] a partir do momento que se conhece a HM [História da Matemática], as aulas ficam mais interessantes e com aprendizado de qualidade" (Viana e Silva, 2007, p. 6).

Ressalte-se que a contextualização histórica de um conteúdo matemático não serve apenas para se tornar uma aula mais interessante ou para dar significado a um conteúdo. Ela vai além, pois ajuda o aluno a perceber que a Matemática não está isolada dos demais saberes. Segundo Miguel e Miorim (*apud* Oliveira, Alves e Neves), as contribuições são da forma:

> (1) "A matemática como uma criação humana; (2) as razões pelas quais as pessoas fazem Matemática; (3) as necessidades práticas, econômicas e físicas que servem de estímulo ao desenvolvimento das ideias matemáticas; (4) as conexões existentes entre a matemática e filosofia, matemática e religião, matemática e lógica,

> etc.; (5) a curiosidade estritamente intelectual que pode levar a generalização e extensão de ideias e teorias; (6) as percepções que os matemáticos têm do próprio objeto da matemática, as quais mudam e se desenvolvem ao longo do tempo; (7) a natureza de uma estrutura, de um axiomatização e de uma prova." (Miguel e Miorim, 2004, p. 33)

Portanto, se a história da Matemática for utilizada em sala de aula, será criado um ambiente que proporciona ao aluno um aprendizado significativo, por meio do qual será possível que o educando participe da descoberta dos conteúdos, observe os aspectos humanos envolvidos no desenvolvimento de uma teoria e analise o contexto histórico de surgimento dela (as circunstâncias em que ela foi desenvolvida, crenças, emoções, afetos e outros itens envolvidos no surgimento da teoria).

4.2 A Resolução de Problemas como Estratégia de Ensino-aprendizagem

O ensino da Matemática por meio da resolução de problemas auxilia na contextualização dos conteúdos, uma vez que a Matemática é utilizada para resolver diversos problemas e situações no cotidiano.

A recomendação de que o ensino seja contextualizado está nos PCN+:

> "Aprender Matemática de uma forma contextualizada, integrada e relacionada a outros conhecimentos traz em si o desenvolvimento de competências e habilidades que são essencialmente formadoras, à medida que instrumentalizam e estruturam o pensamento do aluno, capacitando-o para compreender e interpretar situações, para se apropriar de linguagens específicas, argumentar, analisar e avaliar, tirar conclusões próprias, tomar decisões, generalizar e para muitas outras ações necessárias à sua formação." (Brasil, 2010, p. 111)

Para que a metodologia possa ser utilizada, é importante que alunos e professores saibam a diferença entre um problema matemático e um exercício matemático. O primeiro é definido como

> "uma situação que demanda a realização de uma sequência de ações ou operações para obter um resultado. Ou seja, a solução não está disponível de início, mas é possível construí-la" (PCN, 1998, p. 41)

ou, segundo Silveira, como

> “toda situação requerendo a descoberta de informações matemáticas desconhecidas para a pessoa que tenta resolvê-lo e/ou a invenção de uma demonstração de um resultado matemático dado.” (Silveira, 2001)

O segundo, de acordo com Silveira é

> “uma atividade de adestramento no uso de alguma habilidade/conhecimento matemático já conhecido pelo resolvedor, como a aplicação de um algoritmo conhecido, de uma fórmula conhecida” (Silveira, 2001)

Dessa forma, se na atividade que está sendo realizada os alunos conseguem interpretar um enunciado, estruturam as situações apresentadas (não necessariamente todas elas), desenvolvem uma ou mais estratégias de resolução e efetuam a verificação do resultado, então está sendo utilizado um problema matemático. Ressalta-se que o foco não é ensinar os alunos a resolverem problemas, mas sim ensinar Matemática por meio da resolução de problemas.

Nesse sentido, para que um professor utilize o método de resolução de problemas para o ensino de Matemática, ele deve desenvolver atividades que despertem a curiosidade dos alunos – com problemas compatíveis com o desenvolvimento intelectual deles – e que sejam capazes de estimular os educandos a desenvolverem um trabalho investigativo – seja com o auxílio do professor ou com o auxílio de colegas –, a fim de encontrar a solução do problema. Essa maneira de interagir se assemelha ao que é sugerido por Vygotsky, pois permite que o discente construa o conhecimento, passando da zona de desenvolvimento proximal (na qual não consegue realizar todas as atividades sozinho, sendo dependente do auxílio recebido pelo professor ou por outros colegas) para a zona de desenvolvimento real (quando, então, terá apreendido as novas teorias matemáticas e será capaz de resolver os problemas propostos de forma independente).

Vygotsky (1999) ainda destaca a importância dos problemas para a formação dos conceitos:

> “A formação de conceitos é o resultado de uma complexa atividade em que todas as funções intelectuais básicas tomam parte. No entanto, o processo não pode ser reduzido à associação, à atenção, à formação de imagens, à inferência ou às tendências dominantes. Todas são indispensáveis, porém insuficientes sem o uso do signo, ou a palavra, como meio pelo qual conduzimos as nossas operações mentais, controlamos o seu curso e as canalizamos em direção à solução de um problema.” (Vygotsky, 1999, p. 72-73)

o que corrobora a tese de que a solução de problemas pode ser utilizada para a construção dos conceitos matemáticos.

4.3 O Uso do Computador em Sala de Aula

Atualmente vivemos em uma sociedade em que a quantidade de informações transmitidas e a velocidade de transmissão estão cada vez maiores. A internet se tornou algo praticamente essencial no cotidiano das pessoas e é notória uma constante busca de inserção social e participação na nova realidade digital pelos indivíduos. A escola não pode ficar de fora dessa nova realidade. Ela precisa criar situações em que seja possível a utilização do potencial didático dos computadores, softwares, tablets, smartphones, internet, etc. na colaboração ao processo de ensino-aprendizagem.

Isso é o que propõe Masseto (*apud* Quiles, 2009),

> "é impossível dialogarmos sobre tecnologia e educação, inclusive educação escolar, sem abordarmos a questão do processo de aprendizagem. Com efeito, a tecnologia apresenta-se como meio, como instrumento para colaborar no desenvolvimento do processo de aprendizagem" (Masseto, 2006, p. 139)

Para que essa colaboração ao processo de ensino-aprendizagem ocorra, é necessário mais que o simples uso de um computador em sala de aula. O domínio do uso de um software por parte do docente também não garante que serão criadas situações favoráveis ao aprendizado. É preciso algo mais. O professor tem que reconstruir o conhecimento, modificar sua prática. Nesse sentido, afirma Oliveira (2007, p. 59):

> Os recursos computacionais em si mesmos, quando amplamente dominados pelo professor, não são suficientes para garantir uma ação educacional diferenciada, se não estiverem claras e fundamentas as teorias. Assim, além da necessidade de saber lidar com o computador, o professor deve entregar-se ao processo de construir para si mesmo um novo conhecimento, incorporando não somente os princípios que estão sendo atualmente desenvolvidos sobre informática e educação, mas acima de tudo, passando pelas considerações teóricas sobre a aprendizagem que melhor explicam a aquisição do conhecimento e o desenvolvimento cognitivo. Trata-se de dominar o conhecimento científico de uma maneira ampla e necessária para o seu próprio aprimoramento intelectual.

Valente (1999) também nos diz que

> Caberá ao professor saber desempenhar um papel de desafiador, mantendo vivo o interesse do aluno em continuar a buscar novos conceitos e estratégias de uso desses conceitos, incentivando relações sociais de modo que os alunos possam aprender uns com os outros a trabalhar em grupo. Além disso, o professor deverá servir como modelo de aprendiz e ter um profundo conhecimento dos pressupostos teóricos que embasam os processos de construção de conhecimento e das tecnologias que podem facilitar esses processos. (1999, p. 40)

Portanto, o computador não deve ser utilizado para tornar informatizada uma aula tradicional. Isso não trará nenhum benefício. Ele deve ser empregado para criar um ambiente colaborativo, investigativo, no qual os alunos possam formular e testar conjecturas. Um ambiente no qual os educandos sejam ativos e busquem a construção do conhecimento, não sendo recipientes em que as informações são "depositadas". Se o computador for empregado com esse fim, haverá o desenvolvimento nos alunos de habilidades voltadas à organização do pensamento e raciocínio, além da ampliação de uma postura autônoma e reflexiva, por meio do desenvolvimento de habilidades voltadas à tomada de decisões e construção do próprio conhecimento.

Além do exposto, para que o computador cumpra seu papel de colaborar no processo de ensino-aprendizagem, é preciso que suas potencialidades sejam aproveitadas. Isso pode ser realizado por meio de softwares educacionais ou até mesmo por meio do uso de softwares não predominantemente educacionais – como planilhas eletrônicas. Esses programas permitem a representação dos objetos matemáticos de forma diferenciada e facilitam a compreensão dos conteúdos. Além disso, podem ser utilizados para que procedimentos rotineiros da matemática – contas grandes, por exemplo – sejam realizados de forma mais rápida e, então, os educandos ficam com mais tempo para a reflexão e a análise das situações-problema.

Dessa forma, com o uso do potencial dos computadores e com uma mudança na prática docente, as aulas serão mais prazerosas e dinâmicas e os alunos terão outra visão sobre o que é a Matemática.

4.4 Paradigma Construtivista

Os Parâmetros Curriculares Nacionais para o Ensino Médio recomendam o desenvolvimento de uma proposta de ensino que busque o

> "desenvolvimento de conhecimentos práticos, contextualizados, que respondam às necessidades da vida contemporânea, e o desenvolvimento de conhecimentos mais amplos e abstratos, que correspondam a uma cultura geral e a uma visão de mundo" (BRASIL, 2000, p. 6).

Caminha nesse sentido o paradigma construtivista, no qual a Matemática não é vista como objeto de ensino e sim objeto de aprendizagem, por meio do qual é procurada a inserção do aluno em situações de investigação, exploração e descobrimento, ou seja, o discente passa a ser considerado ativo no processo de construção do conhecimento.

São vários os autores que oferecem subsídios teóricos sobre o construtivismo, podendo-se citar Piaget, Vygotsky e Ausubel. As atividades a serem propostas, na concepção teórica piagetiana, devem provocar desequilíbrios que, após equilibração e acomodação, levam os alunos ao aprendizado. Na concepção de Vygotsky, as atividades atuam na zona de desenvolvimento proximal, buscando a ligação entre o nível de desenvolvimento real e o potencial. E quanto à concepção teórica de Ausubel, afirma Moreira e Masini (2006) que para ocorrer uma aprendizagem significativa é preciso estabelecer, *a priori*, uma comparação entre as concepções que o aluno já possui e o conceito a ser apresentado, devendo o professor promover a interação da informação nova com estruturas de conhecimento existentes nos indivíduos. Essas estruturas já existentes são definidas como **conceitos subsunçores** Assim, Ausubel define o processo de aprendizagem significativa como aquele em que o conceito subsunçor (que é o conceito de referência dos indivíduos) é transformado ou adaptado ao conceito novo, sendo estabelecido um novo elemento significativo, sem que os conceitos percam seus significados.

Estas considerações estão em consonância com os PCN do Ensino Médio, uma vez que ao estabelecer que os alunos devam ver a Matemática como ciência, é afirmado que é

> "importante que o aluno perceba que as definições, demonstrações e encadeamentos conceituais e lógicos têm a função de construir novos conceitos e estruturas a partir de outros e que servem para validar intuições e dar sentido às técnicas aplicadas" (BRASIL, 2000, p. 40-41).

As propostas baseadas na abordagem construtivista abrangem, dentre outras, situações que envolvem a resolução de problemas como proposta metodológica, o uso de computadores e a história da Matemática como fator motivacional. A primeira tem por objetivo estimular os

alunos, com o uso de situações problemas, a serem investigadores e exploradores na busca e construção dos novos conceitos. A segunda é de extrema importância, haja vista a influência da tecnologia no cotidiano dos discentes. Valente (1999, p. 24-25) afirma que

> "[...] o computador pode enriquecer ambientes de aprendizagem onde o aluno, interagindo com os objetos desse ambiente, tem a chance de construir seu conhecimento. Nesse caso, o conhecimento não é passado ao aluno. O aluno não é mais instruído, ensinado, mas é o construtor do seu próprio conhecimento. Esse é o paradigma construcionista onde a ênfase está na aprendizagem ao invés de estar no ensino; na construção do conhecimento e não na instrução." (Valente ,1999, p. 24-25)

E a terceira é realizada pela contextualização histórica do objeto de estudo, inserindo-o num processo histórico e cultural, como resultado da construção humana, conforme sugerido nos PCN+ do Ensino Médio (BRASIL, 2002).

Dessa forma, por meio do paradigma construtivista, é possível atingir os objetivos propostos nos PCN do Ensino Médio, haja vista que se pode cultivar no aluno, segundo uma abordagem formativa da Matemática,

> "[...] a capacidade de resolver problemas genuínos, gerando hábitos de investigação, proporcionando confiança e desprendimento para analisar e enfrentar situações novas, propiciando a formação de uma visão ampla e científica da realidade, a percepção da beleza e da harmonia, o desenvolvimento da criatividade e de outras capacidades pessoais." (BRASIL, 2000, p.6)

e, numa abordagem instrumental, prepara-se o discente para a atividade profissional, desenvolvendo a iniciativa e a segurança para adaptar os conceitos matemáticos em diferentes contextos. Com isso, prepara-se o aluno para selecionar informações, analisá-las, tomar decisões, bem como avaliar possibilidades e adequações tecnológicas em diferentes situações.

5 PROPOSTAS DE ATIVIDADES

5.1 Conteúdo Disciplinar

Nos códigos lineares corretores de erros são utilizados, principalmente, os seguintes conteúdos: sistema de numeração (na escolha do alfabeto), operações com vetores (soma de vetores, produto por escalar, dependência e independência linear), matrizes (na matriz geradora G e na matriz teste de paridade H) e sistemas lineares (na verificação de que uma palavra pertence ao código sem o uso da matriz H).

5.2 Proposta

A proposta de atividades que será apresentada neste capítulo foi elaborada para atingir todos os objetivos elencados na fundamentação pedagógica. Para tanto, serão apresentados aos discentes exemplos de como a Matemática está presente no cotidiano. Além disso, serão analisadas situações-problema contextualizadas, as quais, para serem resolvidas, exigirão que os alunos adotem uma postura crítica e reflexiva, sendo ativos no processo de ensino-aprendizagem.

As referidas atividades, além de tornarem o estudo da Matemática mais prazeroso, ainda auxiliarão no desenvolvimento do pensamento algébrico pelos educandos e, também, podem despertar o interesse deles pela ciência, fazendo com que alguns dos alunos se voluntariem a participar em projetos de iniciação científica para o Ensino Médio.

5.3 Atividades

5.3.1 Atividade 1

Título: Um exemplo de transmissão de informações: “passando cola”

Material/Recurso necessário: quadro negro (ou branco), giz (ou caneta piloto), uma avaliação interdisciplinar (impressa para cada um dos alunos), pequenos pedaços de papel.

Divisão da Turma: não haverá formação de grupos nessa atividade. Os alunos deverão responder a todas as questões da prova e poderão consultar aos colegas que estejam próximos a eles.

Tempo estimado: 60 minutos

Descrição sucinta:

Esta atividade tem por obtivo mostrar aos educandos como é possível codificar e decodificar informações.

Aspectos operacionais:

O professor apresentará aos alunos uma prova objetiva, ou seja, uma "prova de múltipla escolha", a qual pode conter questões sobre atualidades ou algum conteúdo da Matemática ou de outras disciplinas. Um exemplo de avaliação está no apêndice A.

