UNIVERSIDADE ESTADUAL DE FEIRA DE SANTANA

Departamento de Ciências Exatas

PROFMAT - Mestrado Profissional em Matemática em Rede Nacional

Produto Educacional

CÓDIGOS CORRETORES DE ERROS: UMA PROPOSTA APLICADA AO ENSINO MÉDIO.

Aldo Brito de Jesus
Maurício de Araujo Ferreira

Feira de Santana
Novembro de 2024

Sumário

1 Introdução **1**

2 Sequência Didática **3**

2.1 Aula 1 - Corrigindo Erros 3

2.2 Aula 2 - Códigos de Hamming 6

2.3 Aula 3 - Matrizes, Planilhas e Códigos de Hamming 11

3 Conclusão **13**

Referências Bibliográficas **14**

A Construção da Máquina **15**

B Representação do Salão **24**

C Referencial teórico **26**

C.1 Códigos Corretores de Erros 26

C.2 Código de Hamming 33

Capítulo 1

Introdução

O presente produto educacional é resultado do trabalho desenvolvido em uma dissertação de mestrado do PROFMAT. O objetivo é apresentar uma proposta de abordagem códigos corretores de erros no Ensino Médio, possibilitando assim que o professor utilize este produto educacional como material para um minicurso que tenha estudantes desta etapa de ensino como público alvo.

A implementação do Novo Ensino Médio e a criação da Base Nacional Comum Curricular (BNCC) traz uma nova visão para o ensino da Matemática na Educação Básica. A nova proposta de ensino coloca o aluno como um ser ativo e corresponsável pelo processo de ensino e aprendizagem. O ensino da Matemática deixa de ser uma mera transmissão de conteúdos e estratégias de resolução de questões e coloca o aluno como um "ser matemático", ou seja, um ser capaz de usar e criar matemática para as demandas sociais e tecnológicas [2].

De acordo com a BNCC [2],"no Ensino Médio o foco é a construção de uma visão integrada da Matemática, aplicada à realidade em diferentes contextos." Tendo isso em vista, enxergamos os códigos corretores de erros como uma oportunidade para o professor expor a Matemática em sala de aula como parte da realidade em que vivemos e mostrar para os alunos o potencial desta área de conhecimento para o desenvolvimento de novas tecnologias. Acreditamos ainda que, o trabalho com códigos em sala de aula proporciona aos alunos a oportunidade de vivenciar o fazer matemático, investigando e propondo soluções para desafios do mundo contemporâneo, desenvolvendo assim a segunda habilidade da área de Matemática e suas Tecnologias proposta pela BNCC.

> Propor ou participar de ações para investigar desafios do mundo contemporâneo e tomar decisões éticas e socialmente responsáveis, com base na análise de problemas sociais, como os voltados a situações de saúde, sustentabilidade, das implicações da tecnologia no mundo do trabalho, entre outros, mobilizando e articulando conceitos, procedimentos e linguagens próprios da Matemática. [2]

Apresentamos uma sequência didática que o professor pode aplicar para alunos do terceiro ano do Ensino Médio e/ou fazer uma adaptação para aplicar nas demais turmas, mostrando o potencial da Matemática na vida contemporânea. A sequência didática está organizada em três momentos, no primeiro será utilizado o *Whatsapp* para o entendimento dos conceitos de Códigos Corretores de Erros, no segundo, os estudantes desenvolverão uma atividade numa folha quadriculada que representa um restaurante no qual um robô leva o pedido até a mesa do cliente, e no terceiro, os alunos utilizarão uma máquina construída em planilha eletrônica para simular e resolver problemas de codificação do robô.

Com o objetivo de tornar este material o mais auto-suficiente o quanto possível, incluímos no apêndice uma revisão da literatura, trazendo os conceitos básicos de códigos corretores de erros e códigos de Hamming com uma abordagem por meio de exemplos, desta maneira, em um primeiro momento em sala de aula, o professor do Ensino Médio pode apresentar os conceitos por meio dos exemplos sem a necessidade de recorrer a conceitos mais avançados de álgebra.