O docente irá pedir aos alunos, antes de iniciar a prova, que pensem nas maneiras possíveis de passar cola a um colega com o uso de um pequeno pedaço de papel. Em seguida, o docente deve propor uma forma de transmissão de respostas, sem que seja notado que se trata de algo relativo à prova. Para isso, a ferramenta a ser utilizada será um algoritmo matricial. Façamos o seguinte exemplo. Suponhamos que a questão e sua resposta sejam codificadas por um vetor da forma $\begin{bmatrix} x \\ y \end{bmatrix}$ em que x é o número da questão (1, 2, 3, ... , 8) e y é a alternativa correta, porém as respostas referentes às letras **a**, **b**, **c**, **d**, **e** devem ser trocadas pelos números 1, 2, 3, 4, 5, respectivamente. Dessa forma, a codificação da fonte determina que se a resposta da questão 1 for a letra a, deve ser utilizado o vetor $\begin{bmatrix} 1 \\ 1 \end{bmatrix}$; e se a resposta da questão 6 for a letra c, deve ser utilizado o vetor $\begin{bmatrix} 6 \\ 3 \end{bmatrix}$.

Em seguida, o professor deve escolher uma matriz 2 x 2 invertível[13]. Vamos utilizar, nesta exemplificação, a matriz $A = \begin{bmatrix} 2 & 3 \\ 1 & 2 \end{bmatrix}$ (esta será a matriz que, quando multiplicada pelo cada um dos vetores correspondentes à questão e sua respectiva resposta, irá gerar cada uma das palavras do código de canal).

Portanto, podemos exemplificar que o código de canal de todas as possíveis respostas da questão 1 é obtido por meio dos produtos a seguir:

$$Questão\ 1 - letra\ a: \begin{bmatrix} 2 & 3 \\ 1 & 2 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 5 \\ 3 \end{bmatrix}$$

$$Questão\ 1 - letra\ b: \begin{bmatrix} 2 & 3 \\ 1 & 2 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 2 \end{bmatrix} = \begin{bmatrix} 8 \\ 5 \end{bmatrix}$$

[13] O motivo de a matriz A ser necessariamente uma matriz invertível é que, se ela não o for, a equação matricial AX = B não terá solução. Isso será explicado com mais detalhes na próxima página.

$$Questão\ 1 - letra\ c\text{: } \begin{bmatrix} 2 & 3 \\ 1 & 2 \end{bmatrix} . \begin{bmatrix} 1 \\ 3 \end{bmatrix} = \begin{bmatrix} 11 \\ 7 \end{bmatrix}$$

$$Questão\ 1 - letra\ d\text{: } \begin{bmatrix} 2 & 3 \\ 1 & 2 \end{bmatrix} . \begin{bmatrix} 1 \\ 4 \end{bmatrix} = \begin{bmatrix} 14 \\ 9 \end{bmatrix}$$

$$Questão\ 1 - letra\ e\text{: } \begin{bmatrix} 2 & 3 \\ 1 & 2 \end{bmatrix} . \begin{bmatrix} 1 \\ 5 \end{bmatrix} = \begin{bmatrix} 17 \\ 11 \end{bmatrix}$$

Dessa forma, caso um aluno queira avisar a um colega que a resposta da questão 1 é a letra d, ele deverá "passar cola" informando o vetor $\begin{bmatrix} 14 \\ 9 \end{bmatrix}$.

O discente que está "recebendo a cola", por sua vez, deverá proceder à decodificação da mensagem, a fim de descobrir a que questão se refere a resposta e, ainda, qual é a alternativa (**a**, **b**, **c**, **d** ou **e**) referente à resposta transmitida. Para tanto, uma maneira de decodificar a reposta recebida seria resolver a equação matricial A.X = B, haja vista que são conhecidas as matrizes A e B.

Por exemplo, para o aluno descobrir a que questão se refere e qual a alternativa indicada por meio do vetor $\begin{bmatrix} 14 \\ 9 \end{bmatrix}$, ele deveria resolver

$$\begin{bmatrix} 2 & 3 \\ 1 & 2 \end{bmatrix} . \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 14 \\ 9 \end{bmatrix},$$

que é equivalente a encontrar a solução do sistema

$$\begin{cases} 2x + 3y = 14 \\ x + 2y = 9 \end{cases}.$$

Todavia, existe um método mais rápido para decodificar uma resposta. O referido método consiste no cálculo da matriz coluna X por meio da solução da equação de matrizes a seguir:

$$A.X = B \leftrightarrow A^{-1}.A.X = A^{-1}.B \leftrightarrow I.X = A^{-1}.B \leftrightarrow X = A^{-1}.B.$$

Dessa maneira, podemos concluir que por meio do produto das matrizes $A^{-1}\ por\ B$ é possível obtermos o vetor do código fonte que permitirá a identificação do número da questão e da alternativa correspondente a ela.

É importante ressaltar que a equação matricial resolvida no parágrafo anterior só possui solução se a matriz A é invertível[14]. Podemos destacar uma maneira prática para obter a inversa de uma matriz 2 x 2, que é por meio da seguinte fórmula:

$$Se\ A = \begin{bmatrix} a & c \\ b & d \end{bmatrix}, então\ A^{-1} = \frac{1}{ad - bc} \begin{bmatrix} d & -c \\ -b & a \end{bmatrix}.$$

[14] Se $A \in M_n(\mathbb{R})$ é inversível, então existe A^{-1} tal que $A^{-1}.A = I$. Então, $\det(A^{-1}.A) = \det(I)$. Como $\det(A^{-1}.A) = \det(A^{-1}).\det(A)$ e det(I) = 1, então $\det(A^{-1}) = \frac{1}{\det(A)}$. Por isso, concluímos que para existir a inversão de uma matriz A, seu determinante não pode ser nulo.

Portanto, utilizando a mencionada fórmula, determinamos a inversa de A, que é $A^{-1} = \begin{bmatrix} 2 & -3 \\ -1 & 2 \end{bmatrix}$. Então, o aluno deve efetuar o produto $\begin{bmatrix} 2 & -3 \\ -1 & 2 \end{bmatrix} . \begin{bmatrix} 14 \\ 9 \end{bmatrix} = \begin{bmatrix} 1 \\ 4 \end{bmatrix}$ para obter a "cola" almejada. Ou seja, o educando irá decodificar o código de canal $\begin{bmatrix} 14 \\ 9 \end{bmatrix}$ para o código de fonte $\begin{bmatrix} 1 \\ 4 \end{bmatrix}$, o qual indica que a resposta se refere à primeira questão, alternativa d.

Estabelecida a forma de codificação e decodificação, o professor deve autorizar o início da prova. Cada um dos alunos, ao finalizar a prova, deverá escrever suas respostas no formato codificado pelo código de canal na folha de respostas. Após todos os alunos realizarem a prova, o professor deverá colocar as respostas no quadro, porém na forma codificada pelo canal, deixando que os alunos procedam à verificação delas.

Respostas:

$$\begin{bmatrix} 5 \\ 3 \end{bmatrix}, \begin{bmatrix} 13 \\ 8 \end{bmatrix}, \begin{bmatrix} 18 \\ 11 \end{bmatrix}, \begin{bmatrix} 17 \\ 10 \end{bmatrix}, \begin{bmatrix} 16 \\ 9 \end{bmatrix}, \begin{bmatrix} 24 \\ 14 \end{bmatrix}, \begin{bmatrix} 29 \\ 17 \end{bmatrix} e \begin{bmatrix} 28 \\ 16 \end{bmatrix}.$$

1a, 2c, 3d, 4c, 5b, 6d, 7e, 8d.

Procedimento pedagógico:

Esta atividade propõe que o aluno utilize um código de fonte e um código de canal para codificar e decodificar informações. Por meio dela se espera que os educandos entendam para que serve um código e o que motivou o estudo sobre os códigos corretores de erros.

Durante a realização da atividade, é possível que algum aluno reclame que recebeu uma resposta "inexistente". Caso isso não aconteça, o professor deve criar essa situação por meio de um exemplo de vetor que ao ser multiplicado pela matriz inversa não retorna resultados referentes à questão. O docente pode supor, por exemplo, que uma cola recebida tem a forma do vetor codificado pela matriz A = $\begin{bmatrix} 12 \\ 8 \end{bmatrix}$, e que ao proceder à decodificação foi encontrado o resultado $\begin{bmatrix} 2 & -3 \\ -1 & 2 \end{bmatrix} . \begin{bmatrix} 12 \\ 8 \end{bmatrix} = \begin{bmatrix} 0 \\ 4 \end{bmatrix}$, ou seja, ocorreu algum problema, pois não existe a questão zero. Nesse instante, o professor dirá que o algoritmo permite apenas a transmissão de informações, sem ser possível sua correção, e que esse foi o problema que motivou o estudo dos códigos corretores de erros. O professor pode aproveitar a oportunidade para fazer um resumo das condições históricas do surgimento dos códigos corretores de erros. Uma boa fonte de consulta é a Introdução do artigo intitulado "Breve introdução à Teoria dos Códigos Corretores de Erros", de autoria de César Polcino Milies, apresentado no Colóquio

de Matemática da Região Centro-Oeste, que foi realizado de 03 a 06 de novembro de 2009, e está disponível em http://www.sbm.org.br/docs/coloquios/CO-1-09.pdf. Além disso, a presente dissertação também é uma fonte de embasamento teórico sobre o assunto.

5.3.2 Atividade 2

Título: Os códigos corretores estão mais próximos de nós do que imaginamos

Material/Recurso necessário: computador, projetor multimídia, quadro negro (ou branco) e giz (ou caneta piloto), embalagens de produtos onde o código de barras esteja visível

Divisão da Turma: em alguns momentos haverá a formação de duplas ou grupos nessa atividade.

Tempo estimado: 100 minutos

Descrição sucinta:

Passo 1: Tem por objetivo despertar o interesse dos alunos para o estudo dos códigos corretores de erros. Para tanto, será apresentado o exemplo da língua portuguesa, que é um código presente no cotidiano dos educandos.

Passo 2: Apresentar um exemplo de outros códigos presentes no cotidiano dos educandos (UPC e EAN-13) e, também, mostrar que determinados algoritmos não são muito eficazes na detecção de erros.

Passo 3: Apresentar outro exemplo de códigos presentes no cotidiano, o qual possui um mecanismo mais eficaz de detecção do que o que foi exibido no passo 2.

Aspectos operacionais:

O professor deverá reservar com antecedência o computador e o projetor multimídia de sua unidade escolar.

Passo 1:

O docente deverá iniciar a atividade com a projeção do seguinte parágrafo por meio do projetor multimídia:

> “De aorcdo com uma pqsieusa de uma uinrvesriddae ignlsea, não ipomtra em qaul odrem as lrteas de uma plravaa etãso, a úncia csioa iprotmatne é que a piremria e útmlia lrteas etejasm no lgaur crteo. O rseto pdoe ser uma ttaol bçguana que vcoê pdoe anida ler sem pobrlmea. Itso é poqrue nós não lmeos cdaa lrtea isladoa, mas a plravaa cmoo um tdoo”

Fonte: Troca-letras tem fundamento científico.
Disponível em http://www.academus.pro.br/site/pg.asp? pagina=detalhe_variedade&titulo=Variedades &codigo=88&cod_categoria=&nome_categoria=

Ele pedirá, então, aos alunos que façam a leitura do texto. Espera-se que todos consigam ler o parágrafo.

Após a referida leitura, o professor apresentará no quadro palavras soltas, com erros, e pedirá que os alunos digam qual a palavra correta. A primeira sequência de palavras deve possuir uma verificação de erro e correção bem simples e direta. Por exemplo, “cathorro” e “telebisão”. A segunda sequência deve ser diferente. Por exemplo, “aato” e “aaca”. A palavra “aato” pode ser bato, mato, pato, gato, iato, etc. A palavra “aaca” pode ser vaca, maca, etc. Nesses últimos exemplos, é necessário que seja acrescentado um contexto, a fim de que a correção seja efetuada. O professor pode utilizar frases como “a aaca leiteira está em uma pastagem verdinha” ou “o rato bebe leite todo dia”. No primeiro caso, o erro está claro e a palavra correta é vaca. No segundo, aparentemente não há erro, pois rato é uma palavra da língua portuguesa. Todavia, rato não bebe leite, e sim o gato. Dessa forma, encontra-se o erro e é efetuada a correção.

Passo 2:

Terminada a primeira parte, o professor deve comentar com os alunos que existem outros códigos muito utilizados no cotidiano, citando o código de barras.

Inicialmente, o docente deverá explicar aos alunos como é formado o código de barras, dando-se ênfase à maneira como é obtido o dígito de verificação, pois é ele quem vai informar se houve ou não erro na digitação da sequência numérica referente ao código. Essa explicação pode ser apresentada com o uso do projetor multimídia ou com o quadro (negro ou branco).

Após, o docente deverá aplicar a atividade descrita abaixo, na qual alguns exercícios são adaptações do que foi apresentado nas dissertações de mestrado que tem por tema “Aritmética: código de barras e outras aplicações de congruências” apresentada por Josiane Colombo Pedrini Esquinca em 2013 (consulta por meio do site http://bit.profmat-

sbm.org.br/xmlui/handle/123456789/371) e “Propostas de Utilização de Códigos de Barras como Recurso Didático para o Ensino de Matemática” apresentada por Valeska Aparecida Rodrigues da Silva em 2013 (consulta por meio do site http://bit.profmat-sbm.org.br/xmlui/handle/123456789/141/browse?value=VALESKA+APARECIDA+RODRIGUES+DA+SILVA&type=author).

O professor, então, após a explicação do funcionamento dos códigos UPC e EAN-13, deve distribuir aos alunos uma folha de questões (folha de atividades 2) e pode sugerir que eles formem duplas a fim de resolver as questões. A folha de atividades 2 está no final deste capítulo.

Espera-se, ao final da atividade, que os alunos consigam entender que a construção de um bom código corretor de erros não é uma tarefa simples, uma vez que é possível verificar, por meio dos exercícios, que em alguns casos ocorreu erro, mas ele sequer foi detectado.

Informações complementares referentes ao passo 2:

Na década de 1970 começou a ser utilizado um código para identificação de produtos. Esse código foi elaborado por Geoge J. Laurer e ficou conhecido como UPC (Universal Product Code), sendo adotado nos Estados Unidos e no Canadá. Ele consistia de listras verticais alternadas, nas cores preta e branca, com um número indicado abaixo delas. A sequência dos números totalizava 12 dígitos. As larguras das barras verticais variam, podendo ser finas, médias, grossas ou muito grossas.

A interpretação das barras deve ser realizada em conformidade com a tabela abaixo.

Tabela 6

Listras	**Fina**	**Média**	**Grossa**	**Muito grossa**
Branca	0	00	000	0000
Preta	1	11	111	1111

Fonte: Códigos de Barras e outras Aplicações de Congruências.
Disponível em http://bit.profmat-sbm.org.br/xmlui/handle/123456789/371.

A leitura relativa à espessura e às corres das barras é realizada de forma que a cada quatro barras é obtida uma sequência de sete dígitos de zeros e uns. A referida sequência dá origem aos dígitos que aparecem abaixo das barras verticais, de acordo com a tabela a seguir.

Tabela 7

Dígito	Lado esquerdo	Lado direito
0	0001101	1110010
1	0011001	1100110
2	0010011	1101100
3	0111101	1000010
4	0100011	1011100
5	0110001	1001110
6	0101111	1010000
7	0111011	1000100
8	0110111	1001000
9	0001011	1110100

Fonte: A Matemática dos Códigos de Barras.
Disponível em http://mat.ufg.br/bienal/2006/mini/polcino.pdf

Realizada a identificação dos algarismos que aparecem abaixo das barras verticais, podemos enumerar seus significados. O primeiro algarismo indica o tipo de produto, a sequência que se segue de outros cinco determina o fabricante, a sequência seguinte de mais cinco informa a classificação do produto e o último é um dígito verificador.

Por exemplo:

Figura 2
Fonte: Figura adaptada de
https://sistemas.ufms.br/sigpos/portal/trabalhos/download/1131/cursold:148

O número 036000 (lado esquerdo) e 291452 (lado direito) corresponde à sequência 0001101 – 0111101 – 0101111 – 0001101 – 0001101 – 0001101 – 1101100 – 1110100 – 1110010 – 1011100 – 1001110 – 1101100.

A leitora do código consegue distinguir o lado direito do esquerdo, pois as sequências do lado esquerdo sempre têm um número ímpar de 1's ("uns") e as do direito um número par de 1's ("uns"). Dessa forma, é possível à máquina efetuar a leitura correta do código de barras, seja passado o leitor da direita para esquerda, seja da esquerda para direita.

Todavia, apesar de a máquina possuir essa capacidade de leitura, acontece, algumas vezes, de ela não conseguir realizar a identificação do produto. Pode haver alguma imperfeição na embalagem ou ela pode estar molhada, por exemplo. Nesse caso, é preciso que a sequência numérica abaixo das barras seja digitada. É nessa situação que é possível notar a importância do dígito verificador, pois ele permite a identificação de erros de digitação da referida sequência numérica.

A citada verificação se dá da seguinte forma. É calculado o produto escalar entre o vetor de pesos (3, 1, 3, 1, 3, 1, 3, 1, 3, 1, 3, 1), que é fixo, e o vetor construído por meio da sequência dos doze dígitos do código. O resultado dessa conta deve ser um múltiplo de dez. Caso isso não aconteça, é certo que ocorreu algum erro e a máquina informa isso na tela do computador.

Vamos exemplificar essa situação por meio da figura de código de barras da figura 2. Para isso, devemos tomar o vetor (3, 1, 3, 1, 3, 1, 3, 1, 3, 1, 3, 1) e fazer o produto escalar com o vetor relativo à sequência numérica do código de barras (0, 3, 6, 0, 0, 0, 2, 9, 1, 4, 5, X). O resultado desta conta deve ser um múltiplo de 10. No caso do nosso exemplo, obtemos: $3.0 + 1.3 + 3.6 + 1.0 + 3.0 + 1.0 + 3.2 + 1.9 + 3.1 + 1.4 + 3.5 + 1x = 58 + x$. Para achar o valor de x, então, precisamos pensar que ele é o menor valor inteiro não negativo que quando somado a 58 resulta no múltiplo de 10 mais próximo de 58, ou seja, devemos fazer $x = 2$, que é exatamente o dígito de verificação que aparece na figura.

O código descrito acima (UPC) foi alterado em 1976 por Laurer, a fim de ser realizado o acréscimo de um dígito. O código passou então a possuir treze dígitos e recebeu o nome de EAN-13 (European Article Numbering system). Todavia, a mudança foi realizada de forma que o leitor de barras pudesse identificar tanto os códigos UPC quanto os códigos EAN-13. Para isso, foi acrescido um zero antes do código UPC, sendo mantidas as demais informações. Dessa maneira, EUA e Canadá são identificados por um zero como o primeiro número do código. Nos demais países, a identificação é realizada por meio dos três primeiros dígitos, conforme descrito na figura abaixo.

Figura 3
Fonte: Aritmética: Códigos de Barras e outras Aplicações de Congruências.
Disponível em http://bit.profmat-sbm.org.br/xmlui/handle/123456789/371. Figura adaptada.

As listras verticais alternadas, nas cores preta e branca, seguem a mesma codificação que o código UPC.

Tabela 8

Listras	Fina	Média	Grossa	Muito grossa
Branca	0	00	000	0000
Preta	1	11	111	1111

Fonte: Aritmética: Códigos de Barras e outras Aplicações de Congruências.
Disponível em http://bit.profmat-sbm.org.br/xmlui/handle/123456789/371.

Também é mantida a leitura de forma semelhante, pois a cada quatro barras verticais é associada uma sequência de sete dígitos de 0's ("zeros") e 1's ("uns"), e um número é indicado abaixo das barras verticais. Todavia, a tabela em que são mostrados os dígitos correspondentes a cada um dos blocos de 0's ("zeros") e 1's ("uns") é a seguinte:

Tabela 9

Dígito	Lado esquerdo (ímpar)	Lado esquerdo (par)	Lado direito
0	0001101	0100111	1110010
1	0011001	0110011	1100110
2	0010011	0011011	1101100
3	0111101	0100001	1000010
4	0100011	0011101	1011100
5	0110001	0111001	1001110
6	0101111	0000101	1010000
7	0111011	0010001	1000100
8	0110111	0001001	1001000
9	0001011	0010111	1110100

Fonte: A Matemática dos Códigos de Barras.
Disponível em http://mat.ufg.br/bienal/2006/mini/polcino.pdf

Podemos verificar que a codificação do lado direito é idêntica ao sistema UPC (número par de uns). Todavia, a codificação do lado esquerdo foi alterada. Ela agora pode começar com um número par ou ímpar de dígitos iguais a 1, conforme a tabela apresentada acima.

Falta apenas a indicação do dígito inicial. Ele é escolhido por meio de uma alternância entre a quantidade de dígitos pares ou ímpares, conforme abaixo.

Tabela 10 (continua)

Dígito inicial	1º	2º	3º	4º	5º	6º
0	Ímpar	Ímpar	Ímpar	Ímpar	Ímpar	Ímpar
1	Ímpar	Ímpar	Par	Ímpar	Par	Par
2	Ímpar	Ímpar	Par	Par	Ímpar	Par
3	Ímpar	Ímpar	Par	Par	Par	Ímpar
4	Ímpar	Par	Ímpar	Ímpar	Par	Par
5	Ímpar	Par	Par	Ímpar	Ímpar	Par
6	Ímpar	Par	Par	Par	Ímpar	Ímpar
7	Ímpar	Par	Ímpar	Par	Ímpar	Par
8	Ímpar	Par	Ímpar	Par	Par	Ímpar
9	Ímpar	Par	Par	Ímpar	Par	Ímpar

Fonte: A Matemática dos Códigos de Barras.
Disponível em http://mat.ufg.br/bienal/2006/mini/polcino.pdf

Assim, a partir das informações acima podemos fazer um exemplo. Tomemos o código de barras da página anterior (4-891668-326689). O código é iniciado pelo número quatro. Então a sequência do lado esquerdo é ímpar – par – ímpar – ímpar – par – par. Desse modo, a correspondência obtida da tabela é 8 → 0110111; 9 → 0010111, 1 → 0011001, 6 → 0101111, 6 → 0000101, 8 → 0001001. A sequência do lado direito é 3 → 1000010, 2 → 1101100, 6 → 1010000, 6 → 1010000, 8 → 1001000 e 9 → 1110100.

É importante ressaltar que o leitor de barras consegue identificar corretamente a sequência numérica, mesmo que a leitura aconteça de forma invertida, pois a sequência da direita sempre começa com o algarismo um e a sequência da esquerda sempre começa com zero.

De forma semelhante ao realizado no código UPC, pode acontecer de leitor não conseguir realizar a identificação do produto, sendo necessária a digitação da sequência numérica. Nesse caso, o dígito verificador, como nos códigos UPC, demonstrará sua importância, pois ele permitirá a identificação de erros de digitação da referida sequência numérica.

A citada verificação se dará da seguinte maneira. É calculado o produto escalar entre o vetor de pesos (1, 3, 1, 3, 1, 3, 1, 3, 1, 3, 1, 3, 1), que é fixo, e o vetor construído por meio da sequência dos treze dígitos do código. O resultado dessa conta deve ser um múltiplo de dez. Se isto não acontecer, ocorreu algum erro e a máquina informa isso na tela do computador.

Vamos exemplificar a situação descrita no parágrafo anterior utilizando o código de barras que tem por numeração a sequência 4-891668-326689. Para isso, vamos tomar o vetor (1, 3, 1, 3, 1, 3, 1, 3, 1, 3, 1, 3, 1) e fazer o produto escalar com o vetor relativo à sequência numérica do código de barras (4, 8, 9, 1, 6, 6, 8, 3, 2, 6, 6, 8, X). O resultado desta conta deve ser um múltiplo de 10. No nosso exemplo, obtemos: 1.4 + 3.8 + 1.9 + 3.1 + 1.6 + 3.6 + 1.8 + 3.3 + 1.2 + 3.6 + 1.6 + 3.8 + 1x = 131 + x. Para achar o valor de x, então, devemos somar o menor valor inteiro não negativo que transforme 131 no múltiplo de 10 mais próximo de 131, ou seja, devemos fazer x = 9, que é exatamente o dígito de verificação que aparece no código.

Passo 3:

O professor vai citar outro código presente no cotidiano: o código do cadastro de pessoas física (CPF).

Inicialmente, o docente irá explicar que o referido código é composto de onze dígitos, em que os nove primeiros são de identificação e os dois últimos são dígitos verificadores. Em

seguida, o professor deve apresentar o algoritmo que permite a detecção de que ocorreu algum erro na digitação do número do CPF.

Verifiquemos um exemplo. Vamos examinar a autenticidade do CPF 043.658.306-27. Para isso, primeiramente, devemos calcular o resultado: **1**.0 + **2**.4 + **3**.3 + **4**.6 + **5**.5 + **6**.8 + **7**.3 + **8**.0 + **9**.6 = 189. Em seguida, dividimos 189 por 11. O resto da mencionada divisão é 2. Após, tomamos o resto encontrado, que agora passa a ser o dividendo, e dividimos por 10. O novo resto encontrado também é 2. Dessa forma, encontramos o primeiro dígito de controle, que é 2, idêntico ao informado no CPF 043.658.306-27. Agora, realizaremos a outra verificação, correspondente ao segundo dígito de controle. Para isso, vamos calcular, inicialmente, o resultado: **1**.4 + **2**.3 + **3**.6 + **4**.5 + **5**.8 + **6**.3 + **7**.0 + **8**.6 + **9**.2 = 172. Em seguida, dividimos 172 por 11. O resto da mencionada divisão é 7. Após, tomamos o resto encontrado, que agora passa a ser o dividendo, e dividimos por 10. O novo resto encontrado também é 7. Ou seja, o segundo dígito de controle é 7, confirmando a autenticidade do CPF 043.658.306-27.

O professor proporá aos educandos que verifiquem a autenticidade de seus respectivos números de CPF. Caso algum aluno não tenha ou não lembre de seu CPF, ele pode realizar a tarefa junto a um colega.

Informações complementares referentes ao passo 3:

O número do CPF de um indivíduo tem a forma $a_1a_2a_3a_4a_5a_6a_7a_8a_9a_{10}a_{11}$, em que cada um dos a_i para $1 \leq i \leq 9$ é um número de identificação, a_{10} e a_{11} são dígitos de controle, e os a_{is} são números não negativos de um algarismo.

O seguinte algoritmo pode ser utilizado para calcular os dígitos de controle.

$$a_{10} = \left(\sum_{i=1}^{9} i \,.\, x_i \ (mod\ 11) \right) (mod\ 10) \quad e$$

$$a_{11} = \left(\sum_{i=2}^{10} (i-1) \,.\, x_i \ (mod\ 11) \right) (mod\ 10).$$

Mais informações sobre esse assunto podem ser obtidas por meio do Trabalho de Conclusão de Curso do Mestrado Profissional em Matemática de autoria de Fernanda Rodrigues Alves Costa e Marcelo Oliveira Veloso, disponível em http://bit.profmat-sbm.org.br/xmlui/handle/123456789/1166.

Procedimento pedagógico

O passo 2 é com certeza o mais complicado para o aluno entender. Então, é necessária a utilização de mais tempo na explicação e realização de exemplos nesse passo. Para isso, sugerimos a utilização de mais exemplos para o melhor aprendizado deste conteúdo, podendo o professor trazer mais alguns códigos de barras de produtos de sua casa.

É importante o professor salientar que o erro, se existir, em diversas hipóteses[15] pode ser identificado (evita que uma pessoa compre um produto e pague por outro), mas em nenhuma delas pode ser corrigido, isto é, o código do produto terá que ser passado novamente pelo leitor ou digitado manualmente.

Abaixo sugerimos outras possibilidades de passos para a atividade 2, que deverá ser utilizada caso os alunos tenham demonstrado boa compreensão do conteúdo e interesse pelo assunto.

Possibilidade de outros passos na atividade 2:

O professor pode se utilizar de mais exemplos semelhantes aos apresentados nos passos 2 e 3. Para tanto, basta utilizar como referência de pesquisa a dissertação de mestrado "Uma análise dos esquemas de dígitos verificadores usados no Brasil" apresentada por Natália Pedroza de Souza em 2013, no Centro de Tecnologia e Ciência do Instituto de Matemática e Estatística da UERJ. O arquivo está disponível em http://www.bdtd.uerj.br/tde_busca/arquivo.php?codArquivo=6100. Inclusive, na elencada dissertação, é apresentada a explicação do motivo pelo qual o CPF é um código melhor que os códigos de barra, pois a probabilidade de não detecção de erros neste último é muito menor.

5.3.3 Atividade 3

Título: "Desarmando uma bomba"

Material/Recurso necessário: computador, projetor multimídia, quadro negro (ou branco) e giz (ou caneta piloto).

[15] Em algumas hipóteses o erro sequer é identificado. Para mais informações, deve ser consultada a referência bibliográfica [27].

Divisão da Turma: Em alguns momentos da realização da atividade haverá divisão da turma em grupos de pelo menos 3 alunos.

Tempo estimado: 100 minutos

Descrição sucinta:

Construção de um código capaz de detectar dois erros e corrigir até um. O algoritmo será construído supondo-se que não houve uma avaliação prévia da quantidade de erros que acontecem quando a informação é transmitida pelo canal. A consequência disso é que em algumas situações não é possível a correção dos erros.

A atividade é importante para demonstrar que o código deve ser construído após uma avaliação do canal, a fim de que a quantidade de redundâncias acrescentadas seja suficiente para que ocorra a correção da informação com certo grau de certeza.

Nesse exemplo não será utilizado um código linear.

Aspectos operacionais:

O professor deverá explicar aos educandos quais os primeiros elementos necessários para a criação de um código: alfabeto e comprimento das palavras. Em seguida, definirá o conjunto $\mathbb{F}_q^n$, exemplificando (pode ser utilizado como alfabeto o conjunto $\mathbb{F}_2 = \{0, 1\}$ e analisados os conjuntos $\mathbb{F}_2^3$, $\mathbb{F}_2^4$ e $\mathbb{F}_2^5$). Após, o docente explicará aos alunos que um código pode ser um subconjunto qualquer de $\mathbb{F}_q^n$.