Capítulo 2

Sequência Didática

Neste capítulo apresentaremos uma sequência didática para o trabalho com códigos corretores de erros em uma turma de terceiro ano do Ensino Médio. Trata-se de uma sequências de aulas ligadas entre si, com o objetivo de apresentar e explorar os códigos corretores para introduzir vários outros conceitos da Matemática que descreveremos no decorrer do texto.

A primeira atividade trata-se de uma pequena brincadeira utilizando o WhatsApp como recurso didático e tem como objetivo fazer uma apresentação inicial sobre distância de Hamming, redundâncias e outros conceitos iniciais dos códigos corretores de erros. Na segunda atividade trabalhamos especificamente com o código de Hamming $(7,4)$, apresentamos uma situação problema na qual os alunos devem utilizar o código $ham(7,4)$ para codificar os comandos de um determinado robô. Na terceira e última atividade, exploramos as planilhas eletrônicas como ferramenta para fazer simulação de erros e realizar alguns procedimentos no processo de transmissão e correção das palavras no código $ham(7,4)$, e propomos que os alunos utilizem a planilha para resolver os problemas da atividade anterior, de modo que seja possível visualizar as vantagens do uso do computador em algumas tarefas matemáticas.

2.1 Aula 1 - Corrigindo Erros

Nesta primeira aula o professor deve convidar o aluno a participar da atividade matemática. Este convite pode ser feito com uma breve apresentação sobre códigos corretores de erros semelhante ao que foi feito no Capítulo C.1 deste trabalho.

- *Tema da Aula:* Corrigindo erros.
- *Conteúdo:* Códigos Corretores de Erros.
- *Competências da BNCC:* COMPETÊNCIA ESPECÍFICA 1. Utilizar estratégias, conceitos e procedimentos matemáticos para interpretar situações em diversos contextos, sejam atividades cotidianas, sejam fatos das Ciências da Natureza e Humanas,

das questões socioeconômicas ou tecnológicas, divulgados por diferentes meios, de modo a contribuir para uma formação geral.

- *Habilidade da BNCC:* (EM13MAT106) Identificar situações da vida cotidiana nas quais seja necessário fazer escolhas levando-se em conta os riscos probabilísticos (usar este ou aquele método contraceptivo, optar por um tratamento médico em detrimento de outro etc.).

- *Objetivos:* Convidar o aluno a refletir sobre o uso da Matemática nos códigos corretores de erros; Conhecer os códigos corretores de erros por meio de exemplos.

- *Tempo:* 50 minutos.

- *Desenvolvimento:*

O professor deverá iniciar a aula com a brincadeira "Teclado Quebrado", que é um jogo simples, no qual cada equipe deverá corrigir erros de digitação em uma conversa no WhatsApp.

Antes de iniciar o jogo, o professor deve dividir a turma em grupos de 4 alunos. Pelo menos um membro de cada uma das equipes deve ter um smartfone com acesso a internet no qual esteja instalado o aplicativo do WhatsApp. Em seguida o educador deve criar um grupo de WhatsApp, no qual serão enviadas as mensagens, e adicionar um aluno de cada equipe. É importante que os alunos desativem o corretor ortográfico do teclado.

Na primeira rodada da disputa, o professor deverá enviar mensagens no grupo contendo um erro em alguma das palavras e as equipes deverão fazer a correção. A equipe que conseguir corrigir o erro mais rapidamente e reenviar a mensagem no grupo ganha 5 pontos. É interesse que as frases iniciais sejam de fácil correção e que gradativamente aumente a dificuldade de correção. Veja alguns exemplos de frases que o professor pode utilizar:

- O cathorro é bravo. (O cachorro é bravo);

- Comprei um helular novo. (Comprei um celular novo);

- Cortei o bolo com a maca. (Cortei o bolo com a faca);

- Tenho uma xaca leiteira. (Tenho uma vaca leiteira);

- Qual é o preço dessa kala? (Qual é o preço dessa mala);

- Na minha casa tem um xato. (Na minha casa tem um pato);

Na segunda rodada, com o intuito de deixar a correção mais difícil, o professor deve usar apenas palavras. Neste caso, cada equipe que corrigir a palavra de maneira correta

ganha 10 pontos. Nesta rodada existe a possibilidade de pedir uma dica em relação a palavra, no entanto, acertos com uma dica vale apenas a metade dos pontos, ou seja, apenas 5 pontos. A dica deve ser uma outra palavra que tenha relação com a palavra que foi enviada. Esta segunda rodada é importante para que os alunos percebam que o contexto ajuda no processo de correção das palavras. Veja alguns exemplos de palavras com erros:
capendário(calendário); *coléfio* (colégio); *camepo* (camelo); *ratal* (natal); *xovo* (novo); *banata*(batata); *yala* (tala); *wola* (mola);

Na terceira rodada cada equipe deverá criar uma mensagem com no máximo um erro de digitação em cada palavra e enviar para o professor. As equipes deverão enviar também a frase corrigida. Em seguida o professor deverá enviar todas as mensagens no grupo para as equipes tentarem corrigir e reenviar. É claro que uma equipe não pode enviar a frase criada por ela mesma. Caso alguma das mensagens não seja corrigida pelas equipes, o grupo que criou a mensagem ficará com os 10 pontos da frase.

A partir das estratégias apresentadas pelos alunos para realizar o processo de correção das frases e palavras, o professor pode introduzir a ideia de redundância e e fazer uma introdução sobre os códigos corretores de erros, apresentando as dificuldades que existem no processo de transmissão de uma mensagem por meio digital e a necessidade de corrigir os erros cometidos.

Durante a discussão do problema, provavelmente os alunos vão usar em algum momento a ideia de proximidade entre duas palavras, desta maneira o professor pode apresentar a definição de métrica de Hamming e generalizar para um código qualquer. É importante que os alunos percebam que apenas a métrica de Hamming não é suficiente para realizar o processo de correção e que é necessário ter outras ferramentas. Neste momento o professor pode apresentar o código F_2^2 do robô como um outro exemplo de código no qual temos problemas com a distância entre as palavras e propor o seguinte problema:
Problema: Considere o código $\{00000,\ 01011,\ 10110,\ 11101\}$ e a codificação abaixo:

$$\begin{array}{rcccc} \text{Leste} & \longrightarrow & 00 & \longrightarrow & 00000 \\ \text{Oeste} & \longrightarrow & 01 & \longrightarrow & 01011 \end{array} \qquad \begin{array}{rcccc} \text{Norte} & \longrightarrow & 10 & \longrightarrow & 10110 \\ \text{Sul} & \longrightarrow & 11 & \longrightarrow & 11101. \end{array}$$

Suponha que durante a transmissão de uma mensagem para um determinado robô tenha ocorrido no máximo um erro em cada palavra. Você saberia descrever os comandos enviados para o robô que recebeu a mensagem:

$$00000\ \ 10111\ \ 11101\ \ 10000\ \ 11011\ \ 11111?$$

O professor deve explorar este problema para apresentar a necessidade de acrescentar redundâncias na palavra de um determinado código, como apresentado no Capítulo C.1.

- *Materiais necessários:* Material corriqueiro do professor.
- *Avaliação:* Participação na realização das tarefas.
- *Referências:* [5], [3] e [6].

2.2 Aula 2 - Códigos de Hamming

Nesta segunda aula será necessário que os alunos tenham uma noção de base binária, neste caso, o professor pode dedicar vinte minutos da aula para fazer uma breve explanação em relação a isso. Para explicar um pouco sobre os códigos de Hamming, o professor deve utilizar o texto da Seção C.2.