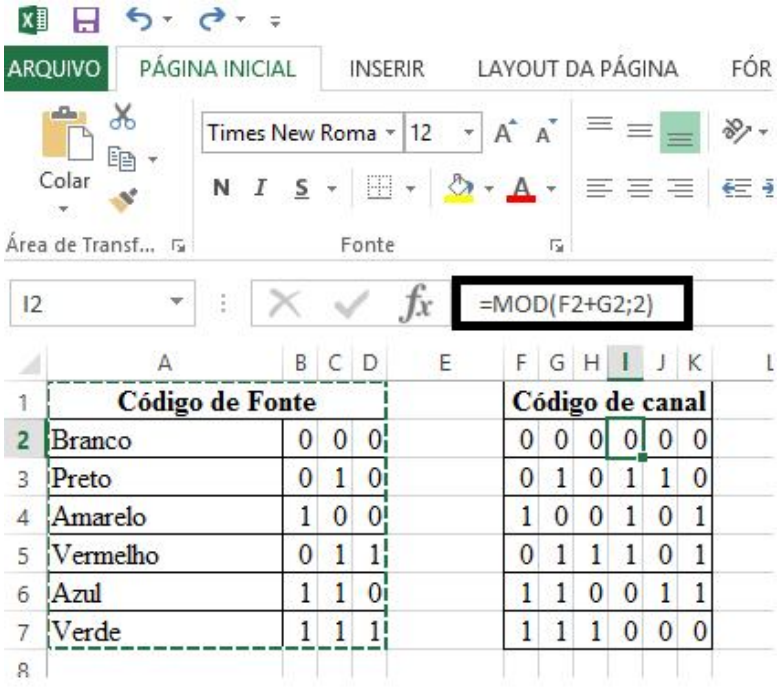
Esclarecidos os pontos acima, o docente proporá aos estudantes a construção de uma codificação para desarmar a bomba. Para tanto, pode-se supor que a bomba tenha fios das seguintes cores: branco, preto, amarelo, vermelho, azul e verde. Uma possível codificação para a fonte está na tabela abaixo:

Tabela 11

Código de Fonte	
Branco	0 0 0
Preto	0 1 0
Amarelo	1 0 0
Vermelho	0 1 1
Azul	1 1 0
Verde	1 1 1

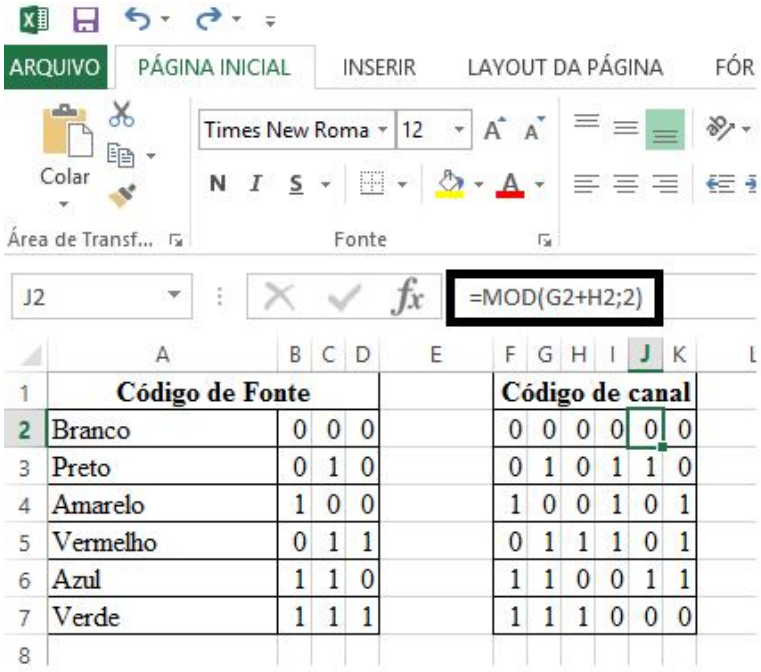
Em seguida, escolhe-se a quantidade de dígitos que serão acrescidos à informação – as chamadas redundâncias –, a fim de que seja possível a identificação e correção de erros.

Suponhamos que sejam acrescentados 3 (três) dígitos, segundo a seguinte regra: (x, y, z, x + y, y + z, x + z). Utilizando-se de uma planilha eletrônica, podemos apenas preencher o valor do código da fonte e colocar as seguintes fórmulas para obter as palavras do código:



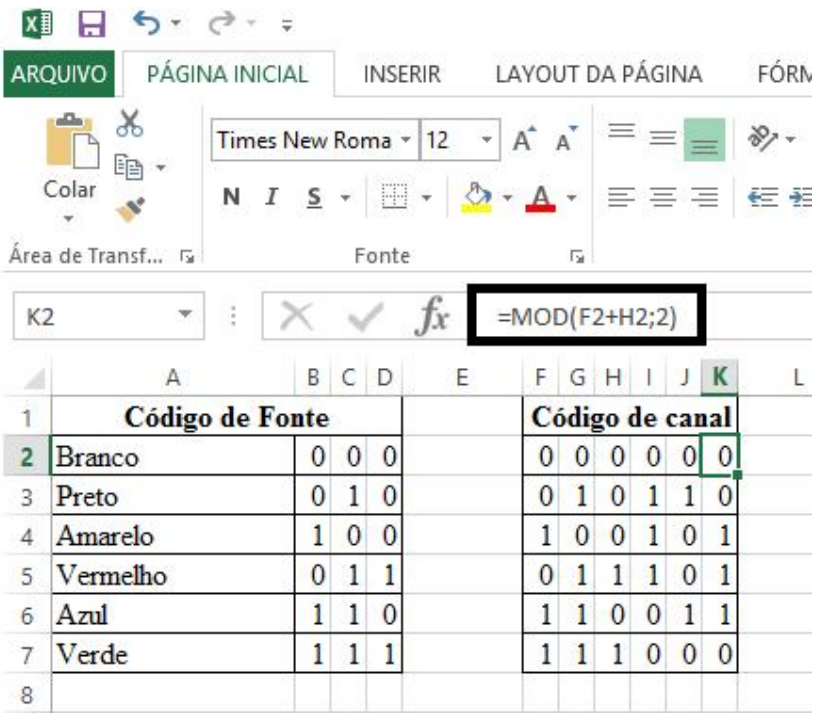
	A	B	C	D	E	F	G	H	I	J	K
1	**Código de Fonte**					**Código de canal**					
2	Branco	0	0	0		0	0	0	0	0	0
3	Preto	0	1	0		0	1	0	1	1	0
4	Amarelo	1	0	0		1	0	0	1	0	1
5	Vermelho	0	1	1		0	1	1	1	0	1
6	Azul	1	1	0		1	1	0	0	1	1
7	Verde	1	1	1		1	1	1	0	0	0
8											

Figura 4



	A	B	C	D	E	F	G	H	I	J	K
1	**Código de Fonte**					**Código de canal**					
2	Branco	0	0	0		0	0	0	0	0	0
3	Preto	0	1	0		0	1	0	1	1	0
4	Amarelo	1	0	0		1	0	0	1	0	1
5	Vermelho	0	1	1		0	1	1	1	0	1
6	Azul	1	1	0		1	1	0	0	1	1
7	Verde	1	1	1		1	1	1	0	0	0
8											

Figura 5



	A	B	C	D	E	F	G	H	I	J	K	L
1	**Código de Fonte**					**Código de canal**						
2	Branco	0	0	0		0	0	0	0	0	0	
3	Preto	0	1	0		0	1	0	1	1	0	
4	Amarelo	1	0	0		1	0	0	1	0	1	
5	Vermelho	0	1	1		0	1	1	1	0	1	
6	Azul	1	1	0		1	1	0	0	1	1	
7	Verde	1	1	1		1	1	1	0	0	0	
8												

Figura 6

A função “MOD(a;b)”, nas planilhas eletrônicas, informa o resto da divisão do número a pelo número b. No caso, como o alfabeto é {0, 1}, está sendo utilizado o resto da divisão por 2.

Assim, para preencher o código de canal, inicialmente repetiremos o código de fonte no intervalo F2:H7. Após, vamos colar as fórmulas indicadas nas figuras 4,5 e 6 em todas as células do intervalo I3:K7. Dessa forma, será construído o código de canal abaixo.

Tabela 12

Código de canal					
0	0	0	0	0	0
0	1	0	1	1	0
1	0	0	1	0	1
0	1	1	1	0	1
1	1	0	0	1	1
1	1	1	0	0	0

Elaborado o código, o professor continuará explicando aos educandos mais itens necessários à compreensão do funcionamento de um algoritmo de detecção e correção de erros. Para isso, ele apresentará o que é a distância entre as palavras de um código, bem como

qual o procedimento para obter a distância mínima e identificar a quantidade de erros que podem ser corrigidos (definição 2.3, definição 2.6, definição 2.7 e teorema 2.3).

Apresentados os conceitos acima, o docente procederá ao cálculo das distâncias entre todas as palavras do código que foi criado. Portanto, serão calculadas $\binom{6}{2} = \frac{6!}{2!.4!} = 15$ distâncias.

Utilizando-se de uma planilha eletrônica[16]:

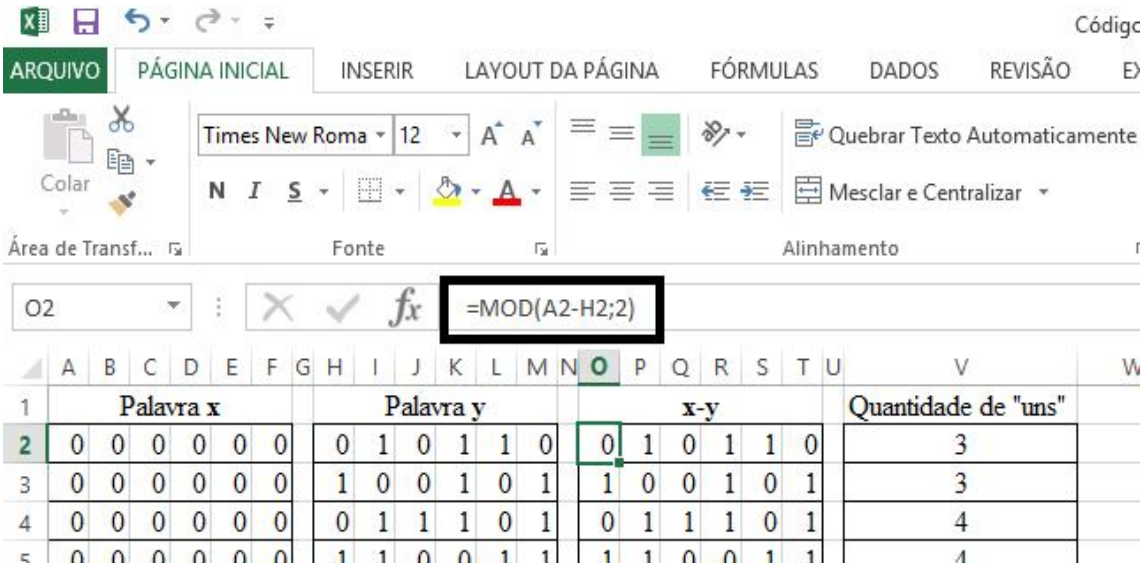


Figura 7

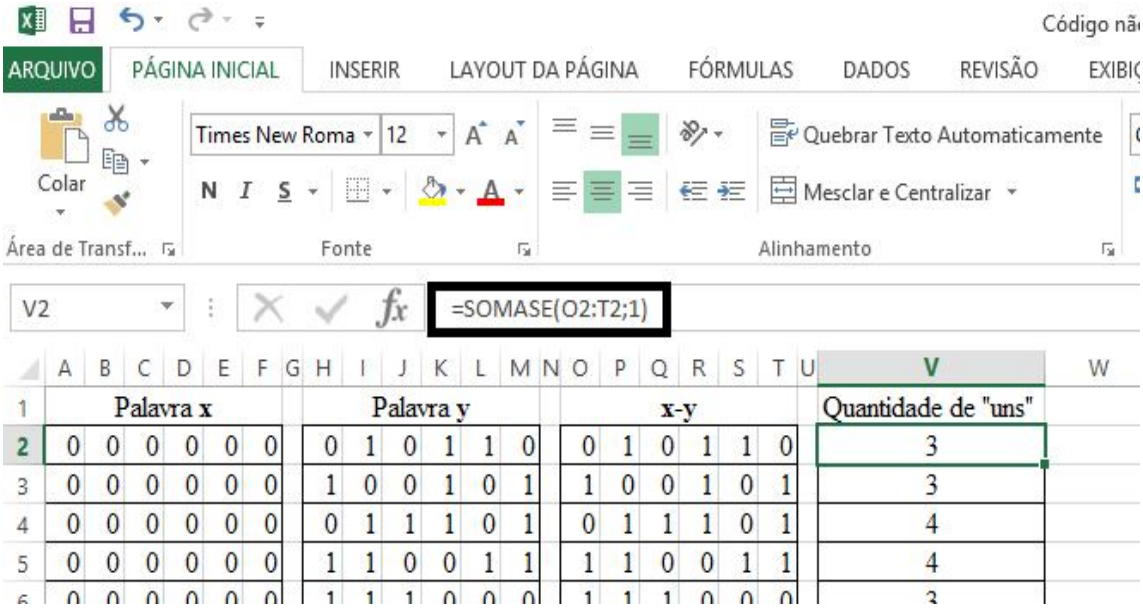


Figura 8

[16] Observemos que a função "SOMASE(intervalo; critérios; intervalo_soma)", nas planilhas eletrônicas, faz a adição das células especificadas por um determinado critério ou condição.

podemos construir a tabela a seguir,

Tabela 13

Palavra **x**	Palavra **y**	**x-y**	Quantidade de "1's"
0 0 0 0 0 0	0 1 0 1 1 0	0 1 0 1 1 0	3
0 0 0 0 0 0	1 0 0 1 0 1	1 0 0 1 0 1	3
0 0 0 0 0 0	0 1 1 1 0 1	0 1 1 1 0 1	4
0 0 0 0 0 0	1 1 0 0 1 1	1 1 0 0 1 1	4
0 0 0 0 0 0	1 1 1 0 0 0	1 1 1 0 0 0	3
0 1 0 1 1 0	1 0 0 1 0 1	1 1 0 0 1 1	4
0 1 0 1 1 0	0 1 1 1 0 1	0 0 1 0 1 1	3
0 1 0 1 1 0	1 1 0 0 1 1	1 0 0 1 0 1	3
0 1 0 1 1 0	1 1 1 0 0 0	1 0 1 1 1 0	4
1 0 0 1 0 1	0 1 1 1 0 1	1 1 1 0 0 0	3
1 0 0 1 0 1	1 1 0 0 1 1	0 1 0 1 1 0	3
1 0 0 1 0 1	1 1 1 0 0 0	0 1 1 1 0 1	4
0 1 1 1 0 1	1 1 0 0 1 1	1 0 1 1 1 0	4
0 1 1 1 0 1	1 1 1 0 0 0	1 0 0 1 0 1	3
1 1 0 0 1 1	1 1 1 0 0 0	0 0 1 0 1 1	3

e, então, concluir que a distância mínima é **d** = 3 e que o código é capaz de identificar até dois erros. Além disso, como $\boldsymbol{\kappa} = \left[\frac{d-1}{2}\right] = \left[\frac{3-1}{2}\right] = 1$, o código pode corrigir 1 erro.

Em seguida, o professor deve explicar aos alunos mais um conceito, o de disco com centro em uma palavra **c** do código e raio **r**. Após isso, será possível a construção dos discos de raio 1 para o código exemplificado, conforme abaixo:

D (000000, 1) = {000000, 100000, 010000, 001000, 000100, 000010, 000001}

D (010110, 1) = {010110, 110110, 000110, 011110, 010010, 010100, 010111}

D (100101, 1) = {100101, 000101, 110101, 101101, 100001, 100111, 100100}

D (011101, 1) = {011101, 111101, 001101, 010101, 011001, 011111, 011100}

D (110011, 1) = {110011, 010011, 100011, 111011, 110111, 110001, 110010}

D (111000, 1) = {111000, 011000, 101000, 110000, 111100, 111010, 111001}

Finalizando as explicações referentes à forma de correção e decodificação de mensagens, o docente irá comentar com os alunos que a correção de um erro é realizada por meio da substituição da palavra errada por aquela que lhe é mais próxima, ou seja, basta identificar em que disco está a palavra e trocá-la pela palavra do código que foi utilizada para construir o disco.

É importante ressaltar que o código foi construído para efetuar a correção de até um erro. Todavia, não sabemos a quantidade de erros que ocorre na transmissão. A atividade permitirá a verificação da importância de saber, previamente, a quantidade de erros que acontecem quando a informação é transmitida pelo canal.

Cumpridos os itens anteriores, o docente pode iniciar a atividade. Ela consiste na seguinte situação. Considera-se que os alunos fazem parte da equipe de ações táticas da polícia de um país e que uma denúncia foi recebida de que há uma bomba na escola. Um membro da equipe, de posse de uma câmera, procede ao local da bomba e envia uma imagem do objeto ao quartel. Lá, a imagem da bomba é comparada com outros padrões e é identificada a forma de desarmá-la. Todavia, antes de ser enviada pelo quartel a sequência de fios que deve ser cortada, ocorre uma pane nos equipamentos eletrônicos e o policial não consegue mais receber orientações da base. No entanto, ele sabe que a base continua a ouvi-lo, pois a central eletrônica referente ao envio do sinal sonoro continua ativa.

O policial, então, procura em sua mochila outro objeto para se comunicar com a base e apenas encontra um aparelho mais antigo, que transmite sequências de seis dígitos de 0's ("zeros") e 1's ("uns"). Ele decide utilizar o equipamento e, como o aparelho havia sido substituído por causa dos erros que ocorriam na transmissão, o policial cria um código capaz de corrigir até um erro (já foi exemplificado anteriormente[17]) e solicita à base que envie as informações para ele pelo equipamento antigo. O problema é que o policial não sabe a quantidade de erros que ocorrem quando uma informação é recebida pelo aparelho. Ele deverá avaliar cada situação.

Após a apresentação acima, em que a situação-problema é proposta aos grupos, o professor assume o papel da "base", e começa a enviar sequências de dígitos a cada um dos grupos, a qual contém a sequência dos fios a serem cortados.

Inicialmente, o docente deve enviar sequências que podem ser corrigidas pelo código. Por exemplo, se para desarmar a bomba devem ser cortados os fios verde, vermelho e branco, a codificação sem erros é 111000, 011101 e 000000, respectivamente. Todavia, como será considerado que ocorre um erro na transmissão, o professor irá informar uma sequência numérica diferente. Suponhamos que ela seja a seguinte: 111001, 001101 e 000100, respectivamente.

Os alunos, então, devem procurar as referidas palavras nos discos de raio 1 (um):

D (111000, 1) = {111000, 011000, 101000, 110000, 111100, 111010, 111001}

[17] Uma palavra com código fonte (x, y, z) é codificada para uma palavra de código de canal por meio do vetor (x, y, z, x + y, y + z, x + z). Os caracteres x, y, z são elementos do conjunto $\mathbb{F}_2 = \{0, 1\}$.

D (011101, 1) = {011101, 111101, 001101, 010101, 011001, 011111, 011100}

D (000000, 1) = {000000, 100000, 010000, 001000, 000100, 000010, 000001}

Dessa forma, como foi possível identificar as palavras nos discos acima, podemos fazer a substituição delas pelas palavras que dão origem aos discos, ou seja, 111001 por 111000, 01101 por 011101 e 000100 por 00000. Esse é caso em que, aparentemente, é possível a correção da mensagem.

Após o envio pelo docente de algumas mensagens e a consequente decodificação por cada um dos grupos, o professor poderá enviar uma mensagem igual a todos os alunos. Todavia, essa mensagem deve conter dois erros. Suponhamos, utilizando a mesma sequência do parágrafo anterior – verde, vermelho e branco – que seja recebida a sequência 101100, 001100 e 001010. Os alunos deverão proceder da mesma maneira que estavam fazendo antes: procurando as palavras nos discos. Todavia, eles não encontrarão em disco algum as palavras fornecidas, e começarão a questionar o professor. Nesse momento, o docente, com o uso do projetor multimídia e de planilhas eletrônicas, explicará aos educandos que ocorreram dois erros na transmissão, ou seja, eles devem procurar outra forma para descobrir a palavra correta. Ele sugere, então, que seja procurada a palavra do código mais próxima e, para isso, calcula as distâncias abaixo:

Tabela 14

Palavra **x**	Palavra **y**	**x-y**	Quantidade de "uns"
0 0 0 0 0 0	1 0 1 1 0 0	1 0 1 1 0 0	3
0 1 0 1 1 0	1 0 1 1 0 0	1 1 1 0 1 0	4
1 0 0 1 0 1	1 0 1 1 0 0	0 0 1 0 0 1	2
0 1 1 1 0 1	1 0 1 1 0 0	1 1 0 0 0 1	3
1 1 0 0 1 1	1 0 1 1 0 0	0 1 1 1 1 1	5
1 1 1 0 0 0	1 0 1 1 0 0	0 1 0 1 0 0	2
0 0 0 0 0 0	0 0 1 1 0 0	0 0 1 1 0 0	2
0 1 0 1 1 0	0 0 1 1 0 0	0 1 1 0 1 0	3
1 0 0 1 0 1	0 0 1 1 0 0	1 0 1 0 0 1	3
0 1 1 1 0 1	0 0 1 1 0 0	0 1 0 0 0 1	2
1 1 0 0 1 1	0 0 1 1 0 0	1 1 1 1 1 1	6
1 1 1 0 0 0	0 0 1 1 0 0	1 1 0 1 0 0	3
0 0 0 0 0 0	0 0 1 0 1 0	0 0 1 0 1 0	2
0 1 0 1 1 0	0 0 1 0 1 0	0 1 1 1 0 0	3
1 0 0 1 0 1	0 0 1 0 1 0	1 0 1 1 1 1	5
0 1 1 1 0 1	0 0 1 0 1 0	0 1 0 1 1 1	4
1 1 0 0 1 1	0 0 1 0 1 0	1 1 1 0 0 1	4
1 1 1 0 0 0	0 0 1 0 1 0	1 1 0 0 1 0	3

Dessa forma, os educandos poderão concluir há duas palavras mais próximas de 101100, que são 100101 (amarelo) e 111000 (verde); há duas palavras mais próximas de 001100, que são 000000 (branco) e 011101 (vermelho); e uma palavra mais próxima de 001010 que é 000000 (branco).

Como a sequência que o professor passou é verde, vermelho e branco, apenas uma palavra foi identificada. As outras duas serão escolhidas ao acaso. Isso significa que a bomba pode explodir. Além disso, é exemplificada a tese de que o código consegue identificar a ocorrência de dois erros na transmissão e que a correção fica prejudicada quando ocorre mais de um erro.

Após essa explicação, o professor propõe outro exemplo. Ele irá considerar que será recebida a sequência 101011, 110100 e 001101 para a mensagem verde, vermelho e branco (a mensagem sem erros é 111000, 011101 e 000000, respectivamente). Ou seja, a mensagem será recebida com 3 erros. Os educandos, então, devem procurar nos discos de raio 1 (um) as palavras informadas. Verificar-se-á que apenas uma será encontrada:

D (011101, 1) = {011101, 111101, 001101, 010101, 011001, 011111, 011100}

O fato de essa palavra ter sido encontrada em um disco traz como consequência a falsa ideia de que a sequência 001101 deve ser corrigida para 011101, o que não é verdade. Nesse caso, a bomba também explode. Assim, concluímos que quando ocorreram três erros o código além de não ser capaz de identificar o erro, também nos induz a uma falsa correção.

Assim, para que não ocorram problemas semelhantes a esses, o canal deve ser avaliado previamente a fim de que haja a identificação da quantidade de erros que ocorrem na transmissão para, posteriormente, ser construído um código capaz de corrigir com maior grau de certeza a informação recebida.

Procedimento pedagógico

Essa é a primeira atividade em que os alunos construirão um código que seja capaz de corrigir erros. Para isso, será necessário que o docente explique vários conceitos – alfabeto, comprimento, distância entre palavras, distância mínima, parâmetro **κ**, disco de centro em uma palavra **c** do código e raio **r**, além do critério de correção por meio da palavra mais próxima. Ressalta-se que este código não será um código linear, pois ele não é um subespaço vetorial de $\mathbb{F}_2^6$, haja vista que se efetuarmos a soma das palavras do código 010110 e 011101,

obteremos 001011, que é uma palavra que não pertence ao código (definição 3.3, subespaços vetoriais).

Esse código reforça a ideia de que é importante que saibamos a quantidade de erros que ocorrem na transmissão das informações, sendo que este dado é obtido estatisticamente. O aluno deverá concluir, ao fim da atividade, que não basta estabelecer um código de maneira aleatória, mas que é preciso construir um algoritmo que tenha realmente a capacidade de codificar e decodificar as informações com um certo grau de certeza. E isso não acontece na questão. Portanto, a bomba pode explodir.

5.3.4 Atividade 4

Título: O “jogo da velha”

Material/Recurso necessário: computador, projetor multimídia, quadro negro (ou branco) e giz (ou caneta piloto)

Divisão da Turma: Inicialmente, para os procedimentos de elaboração do código, não haverá divisão da turma. Posteriormente, para realização do jogo, a turma será dividida em dois grupos e, estes grupos, também serão divididos em duas equipes, em que uma será responsável pela codificação das jogadas e outra pela decodificação.

Tempo estimado: 120 minutos

Descrição sucinta:

Construção de um código linear perfeito capaz de detectar dois erros e corrigir um erro. Nesta atividade vamos supor que houve uma verificação estatística e que o canal ocasiona até um erro na transmissão da informação. A partir disso, construiremos um código perfeito que seja capaz de corrigir os erros.

Aspectos operacionais:

O professor deverá recordar com os alunos os conceitos apresentados na última atividade. Em seguida, deverá explicar que existem códigos chamados de lineares (definição 3.7). Mas, para que haja um bom entendimento dos referidos códigos, o docente deverá esclarecer, de forma sucinta, o que é um subespaço vetorial. Inclusive, o professor pode citar

que o código exemplificado na atividade anterior não era subespaço de $\mathbb{F}_2^6$, pois a soma das palavras 010110 e 011101, pertencentes ao código, é 001011, que não pertence ao código.

Após essa introdução, o professor irá explicar aos alunos que o procedimento a ser adotado nessa atividade será diferente da anterior. Inicialmente, saberemos a quantidade de erros que ocorrem na transmissão. Então, a partir dessa informação, construiremos um código linear capaz de detectar e corrigir os erros.

Antes de iniciar a construção de um código de canal linear perfeito, devemos construir um código de fonte. Para isso, o professor deverá explicar à turma como se joga o jogo da velha (caso o professor não conheça o jogo da velha, ele poderá consultar o link http://pt.wikipedia.org/wiki/Jogo_da_velha para esclarecimentos) e compará-lo com uma matriz 3 x 3:

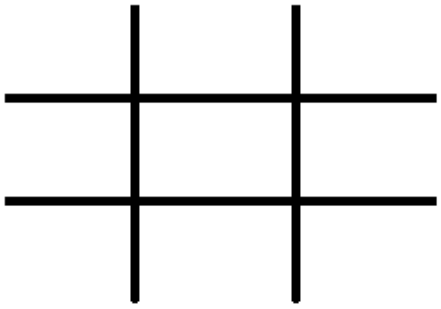


Figura 9

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$

Em seguida, o docente deve propor a criação de um código de fonte que seja capaz de indicar em que posição deve ser realizada uma jogada, ou seja, onde deve ser colocado X ou O. Para tanto, ele deve estar atento ao fato de que são necessárias pelo menos 9 palavras para elaboração de um código que atenda ao referido propósito, já que o tabuleiro tem 9 posições. Portanto, o professor pode sugerir que código a ser criado tenha exatamente **M** = 9 palavras. Em seguida, observando que $3^2 = 9$, o professor deve recomendar que seja utilizado como alfabeto o conjunto $\mathbb{F}_3 = \{0, 1, 2\}$, a fim de que o código construído seja linear ($q^m = 3^2 = 9$; vide definição 3.7).

Suponhamos que o código de fonte esteja em conformidade com o seguinte algoritmo: $a_{ij} = (i-1)(j-1)$. Aplicando a fórmula, cada casa do tabuleiro corresponderá à codificação abaixo:

00	01	02
10	11	12
20	21	22

Figura 10

Apresentado o código da fonte, o professor exemplificará aos alunos que por meio do referido código não são possíveis a identificação e a correção de erros de transmissão. Por exemplo, se 00 for transmito e for recebido como 01, não há como descobrir que aconteceu um erro – as palavras são muito próximas umas das outras. Diante disso, o docente deve explicar aos educandos que existe a necessidade de acrescentar dígitos à informação que está sendo transmitida (as chamadas redundâncias), o que caracteriza a criação de um código de canal. Ele, então, pode sugerir aos alunos o acréscimo de dois dígitos ao código de fonte, o que fará com o que código de canal[18] tenha comprimento **n** = 4.

O professor, após o estabelecimento do comprimento do código de canal, deve recordar com os educandos o que é a distância mínima e citar o teorema 2.3. Só assim será possível a continuidade da construção do código. Em seguida, o docente pode propor a utilização de uma distância mínima **d** = 3 no código de canal, o que acarretará em uma capacidade de correção de $\boldsymbol{\kappa} = \left[\frac{d-1}{2}\right] = \left[\frac{3-1}{2}\right] = 1$ erro. No caso, estaremos supondo que o canal é conhecido e que ele ocasiona até um erro na transmissão.

A partir das informações acima, o docente poderá afirmar que o código a ser construído vai possuir comprimento **n** = 4, dimensão **m** = 2, distância mínima **d** = 3 e será capaz de corrigir até $\boldsymbol{\kappa}$ = 1 erro. Todavia, apesar da definição dos parâmetros anteriores, ainda não foi estabelecido um critério para a construção do código. E mais: nem sabemos se é possível construir um código com os parâmetros indicados para **n**, **m** e **d**. Diante disso, será

[18] Ao fim da atividade, o docente irá explicar aos alunos que foi tomado o valor de **n** = 4 a fim de fosse construído um código linear perfeito.

necessário que o docente faça um breve esclarecimento sobre o que são as matrizes G e H (geradora e de teste de paridade – definição 3.10 e proposição 3.3). Também será necessário explicar quando um conjunto de vetores é linearmente dependente e linearmente independente.

Procedidas às explicações anteriores, o professor irá criar, com o auxílio dos alunos, uma matriz de verificação do código. Para tanto, deverá ser construída uma matriz $H = [-A^t \mid Id_{n-m}]$ de ordem (**n – m**) x **n**. Como **n** = 4 e **m** = 2, a matriz terá ordem 2 x 4 e deve possuir o seguinte formato:

$$H = \begin{bmatrix} a & c & 1 & 0 \\ b & d & 0 & 1 \end{bmatrix}, em\ que\ a, b, c, d \in \{0, 1, 2\}.$$

Como sabemos que **d** = 3, então essa matriz deverá ter (de acordo com o apresentado no item 3.3.5.7) quaisquer duas (= **d – 1** = 3 – 1) colunas linearmente independentes e devem existir três (**d** = 3) colunas linearmente dependentes. Um exemplo em que ocorre isso é na matriz

$$H = \begin{bmatrix} 1 & 2 & 1 & 0 \\ 1 & 1 & 0 & 1 \end{bmatrix}$$

Notemos que as colunas, duas a duas, não são múltiplas uma das outras. Portanto, elas são linearmente independentes. Além disso, (1, 1) = 1.(1, 0) + 1.(0, 1), ou seja, existem três colunas linearmente dependentes.

Construída H, procede-se à construção da matriz G. Conforme afirmado, $H = [-A^t \mid Id_{n-m}]$. Daí, é possível obter a matriz G, pois $G = [Id_m \mid A]$, de ordem **m** x **n**. No nosso exemplo, basta observarmos que

$$-A^t = \begin{bmatrix} 1 & 2 \\ 1 & 1 \end{bmatrix} \leftrightarrow -A = \begin{bmatrix} 1 & 1 \\ 2 & 1 \end{bmatrix} \leftrightarrow A = \begin{bmatrix} 2 & 2 \\ 1 & 2 \end{bmatrix}$$

Não podemos esquecer que estamos realizando as operações acima na base 3:

Tabela 15

+	**0**	**1**	**2**
0	0	1	2
1	1	2	0
2	2	0	1

Assim, podemos dizer que a matriz de codificação é

$$G = \begin{bmatrix} 1 & 0 & 2 & 2 \\ 0 & 1 & 1 & 2 \end{bmatrix}$$

Obtida a matriz G, é possível escrever todas as palavras do código por meio do produto de matrizes **x**.**G**, onde x (matriz linha 1 x 2) é o código da fonte e G (matriz 2 x 4) é a matriz geradora. Cada um dos produtos terá por resultado matrizes coluna (1 x 4) com as palavras do código. As tabelas a seguir foram construídas com o uso de planilhas eletrônicas. Nas figuras abaixo podemos ver algumas fórmulas, as quais foram copiadas e coladas nas células da coluna correspondente.