- *Tema da Aula:* Códigos de Hamming.
- *Conteúdo:* Código de Hamming $(7, 4)$.
- *Competências da BNCC:* COMPETÊNCIA ESPECÍFICA 3. Utilizar estratégias, conceitos, definições e procedimentos matemáticos para interpretar, construir modelos e resolver problemas em diversos contextos, analisando a plausibilidade dos resultados e a adequação das soluções propostas, de modo a construir argumentação consistente; COMPETÊNCIA ESPECÍFICA 4. Compreender e utilizar, com flexibilidade e precisão, diferentes registros de representação matemáticos (algébrico, geométrico, estatístico, computacional etc.), na busca de solução e comunicação de resultados de problemas.
- *Habilidades da BNCC:* (EM13MAT405) Utilizar conceitos iniciais de uma linguagem de programação na implementação de algoritmos escritos em linguagem corrente e/ou matemática.
- *Objetivos:* Conhecer os códigos de Hamming e suas aplicações; Estabelecer relações entre os códigos corretores de erros e o processo de transmissão de mensagens;
- *Tempo:* 100 minutos.
- *Desenvolvimento:*

No primeiro momento da aula o professor deve explicar o conteúdo da Seção C.2. Após a explicação deve apresentar a seguinte situação para os alunos:

Em um determinado restaurante com 25 mesas, não existe garçom, todos os pedidos são registrados por meio do celular e um robô leva o pedido até a mesa do cliente. Para facilitar o serviço do robô, o restaurante possui um salão em forma de quadrado, cada mesa é posicionada em um lugar fixo.

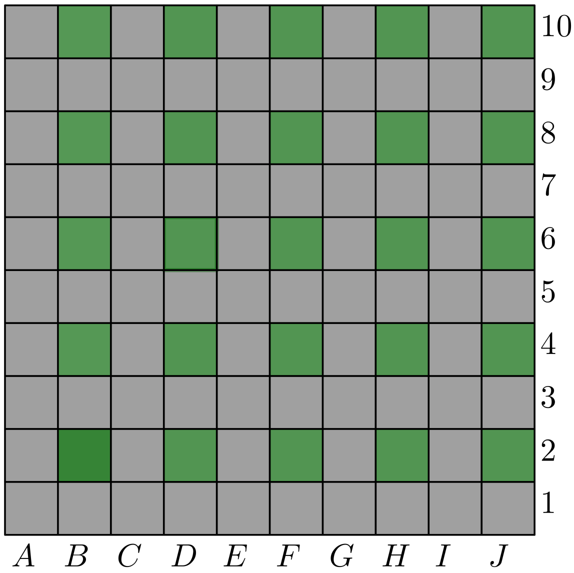


Figura 2.1: Representação do salão.

Na Figura 2.1 temos uma representação da organização das mesas, onde cada quadrado do tabuleiro tem exatamente um metro de lado e cada casa verde demarca a posição de uma mesa. Por exemplo, o primeiro quadrado verde mais abaixo e mais a esquerda representa a posição da mesa $B2$.

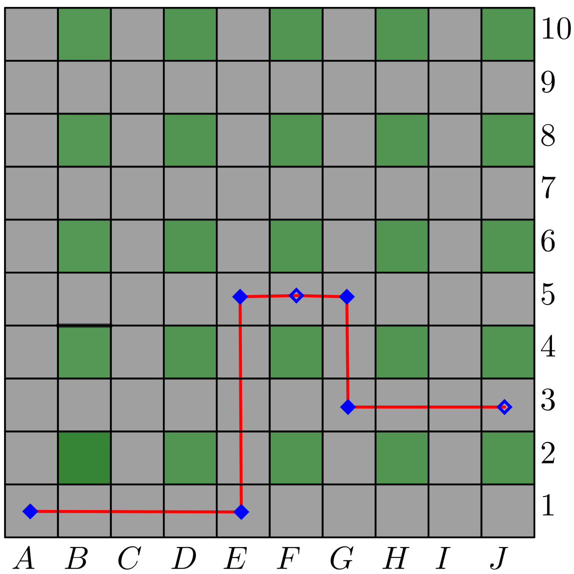


Figura 2.2: Trajeto do Robô.

Para fazer a entrega de um pedido o robô sempre parte do centro da casa $A1$, podendo levar mais de uma pedido por vez respeitando a ordem de cada pedido, e se move de acordo com os comandos recebidos para realizar as entregas, sendo que no final sempre retorna por um dos caminhos mais curto para a posição inicial. A entrega do pedido é realizado pelo lado direito da mesa em relação aos números que aparecem no tabuleiro, isto é, sempre do lado sul. Por exemplo, no trajeto da Figura 2.2, o robô

- partiu da casa $A1$, andou quatro casas para direita, quatro casas para cima, uma casa para direita e entregou o pedido da mesa $F6$ pelo lado sul da mesa;
- andou uma casa para a direita, duas casa para baixo, três casa para direita e entregou o pedido da mesa $J4$ pelo lado sul da mesa.
- retornou para origem pelo caminho mais curto.

Seja n um número natural. Suponha que o robô obedeça aos seguintes comandos:

- $SIGA$ (para o robô andar);
- $PARE$ (para o robô parar e realizar a entrega do pedido);
- xD (para o robô andar x casas para a direita);
- xE (para o robô andar x casas para a esquerda);
- xB (para o robô andar x casas para a baixo);
- xC (para o robô andar x casas para a cima).

Assim, por exemplo, para realizar o trajeto da Figura 2.2 os comandos enviados para o robô são:

$$SIGA \quad 4 \quad D \quad 4 \quad C \quad 1 \quad D \quad PARE \quad SIGA \quad 1 \quad D \quad 2 \quad B \quad 3 \quad D \quad PARE$$

Suponha também que o robô tenha o seguinte código fonte:

COMANDOS	0	1	2	3	4	5	6	7	8	9	D	E	C	B	$SIGA$	$PARE$
	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
PALAVRAS	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
DO CÓDIGO	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

Para transmitir estes comandos podemos usar o código de canal $ham(7,4)$ para o robô, ou seja, seguindo o algoritmo dos códigos de Hamming descrito na Seção C.2, é possível acrescentar redundâncias a cada uma das palavras do código fonte do robô e obter o código de canal abaixo:

0	1	2	3	4	5	6	7	8	9	D	E	C	B	$SIGA$	$PARE$
0	1	0	1	1	0	1	0	1	0	1	0	0	1	0	1
0	1	1	0	0	1	1	0	1	0	0	1	1	0	0	1
0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	1	1	0	1	0	0	1	0	1	1	0	1	0	0	1
0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

Os alunos deverão resolver os seguintes problemas:

Problema 1: Utilizando o código de canal $ham(7,4)$, escreva uma mensagem para ser enviado ao robô que deve partir da casa $A1$ e entregar os pedidos das mesas $D8$, $J10$ e $H4$, sem retornar na origem.

Problema 2: Determine o percurso e as entregas realizadas por um robô que recebeu a seguinte mensagem sem erros:

c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9	c_{10}	c_{11}	c_{12}	c_{13}	c_{14}	c_{15}	c_{16}
0	1	1	0	0	1	1	1	0	1	1	0	0	1	1	1
0	0	0	1	1	1	0	1	0	0	0	1	1	1	0	1
1	0	1	0	1	0	1	1	1	0	1	0	1	0	1	1
0	1	1	1	1	1	1	1	0	0	1	1	1	1	1	1
1	1	0	0	1	0	0	1	1	0	0	0	1	0	0	1
1	0	1	1	0	0	1	1	1	1	1	1	0	0	1	1
0	0	0	0	0	1	0	1	0	1	0	0	0	1	0	1

Problema 3: Um robô recebeu a mensagem abaixo com no máximo um erro em cada palavra. Neste caso é possível fazer a correção e decodificar a mensagem recebida. Determine o percurso e as entregas realizadas pelo robô destacando as palavras erradas e a

posição em que ocorreu o erro.