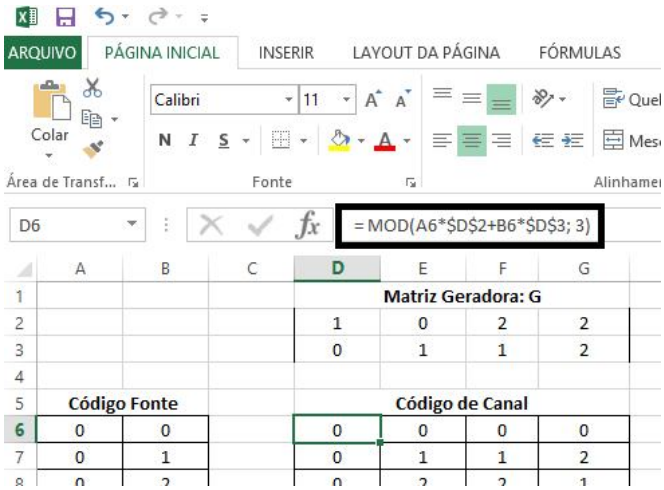


Figura 11

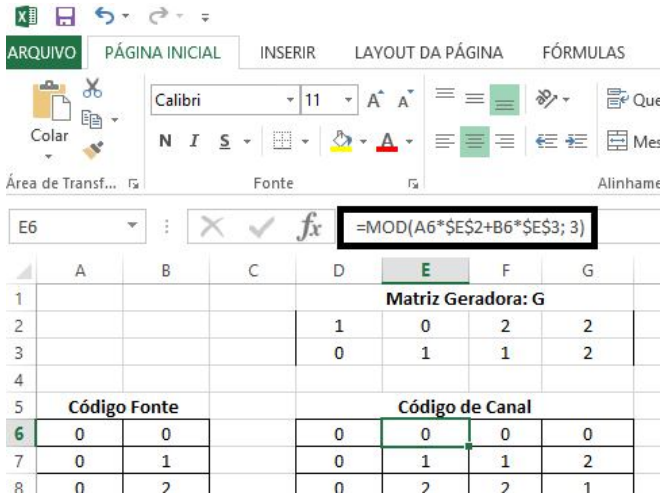


Figura 12

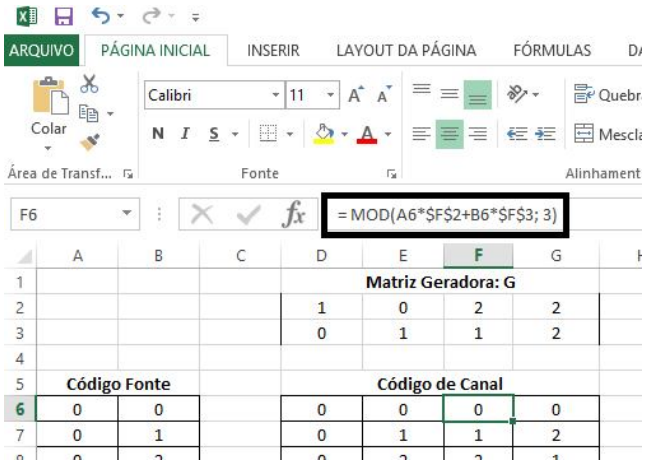


Figura 13

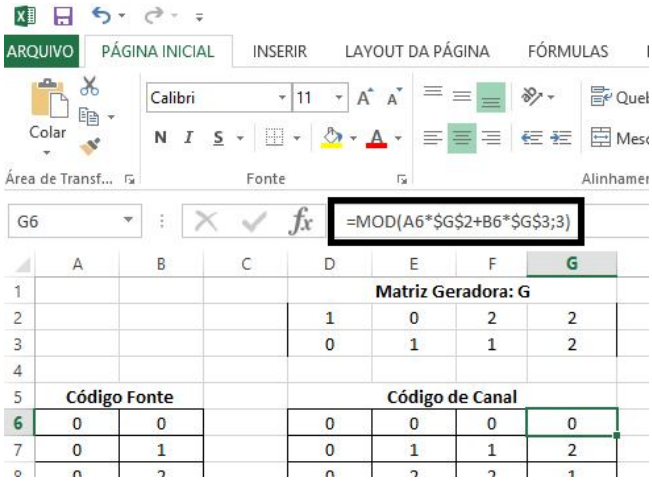


Figura 14

Tabela 16

Matriz Geradora: G

1	0	2	2
0	1	1	2

Código Fonte

0	0
0	1
0	2
1	0
1	1
1	2
2	0
2	1
2	2

Código de Canal

0	0	0	0
0	1	1	2
0	2	2	1
1	0	2	2
1	1	0	1
1	2	1	0
2	0	1	1
2	1	2	0
2	2	0	2

Após a construção das matrizes G e H é possível iniciar o jogo. Para isso, o professor pode propor a divisão da turma em dois grandes grupos. Cada um representará um "jogador". Em cada grupo será realizada a seguinte subdivisão: um aluno será eleito para ser quem dirá ao grupo em qual "casa" será efetuada a jogada do X ou O; os demais alunos formarão dois subgrupos, sendo que o primeiro será responsável pela codificação e o segundo pela decodificação das mensagens.

O jogo funcionará da seguinte maneira. O líder dirá à equipe de codificação em que local pretende que a jogada seja realizada. Ela codificará a informação e o avisará. O líder, então, receberá o vetor referente à jogada e poderá acrescentar um erro antes de encaminhá-la ao líder da equipe adversária. Este receberá o código da jogada e acionará sua equipe de decodificação, a qual irá verificar se a mensagem possui erro e corrigi-la, a fim de que seja efetuada a marcação no jogo da velha que estará no quadro (negro ou branco). Se o grupo marcar uma casa errada, a equipe que enviou a mensagem poderá se manifestar. Para isso, o seu líder deverá explicar no quadro como foi realizada a codificação e, então, o grupo que marcou erroneamente a localização no quadro ficará uma rodada sem jogar. Além disso, ocorrerá a marcação no quadro da jogada correta. Caso seja enviada uma mensagem com mais de um erro ou tenha ocorrido algum equívoco na codificação realizada pelo grupo que efetuou a jogada, este é que ficará uma rodada sem jogar. O procedimento se repetirá até que haja um vencedor ou até que o jogo termine sem vencedores – ocorra empate.

Tomemos um exemplo. Suponhamos que a turma foi separada nos grupos X e O. O grupo X iniciará o jogo. O líder do grupo informa à equipe de codificação que deseja jogar o X na posição a_{22} da matriz. A equipe de codificação, então, verifica que o algoritmo da fonte[19] estabelece que $a_{22} = [(2-1) \quad (2-1)] = [1 \quad 1]$ e efetua o seguinte produto:

$$x.G = [1 \quad 1].\begin{bmatrix} 1 & 0 & 2 & 2 \\ 0 & 1 & 1 & 2 \end{bmatrix} = [1 \quad 1 \quad 0 \quad 1]$$

O líder do grupo, de posse da informação, encaminha ao líder do grupo adversário o vetor $[1 \quad 0 \quad 0 \quad 1]$, em que há um erro.

O líder do grupo O recebe a informação e aciona sua equipe de decodificação. A equipe, então, efetua o seguinte cálculo:

$$H.r^t = \begin{bmatrix} 1 & 2 & 1 & 0 \\ 1 & 1 & 0 & 1 \end{bmatrix}.\begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \end{bmatrix} \neq \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

[19] O algoritmo da fonte, conforme consta no início da atividade, é $a_{ij} = (i-1)(j-1)$

Diante disso, ela conclui que a mensagem foi transmitida com erro. Como já foi estabelecido que no máximo acontece um erro, o vetor do erro tem a forma **e** = (0, ... , x, ..., 0), com $x \neq 0$ na i-ésima posição. Assim, a equipe procederá à busca de qual é a i-ésima posição. Como $He^t = Hr^t$ e $He^t = xh^i$ (h^i é a i-ésima coluna de H), ela verifica que

$$\begin{bmatrix}1\\2\end{bmatrix} = 2.\begin{bmatrix}2\\1\end{bmatrix} = 2.h^2.$$

Portanto, o grupo conclui que o erro ocorreu na segunda coordenada e que o vetor erro é **e** = (0, 2, 0, 0). Assim, a palavra transmitida é **c** = (1001) – (0200) = (1101), a qual corresponde ao código de fonte 11 e, consequentemente, à posição a_{22}.

Agora o grupo O fará sua jogada. O líder escolhe uma posição e informa à equipe de codificação. Suponhamos que seja a_{33}. A referida equipe verifica que o algoritmo da fonte estabelece que $a_{33} = [3-1 \quad 3-1] = [2 \quad 2]$ e efetua o seguinte produto:

$$x.G = [2 \quad 2].\begin{bmatrix}1 & 0 & 2 & 2\\0 & 1 & 1 & 2\end{bmatrix} = [2 \quad 2 \quad 0 \quad 2]$$

O líder do grupo, de posse da informação, encaminha ao líder do grupo adversário o vetor com um erro. Por exemplo, [2 2 0 0].

O líder do grupo X recebe a informação e aciona sua equipe de decodificação. A equipe, então, efetua o seguinte cálculo:

$$H.r^t = \begin{bmatrix}1 & 2 & 1 & 0\\1 & 1 & 0 & 1\end{bmatrix}.\begin{bmatrix}2\\2\\0\\0\end{bmatrix} = \begin{bmatrix}0\\1\end{bmatrix} \neq \begin{bmatrix}0\\0\end{bmatrix}$$

Diante disso, ela conclui que a mensagem foi transmitida com erro. Como já foi estabelecido que no máximo acontece um erro, o vetor do erro tem a forma **e** = (0, ... , x, ..., 0), com $x \neq 0$ na i-ésima posição. Assim, a equipe procederá à busca de qual é a i-ésima posição. Como $He^t = Hc^t$ e $He^t = xh^i$ (h^i é a i-ésima coluna de H), ela verifica que

$$\begin{bmatrix}0\\1\end{bmatrix} = 1.\begin{bmatrix}0\\1\end{bmatrix} = 1.h^4.$$

Portanto, o grupo conclui que o erro ocorreu na quarta coordenada e que o vetor erro é **e** = (0, 0, 0, 1). Assim, a palavra transmitida é **c** = (2200) – (0001) = (2202), a qual corresponde ao código de fonte 22 e, consequentemente, à posição a_{33}.

E o jogo continua até que haja um vencedor ou ocorra o empate.

Após algumas rodadas do jogo, o professor pode propor que os cálculos de codificação e decodificação sejam realizados por meio de planilhas eletrônicas, o que tornará as jogadas mais rápidas e a atividade será mais dinâmica.

Como pode acontecer de algum grupo transmitir uma mensagem com mais de um erro, vamos analisar a seguinte situação. Suponha-se que será efetuada uma jogada na posição a_{33}. Conforme já calculado anteriormente, o vetor que codifica a jogada é [2 2 0 2]. Assim, se for recebido o vetor [2 1 0 0], isto significa que ocorreram dois erros durante a transmissão da mensagem. No momento da decodificação, encontra-se,

$$H.c^t = \begin{bmatrix} 1 & 2 & 1 & 0 \\ 1 & 1 & 0 & 1 \end{bmatrix}.\begin{bmatrix} 2 \\ 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \neq \begin{bmatrix} 0 \\ 0 \end{bmatrix} \; e \; \begin{bmatrix} 1 \\ 0 \end{bmatrix} = 1.\begin{bmatrix} 1 \\ 0 \end{bmatrix} = 1.h^3$$

Nesse caso, é identificado que houve um erro. Todavia, o procedimento de correção nos diz que o vetor erro é (0010) e que a palavra correta é c = (2100) – (0010) = (2120), que embora seja uma palavra que pertence ao código, não é a palavra que foi enviada. Esse cálculo exemplifica a impossibilidade de correção da mensagem quando ocorre mais de um erro na transmissão.

Ao final da atividade o professor deverá explicar aos alunos o motivo pelo qual escolheu o comprimento de palavras **n** = 4, que foi para obter um código linear perfeito. Para isso, o docente deve fazer a verificação de que os parâmetros **M** = 9, **k** = 1, **q** = 3 e **n** = 4 satisfazem as condições descritas na definição 2.8. Além disso, o professor poderá construir todos os discos com centro nas palavras do código e distância 1, a fim de verificar que a união dos mencionados discos é o conjunto $\mathbb{F}_3^4$.

Segue abaixo a relação de discos com os respectivos elementos contidos em cada um deles.

Tabela 17

DISCOS
D(0000, 1) = {0000, 1000, 0100, 0010, 0001, 2000, 0200, 0020, 0002}
D(0112, 1) = {0112, 1112, 2112, 0012, 0212, 0102, 0122, 0110, 0111}
D(0221, 1) = {0221, 1221, 2221, 0021, 0121, 0201, 0211, 0220, 0221}
D(1022, 1) = {1022, 0022, 2022, 1122, 1222, 1002, 1012, 1020, 1021}
D(1101, 1) = {1101, 0101, 2101, 1001, 1201, 1111, 1121, 1100, 1102}
D(1210, 1) = {1210, 0210. 2210, 1010, 1110, 1200, 1220, 1211, 1212}
D(2011, 1) = {2011, 0011, 1011, 2111, 2211, 2001, 2021, 2010, 2012}
D(2120, 1) = {2120, 0120, 2120, 2020, 2220, 2100, 2110, 2121, 2122}
D(2202, 1) = {2202, 0202, 1202, 2002, 2102, 2212, 2222, 2200, 2201}

Procedimento pedagógico

A atividade é mais complexa que as demais sugeridas até o momento. Será necessária constante intervenção do docente no momento das jogadas, auxiliando os alunos. Além disso,

o professor deve incentivar a participação ativa da turma na realização da atividade e assumir um papel de auxiliador na construção do conhecimento.

É importante o professor salientar que, em virtude de o código ser perfeito e o canal causar no máximo um erro, nessa atividade será possível a detecção e a correção de quaisquer palavras.

Informações complementares – determinação do valor de n:

Conforme consta no capítulo referente aos códigos lineares, um código linear C é um subespaço vetorial de $\mathbb{F}_q^n$ de dimensão **m** (**m** < **n**). O conjunto $\mathbb{F}_q$ é um corpo finito de **q** elementos e **n** é o comprimento das palavras do código.

A relação entre o número de palavras **M** de um código e os valores de **q** e **m**, em um código linear, se dá da seguinte forma: $M = q^m$.

Com base nas informações acima é que foi iniciada a elaboração do código utilizado na atividade. De início, foi estabelecido que a quantidade de palavras deveria ser **M** = 9 (que é exatamente a quantidade de casas do tabuleiro do jogo da velha). Assim, como $9 = 3^2$ e como **q** e **m** são números naturais, concluímos que **q** = 3 e **m** = 2. Dessa forma, podemos afirmar que o código tem dimensão **m** = 2 e o alfabeto possui **q** = 3 letras – $\mathbb{F}_3 = \{0, 1, 2\}$.

Com relação à distância mínima, vamos supor que ela seja **d** = 3. Então, podemos dizer que o código permite a identificação de **d – 1** = 2 erros e a correção de **κ** = $\left[\frac{d-1}{2}\right] = 1$ erro.

Pois bem, ante ao exposto, nos resta apenas obter a seguinte informação: o comprimento **n** das palavras. Para obtermos possíveis valores de **n** vamos utilizar a cota de Hamming e o limitante de Singleton. Lembremos, todavia, que o valor de **n** deve ser natural.

A cota de Hamming nos informa que

$$M \leq \frac{q^n}{\sum_{t=0}^{\kappa} \binom{n}{t} (q-1)^t}$$

Daí,

$$9 \leq \frac{3^n}{\binom{n}{0}(3-1)^0 + \binom{n}{1}(3-1)^1}$$

Donde se conclui que

$$1 + 2n \leq 3^{n-2}$$

Fazendo o gráfico no Geogebra das funções y = 1 + 2x e $y = 3^{x-2}$, avaliamos que devemos ter $\boldsymbol{n} \geq 4, n \in N$.

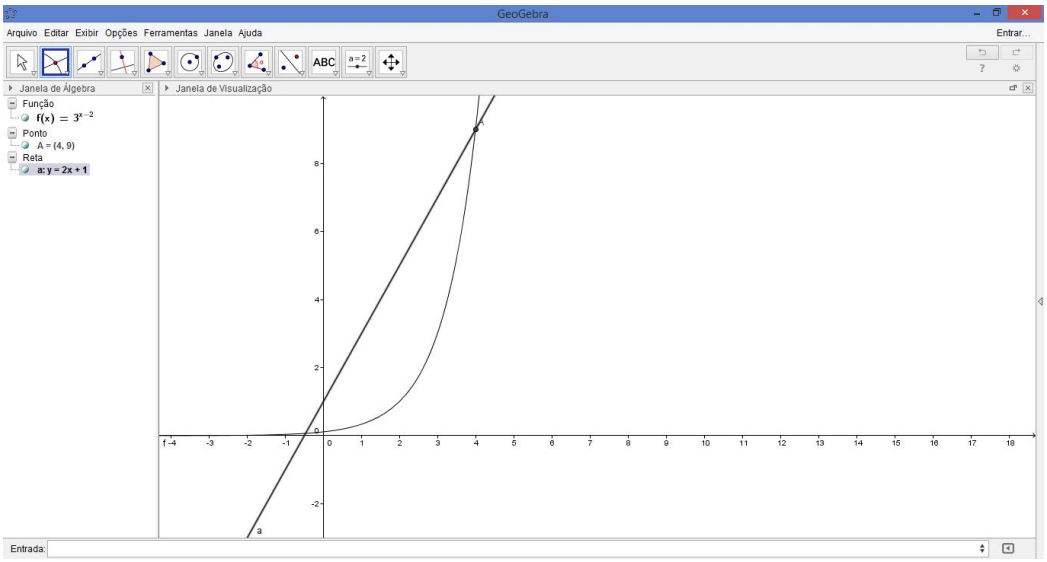


Figura 15

Por outro lado, utilizando a Cota de Singleton, temos que

$$d \leq n - m + 1$$

Daí,

$$3 \leq n - 2 + 1$$

Donde se conclui que

$$4 \leq n$$

Avaliamos, então, que também devemos ter $\boldsymbol{n} \geq 4, n \in N$.

Ante o apresentado, concluímos que o comprimento do código deve ser maior ou igual a 4. Vamos procurar um valor que nos forneça um código perfeito.

Verificação de que o código é um código perfeito:

Podemos notar – com o uso dos parâmetros comprimento **n** = 4, dimensão **m** = 2, distância mínima **d** = 3, capacidade de corrigir até **κ** = 1 erro – que a seguinte equação é satisfeita:

$$M[\sum_{t=0}^{k}(\binom{n}{t})(q-1)^t] = q^n$$

pois,

$$9[\sum_{t=0}^{1}(\binom{4}{t})(3-1)^t] = 9.[\binom{4}{0}(2)^0 + \binom{4}{1}(2)^1 = 9.[1+8] = 81 = 3^4$$

Assim, de acordo com na definição 2.8, podemos dizer que se adotarmos **n** = 4, no presente exemplo, teremos um código perfeito. A vantagem do uso desse código é que sempre que ocorrer 1 (um erro) é possível a sua correção, pois a palavra irá pertencer a um dos discos D (**c**, 1), para **c** pertencente ao código.

Vamos escrever os discos de raio 1 para cada uma das palavras do código. As palavras do código já foram identificadas na seção "aspectos operacionais", da atividade 4. Portanto,

$$\bigcup_{c \in C} D(c, 1) = q^n = 3^4$$

Tabela 17

D(0000, 1) = {0000, 1000, 0100, 0010, 0001, 2000, 0200, 0020, 0002}
D(0112, 1) = {0112, 1112, 2112, 0012, 0212, 0102, 0122, 0110, 0111}
D(0221, 1) = {0221, 1221, 2221, 0021, 0121, 0201, 0211, 0220, 0221}
D(1022, 1) = {1022, 0022, 2022, 1122, 1222, 1002, 1012, 1020, 1021}
D(1101, 1) = {1101, 0101, 2101, 1001, 1201, 1111, 1121, 1100, 1102}
D(1210, 1) = {1210, 0210. 2210, 1010, 1110, 1200, 1220, 1211, 1212}
D(2011, 1) = {2011, 0011, 1011, 2111, 2211, 2001, 2021, 2010, 2012}
D(2120, 1) = {2120, 0120, 2120, 2020, 2220, 2100, 2110, 2121, 2122}
D(2202, 1) = {2202, 0202, 1202, 2002, 2102, 2212, 2222, 2200, 2201}

A tabela da página anterior apenas confirma que, de fato, o código é perfeito.

Outra maneira de identificar quais são as palavras do código:

Construída H, podemos verificar quais palavras pertencem ao código. Como **m** = 2 e **n** = 4, serão acrescentados dois dígitos de redundância à codificação da fonte. Assim, podemos ter palavras de comprimento quatro, segundo os parâmetros da tabela abaixo[20].

Tabela 18

Fonte		Redundâncias	
0	0		
0	1		
0	2		
1	0		
1	1		
1	2		
2	0		
2	1		
2	2		

[20] Os espaços vazios da tabela 18 serão preenchidos por elementos do conjunto $\mathbb{F}_3 = \{0, 1, 2\}$.

Portanto, existem três opções para cada dígito da redundância, uma vez que o alfabeto é {0, 1, 2}. Ou seja, para cada uma das linhas da tabela acima há nove possibilidades de formação de palavra. Todavia, para que a palavra formada pertença ao código, devemos ter $H.c^t = 0$, onde c^t é a transposta de cada linha.

As contas de verificação são um pouco extensas. Então, uma excelente opção é o uso de planilhas eletrônicas. Para tanto, vamos escrever a matriz H e, em seguida, as possibilidades de cada linha. Por meio de fórmulas, as planilhas eletrônicas realizam as contas e nos informam qual a palavra c em que ocorre $H.c^t = 0$.

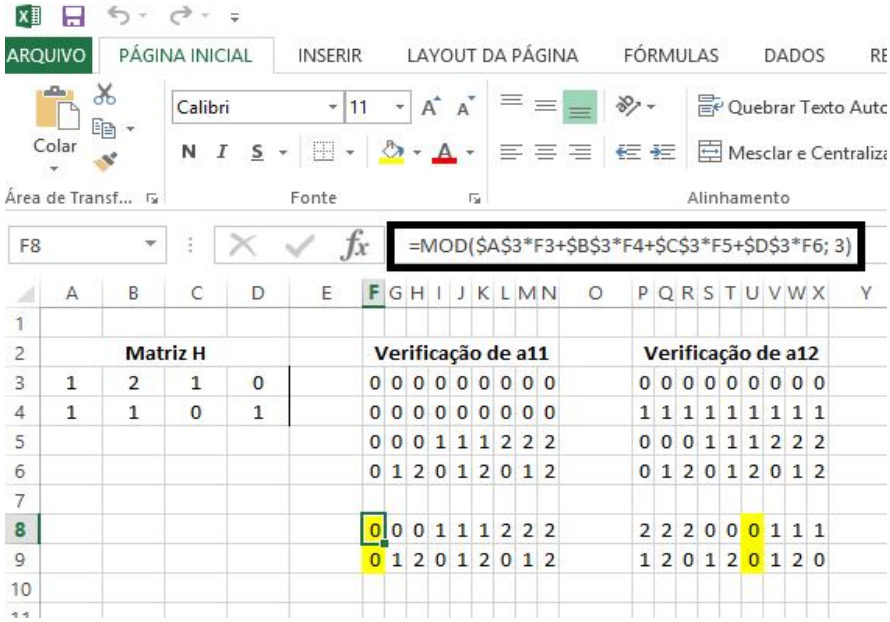


Figura 16

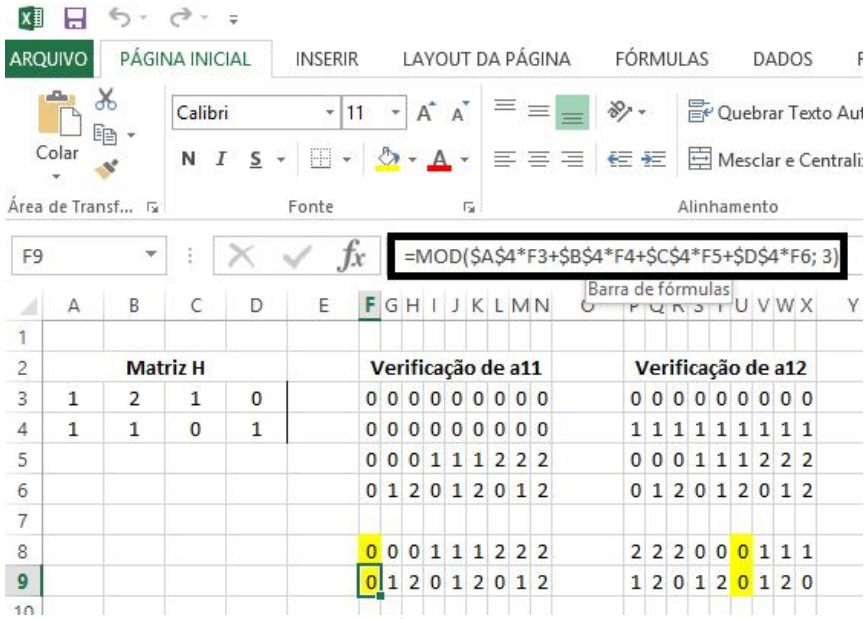


Figura 17

Tabela 19

Matriz H

1	2	1	0
1	1	0	1

Verificação de a_{11}

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	1	1	1	2	2	2	
0	1	2	0	1	2	0	1	2	

0	0	0	1	1	1	2	2	2	
0	1	2	0	1	2	0	1	2	

Verificação de a_{12}

0	0	0	0	0	0	0	0	0	
1	1	1	1	1	1	1	1	1	
0	0	0	1	1	1	2	2	2	
0	1	2	0	1	2	0	1	2	

2	2	2	0	0	0	1	1	1	
1	2	0	1	2	0	1	2	0	

Verificação de a_{13}

0	0	0	0	0	0	0	0	0	
2	2	2	2	2	2	2	2	2	
0	0	0	1	1	1	2	2	2	
0	1	2	0	1	2	0	1	2	

1	1	1	2	2	2	0	0	0	
2	0	1	2	0	1	2	0	1	

Verificação de a_{21}

1	1	1	1	1	1	1	1	1	
0	0	0	0	0	0	0	0	0	
0	0	0	1	1	1	2	2	2	
0	1	2	0	1	2	0	1	2	

1	1	1	2	2	2	0	0	0	
1	2	0	1	2	0	1	2	0	

Verificação de a_{22}

1	1	1	1	1	1	1	1	1	
1	1	1	1	1	1	1	1	1	
0	0	0	1	1	1	2	2	2	
0	1	2	0	1	2	0	1	2	

0	0	0	1	1	1	2	2	2	
2	0	1	2	0	1	2	0	1	

Verificação de a_{23}

1	1	1	1	1	1	1	1	1	
2	2	2	2	2	2	2	2	2	
0	0	0	1	1	1	2	2	2	
0	1	2	0	1	2	0	1	2	

2	2	2	0	0	0	1	1	1	
0	1	2	0	1	2	0	1	2	

Verificação de a_{31}

2	2	2	2	2	2	2	2	2	
0	0	0	0	0	0	0	0	0	
0	0	0	1	1	1	2	2	2	
0	1	2	0	1	2	0	1	2	

2	2	2	0	0	0	1	1	1	
2	0	1	2	0	1	2	0	1	

Verificação de a_{32}

2	2	2	2	2	2	2	2	2	
1	1	1	1	1	1	1	1	1	
0	0	0	1	1	1	2	2	2	
0	1	2	0	1	2	0	1	2	

1	1	1	2	2	2	0	0	0	
0	1	2	0	1	2	0	1	2	

Verificação de a_{33}

2	2	2	2	2	2	2	2	2	
2	2	2	2	2	2	2	2	2	
0	0	0	1	1	1	2	2	2	
0	1	2	0	1	2	0	1	2	

0	0	0	1	1	1	2	2	2	
1	2	0	1	2	0	1	2	0	

Ante ao exposto, podemos montar a seguinte tabela, na qual estão as palavras do código de canal:

Tabela 20

Posição	Código da fonte	Código de Canal
a_{11}	00	0000
a_{12}	01	0112
a_{13}	02	0221
a_{21}	10	1022
a_{22}	11	1101
a_{23}	12	1210
a_{31}	20	2011
a_{32}	21	2120
a_{33}	22	2202

6 CONSIDERAÇÕES FINAIS

Novos métodos de ensino de Matemática têm revolucionado a forma de construção do conhecimento pelos educandos. Essas novas concepções têm trazido o foco do processo de ensino-aprendizagem para o aluno, colocando-o como um sujeito ativo, questionador, reflexivo, capaz de analisar problemas e buscar soluções.

Essa nova maneira de ensinar Matemática faz com que o professor também tenha que modificar sua prática, pois será ele quem criará as condições para que o aprendizado ocorra e, para isso, serão necessárias a elaboração de situações-problema, a contextualização histórica, o uso de computadores em sala de aula, dentre outras situações possíveis. Além disso, o docente também deve adotar uma postura de orientador, e não de "detentor de todo o conhecimento", indicando o caminho a ser seguido pelo educando para que este investigue e resolva as situações-problema.

As atividades propostas seguem esta concepção de ensino-aprendizagem. Elas foram desenvolvidas com o objetivo de contextualizar o estudo de matrizes, vetores, sistemas de numeração e sistemas lineares. Por meio delas os educandos podem analisar problemas, propor soluções, verificar que em determinados momentos a solução não é a melhor possível. Isso acontece porque as atividades tentam recriar a forma como se deu o desenvolvimento do estudo dos códigos corretores de erros, uma vez que é apresentada a seguinte sequência: um modelo de código que não detecta e nem corrige erros; modelos de códigos que apenas detectam erros; um código que detecta e corrige (mas não muito preciso); e um código linear diferenciado, chamado de código perfeito.

Portanto, por meio da sequência proposta, espera-se que os alunos adquiram competências relacionadas ao desenvolvimento do pensamento algébrico, à segurança para adaptar o uso de conceitos matemáticos em outros contextos, à análise de informações e tomada de decisões, bem como comecem a avaliar quando o uso de uma ferramenta tecnológica pode ser útil para resolver um problema.

Dessa forma, podemos concluir que as atividades auxiliam na qualificação dos educandos para, ao final do Ensino Médio, serem inseridos no mercado de trabalho ou para o prosseguimento dos estudos.

7 REFERÊNCIAS

[1] ARMELLA, Luiz Moreno; WALDEGG, Guilhermina. **Educação Matemática.** México: 1992, v.4, p. 7-15.

[2] BITTENCOURT, Simone. **Como nosso cérebro lê?** Pré-Univesp, n. 4, 2010. Disponível em: <http://www.univesp.ensinosuperior.sp.gov.br/preunivesp/503/como-nosso-c-rebro-l-.html>. Acesso em: 10 jun. 2014.

[3] BAHIA, Flaviano. **Um Primeiro Curso sobre Códigos Corretores de Erros**. In: ERMAC 2010: I Encontro Regional de Matemática Aplicada e Computacional, 2010, São João Del Rei, MG. **Anais**... Disponível em <http://www.ufsj.edu.br/portal2-repositorio/File/i-ermac/anais/mini cursos/mc8.pdf>. Acesso em: 15 janeiro 2015.

[4] BRASIL. Ministério da Educação. Secretaria de Educação Média e Tecnológica. **Parâmetros Curriculares Nacionais (Ensino Médio)**. Brasília: MEC, 2000. Disponível em: < http://portal .mec.gov.br/seb/arquivos/pdf/ciencian.pdf. Acesso em: 11 maio 2014.

[5] BRASIL. Ministério da Educação. Secretaria da Educação Média e Tecnológica. **Parâmetros Curriculares Nacionais + (PCN+) - Ciências da Natureza e suas Tecnologias.** Brasília: MEC, 2002. Disponível em: <http://portal.mec.gov.br/seb/arquivos/pdf/CienciasNatureza.pdf>. Acesso em: 11 maio 2014.

[6] COSTELLO, D. J.; HAGENAUER J.; ILMAI H.; WICKER S. B. Applications of Erros-Control Coding. **IEEE Transactions on Information Theory**, out. 98. Disponível em: <http://arquivoescolar.org/bitstream/arquivo-e/132/4/cap2.pdf>. Acesso em: 16 abril 2014.

[7] D'AMBRÓSIO, Ubiratan. **A história da matemática: questões historiográficas e políticas e reflexos na educação matemática**. In: BICUDO, M. A. V. (Ed.). Pesquisa em educação matemática: concepções e perspectivas. São Paulo: UNESP, 1999. p. 97–115. Disponível em: <http://cattai.mat.br/site/files/ensino/uneb/pfreire/docs/HistoriaDaMatematica/Ubiratan_DAmbrosio_doisTextos.pdf>. Acesso em 01 abril 2015.

[8] ESQUINCA, Josiane Colombo Pedrini. **Aritmética: Códigos de Barras e Outras Aplicações de Congruências.** 2013. Dissertação (Mestrado Profissional em Matemática em Rede Nacional) – Centro de Ciências Exatas e Tecnologia, Universidade Federal de Mato Grosso do Sul, Campo Grande – MS, 2013. Disponível em: < http://bit.profmat-sbm.org.br/xmlui/bitstream/handle/123456789/371/2011_00238_JOSIANE_COLOMBO_PEDRINI_ESQUINCA.pdf?sequence=1>. Acesso em: 07 janeiro 2015.

[9] FEIO, Evandro dos Santos Paiva; SILVA, Francisco Hermes Santos da. **Reflexões sobre linguagem e inclusão à luz da teoria da aprendizagem significativa**. Aprendizagem Significativa em revista – v2 (3), pp. 58-68, 2012. Disponível em: <http://www.if.ufrgs.br/asr/artigos/Artigo_ID37/v2_n3_a2012.pdf>. Acesso em: 11 maio 2014.

[10] FIGUEIREDO, Luis Manoel; CUNHA, Marisa Ortega da. **Álgebra Linear I**. Volume 2. 2. ed. Rio de Janeiro: Fundação CECIERJ, 2010.

[11] FONTES, Marício de Moraes; FONTES, Dineusa Jesus dos Santos; FONTES, Miriam de Morais. **O Computador como Recurso Facilitador da Aprendizagem Matemática**. In: I Simpósio Nacional de Ensino de Ciência e Tecnologia, 2009, Universidade Tecnológica Federal do Paraná - UTFPR. **Anais**... Disponível em: <http://www.sinect.com.br/anais2009/artigos/10%20Ensinodematematica/Ensinodematematica_artigo13.pdf>. Acesso em: 13 março 2015.

[12] HEFEZ, Abramo; FERNANDES, Cecília de Souza. **Introdução à Álgebra Linear.** SBM. 1ª Edição. 2012. Coleção Profmat.

[13] HEFEZ, Abramo; VILLELA, Maria Lucia T. **Códigos Corretores de Erros.** 2. ed. Rio de Janeiro: IMPA, 2008.

[14] MENEGHESSO, Carla. **Códigos Corretores de Erros.** 2012. Trabalho de Conclusão de Curso (Licenciatura em Matemática) – Centro de Ciências Exatas e de Tecnologia, Universidade Federal de São Carlos, São Carlos, 2012. Disponível em: <http://www.dm.ufscar.br/dm/attachments/article/5/monografia_carla%20TCC.pdf>. Acesso em: 15 abril 2014.

[15] MILIES, César Polcino. **A Matemática dos Códigos de Barras**. Disponível em: <http://mat.ufg.br/bienal/2006/mini/polcino.pdf>. Acesso em: 10 março 2015.

[16] MILIES, César Polcino. **Breve introdução à Teoria dos Códigos Corretores de Erros**. In: 1º Colóquio de Matemática da Região Centro-Oeste, 2009, UFMS, Campo Grande. **Anais...** São Paulo: Instituto de Matemática e Estatística da Universidade de São Paulo, 2009. Disponível em: <http://www.coloquiodematematica.ufms.br/conteudo/ material/mc09.pdf>. Acesso em: 15 abril 2014.

[17] MILIES, César Polcino. **Breve introdução à Teoria dos Códigos Corretores de Erros**. In: 1º Colóquio de Matemática da Região Nordeste, 2011, UFS, Sergipe. **Anais**... São Paulo: Instituto de Matemática e Estatística da Universidade de São Paulo, 2011. Disponível em: < http://www.sbm.org.br/docs/coloquios/NE-1.04.pdf>. Acesso em: 15 abril 2014.

[18] MOREIRA, Marco Antonio. **Aprendizagem Significativa: Um Conceito Subjacente.** In: Aprendizagem Significativa em Revista – V1(3), p. 25-46, 2011. Disponível em: <http://www.if.ufrgs.br/asr/artigos/Artigo_ID16/v1_n3_a2011.pdf>. Acesso em: 8 janeiro 2015.

[19] OLIVEIRA, Elizabeth M. **Metodologia para o uso da Informática na Educação.** Revista da Sociedade Brasileira de Educação Matemática. SBEM. Ano 13. nº.23. dez. 2007.

[20] OLIVEIRA, José Sávio Bicho de; ALVES, Angela Xavier; NEVES, Sandra Socorro de Miranda. **História da Matemática: Contribuições e Descobertas para o Ensino-aprendizagem de Matemática.** Disponível em: <http://www.sbemrn.com.br/site/II%20erem/comunica/doc/comunica14.pdf>. Acesso em 16 março 2015.

[21] QUILES, Cláudia Natália Saes. **O Uso do Computador na Escola: Mapeando os "Métodos de Ensinar" na Sala de Tecnologias Educacionais (STE).** In: IX Congresso Nacional de Educação – EDUCERE. PUCPR, 2009. Disponível em: < http://www.pucpr.br/eventos/educere/educere2009/anais/pdf/2487_1780.pdf>. Acesso em: 12 janeiro 2015.

[22] RIOS, Isabel Lugão; FIGUEIREDO, Luiz Manoel; CUNHA, Marisa Ortegoza da. **Álgebra Linear I**. Volume 1. 3. ed. Rio de Janeiro: Fundação CECIERJ, 2010.

[23] SILVA, Valeska Aparecida Rodrigues da. **Propostas de Utilização de Códigos de Barras como Recurso Didático para o Ensino da Matemática.** 2013. Dissertação (Mestrado Profissional em Matemática em Rede Nacional) – Instituto de Ciências Exatas, Universidade Federal Rural do Rio de Janeiro, Seropédica, 2013. Disponível em: <http://bit.profmat-sbm.org.br/xmlui/bitstream/handle/123456789/520/2011_00418_VALESKA_APARECIDA_RODRIGUES_DA_SILVA.pdf?sequence=1>. Acesso em: 21 abril 2014.

[24] SILVEIRA, J. F. Porto. **O que é um problema matemático?** 2001. Disponível em: <http://www.mat.ufrgs.br/~portosil/resu1.html>. Acesso em: 20 fevereiro 2015.

[25] SOUZA, Adenilce Oliveira; CÂMARA, Marcos Antonio da. Códigos Corretores de Erros Lineares. **FAMAT em Revista**, Uberlândia, nº 6, p. 75-110, maio 2006. Disponível em: <http://www.portal.famat.ufu.br/sites/famat.ufu.br/files/Anexos/Bookpage/Famat_Revista_06.pdf>. Acesso em: 21 abril 2014.

[26] SOUZA, Mário José de. **Mini Curso Códigos Corretores de Erros**. In: XXIII Semana do IME na Universidade Federal de Goiás, UFG, Goiânia. Notas disponíveis em: <https://semanadoime.mat.ufg.br/up/34/o/min_mario.pdf>. Acesso em: 15 janeiro 2015.

[27] SOUZA, Natália Pedroza de. **Uma Análise dos Esquemas de Dígitos Verificadores Usados no Brasil.** 2013. Dissertação (Programa de Pós-Graduação em Ciências Computacionais da UFRJ) – Centro de Tecnologia e Ciênicas, Instituto de Matemática e Estatística, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2013. Disponível em: < http://www.bdtd.uerj.br/tde_busca/arquivo.php?codArquivo=6100>. Acesso em: 10 março 2015.

[28] VALENTE, José Armando. **O Computador na Sociedade do Conhecimento.** Ministério da Educação. Coleção Informática para a mudança na Educação. Disponível em: <http://www.dominiopublico.gov.br/download/texto/me003150.pdf>. Acesso em: 02 abril 2015.

[29] VELOSO, Marcelo Oliveira; COSTA, Fernanda Rodrigues Alves. **Sistemas de Identificação Modular: uma Aplicação no Ensino Fundamental.** 2014. Trabalho de Conclusão de Curso (Mestrado Profissional em Matemática - PROFMAT), Universidade Federal de São João Del Rei, Alto Paraopeba, 2014. Disponível em: < http://bit.profmat-sbm.org.br/xmlui/bitstream/handle/123456789/1166/2012_00948_FERNANDA_RODRIGUES_ALVES_COSTA.pdf?sequence=1>. Acesso em: 15 março 2015.

[30] VIANA, Marger da Conceição Ventuna; SILVA, Célia Maria da. **Concepção de Professores de Matemática sobre a Utilização da História da Matemática no Processo de Ensino Aprendizagem**. Disponível em: <http://www.limc.ufrj.br/htem4/papers/15.pdf>. Acesso em: 15 março 2015.

[31] VILELA, Vera Lúcia Maria Luciano. **O Lúdico como Instrumento de Aprendizagem no Ensino da Matemática**. Dissertação (Mestrado em Educação) – Universidade Federal de Goiás, Goiânia, 2008. Disponível em: <https://ppge.fe.ufg.br/up/6/o/Dissert-%20Vera.pdf>. Acesso em: 15 janeiro 2015.

[32] VOLOCH, José Felipe. **Códigos corretores de erros.** In: 16º Colóquio brasileiro de Matemática, 1987, IMPA, Rio de Janeiro. **Anais...** Rio de Janeiro: IMPA, 1987. Disponível em: <http://www.impa.br/opencms/pt/biblioteca/cbm/16CBM/16_CBM_87 _06.pdf>. Acesso em: 05 abril 2014.

[33] VYGOTSKY, Lev Semenovich. **A formação social da mente**. 4. ed. Brasileira. São Paulo: Martins Fontes, 1984, p. 160.

APÊNDICE A

Folha da atividade 1

Questão 1: A primeira final de copa do mundo de futebol foi realizada entre quais países?

a) Uruguai e Argentina
b) Argentina e Brasil
c) Alemanha e Itália
d) Uruguai e Brail
e) Inglaterra e Paraguai

Questão 2: Na frase: Antônio e Maria foram ao cinema. O sujeito da frase é classificado em:

a) Oculto
b) Simples
c) Composto
d) Oração sem sujeito
e) Indeterminado

Questão 3: O Brasil vive uma crise hídrica. Qual o nome do rio que abastece os estados do Rio de Janeiro, São Paulo e Minas Gerais e que foi objeto de um recente acordo com a Agência Nacional de Águas para sua utilização?

a) Rio Acre
b) Rio Amazonas
c) Rio São Francisco
d) Rio Paraíba do Sul
e) Rio Guandú

Questão 4: Qual o nome do primeiro presidente eleito pelo voto direto da população após o período da ditadura militar?

a) Fernando Henrique Cardoso
b) Sarney
c) Fernando Collor
d) Dilma
e) Lula

Questão 5: Quais os nomes das camadas eletrônicas de distribuição de elétrons?

a) A, B, C, D, E, F, G
b) K, L, M, N, O, P, Q
c) R, S, T, U, V, W, X
d) M, N, O, P, Q, R, S
e) A, E, I, O, U

Questão 6: Qual o nome do primeiro brasileiro campeão de surf?

a) Guga
b) Pedro Alvarenga
c) José Dirceu
d) Gabriel Medina
e) Mick Fanning

Questão 7: Suponha que sobre uma mesa haja um livro. Qual será a força que a mesa exerce sobre o livro (força normal), sabendo que a força com que a Terra o atrai é de 10N?

a) 5N
b) 8N
c) 15N
d) 3N
e) 10N

Questão 8: Os insetos possem:

a) Um par de asas e 3 pares de pernas
b) Dois pares de asas e 2 pares de pernas
c) Quatro pares de asas e 3 pares de pernas
d) Um ou dois pares de asas e 3 pares de pernas
e) Um par de asas e um par de pernas

APÊNDICE B

Folha da atividade 2

Questão 1: Nos códigos de barras das figuras a seguir, faça o cálculo referente à identificação do dígito verificador.

Fonte: Foto de uma embalagem de chocolate

Fonte: Foto de uma embalagem de milho verde.

Questão 2: Determine o dígito verificado de um produto que tem código de barras correspondente à sequência numérica 789603609511X.

Questão 3: Considere que o número 4891668326689 é a sequência numérica de um código de barras. Verifique em cada caso se o erro será detectado.

a) A sequência digitada é 489166832669<u>8</u> (houve inversão dos dois últimos dígitos)

b) A sequência digitada é 489<u>6</u>168326689 (houve inversão do quarto e quinto dígitos na digitação)

Observações referentes à folha de atividades 2:

Questão 1: A primeira questão pode ser substituída por uma atividade na qual sejam distribuídas aos alunos embalagens de produtos a fim de que eles analisem o código de barras correspondente.

Questão 2:

Resolução:

Como o código contém 13 dígitos, trata-se de um EAN-13. Façamos o produto escalar entre os vetores (1, 3, 1, 3, 1, 3, 1, 3, 1, 3, 1, 3, 1) e (7, 8, 9, 6, 0, 3, 6, 0, 9, 5, 1, 1, X). Teremos, portanto, como resultado, a expressão 101 + x. O menor valor de x inteiro não negativo que pode nos levar ao menor múltiplo de dez mais próximo de 101 é x = 9.

Questão 3:

a)

Resolução:

O código possui 13 dígitos. Trata-se, portanto, de um EAN-13. Devemos fazer o produto escalar entre os vetores (1, 3, 1, 3, 1, 3, 1, 3, 1, 3, 1, 3, 1) e (4, 8, 9, 1, 6, 6, 8, 3, 2, 6, 6, 9, 8). O resulta é 144, que não é um múltiplo de dez. Portanto, o erro será detectado.

b)

Resolução:

Vamos fazer o produto escalar entre os vetores (1, 3, 1, 3, 1, 3, 1, 3, 1, 3, 1, 3, 1) e (4, 8, 9, 6, 1, 6, 8, 3, 2, 6, 6, 9, 8). O resulta é 150, que é um múltiplo de dez. Portanto, o erro não será detectado.