c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9	c_{10}	c_{11}	c_{12}	c_{13}	c_{14}	c_{15}	c_{16}
0	1	1	0	0	0	1	1	0	1	0	0	0	0	0	0
0	0	0	1	1	1	0	1	0	0	1	1	1	1	1	1
1	0	1	0	1	0	1	1	1	0	1	1	1	0	1	1
0	1	1	0	1	0	1	0	0	1	0	1	1	0	0	1
1	1	0	1	1	1	0	1	1	0	0	0	1	1	0	1
1	0	1	1	0	0	1	1	1	0	1	1	0	0	1	1
1	0	0	0	0	1	0	1	0	1	1	0	0	0	1	1

Problema 4: Um robô recebeu a mensagem abaixo para realizar as entregas das mesas $D4$ e $H6$, no entanto, devido algum erro de comunicação no sistema entregou um dos pedidos em uma mesa errada. Determine a mesa que recebeu o pedido errado e explique o que pode ter ocorrido para o robô ter realizado a entrega na mesa errada sem identificar o erro.

c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9	c_{10}	c_{11}	c_{12}	c_{13}	c_{14}	c_{15}	c_{16}
0	0	1	0	0	1	0	0	0	0	1	0	0	1	1	1
0	1	0	1	1	1	0	1	0	0	0	1	1	0	0	1
1	0	1	0	1	0	1	1	1	0	1	0	1	0	1	1
0	1	1	1	1	1	1	1	0	0	1	1	1	1	1	1
1	0	0	0	1	0	0	1	1	0	0	0	1	0	0	1
1	1	1	1	0	0	1	1	1	1	1	1	0	1	0	1
0	0	0	0	0	1	0	1	0	1	0	0	0	1	0	1

Problema 5: Um robô recebeu a mensagem abaixo para realizar algumas entregas, no entanto, devido algum erro de comunicação no sistema, ao sair da primeira mesa que fez a entrega e se deslocar para a segunda, acabou se chocando com uma das mesas do salão. Determine a mesa onde ocorreu o acidente e explique o que pode ter ocorrido.

c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9	c_{10}	c_{11}	c_{12}	c_{13}	c_{14}	c_{15}	c_{16}
0	0	1	1	0	1	1	1	0	1	1	0	0	1	1	1
0	1	0	0	1	1	0	1	0	1	0	1	1	1	0	1
1	0	1	0	1	0	1	1	1	0	1	0	1	0	1	1
0	1	1	1	1	1	1	1	0	1	0	1	1	1	1	1
1	0	0	1	1	0	0	1	1	1	0	0	1	0	0	1
1	1	1	0	0	0	0	1	1	0	1	1	0	0	1	1
0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	1

- *Materiais necessários:* Material corriqueiro do professor.
- *Avaliação:* Atividade impressa.
- *Referências:* [5], [1] e [3]
- *Anexo:* Apêndice B.

2.3 Aula 3 - Matrizes, Planilhas e Códigos de Hamming

Este terceiro momento com os alunos tem como objetivo utilizar o computador para realizar os cálculos e as demais operações necessárias para codificar, corrigir erros e decodificar uma mensagem usando o código $ham(7,4)$. Nesta aula propomos que os alunos utilize uma máquina construída em planilha eletrônica para resolver os problemas da Seção 2.2. Os processos envolvidos na construção é necessário que os alunos tenham um conhecimento básico de lógica, planilhas, teoria de números, matrizes e geometria analítica. No Apêndice A, fazemos a descrição de como construir a máquina, mas, não indicamos que a construção e os processos envolvidos sejam trabalhados no Ensino Médio, o professor pode simplesmente disponibilizar a planilha para os alunos responder aos questionamentos. Acreditamos que esta aula, com a construção da planilha, pode ser apresentada em um minicurso sobre códigos lineares para alunos de um algum curso de Matemática que tenham cursado Teoria dos Números e Geometria Analítica .

- *Tema da Aula:* Matrizes, Planilhas e Códigos de Hamming.
- *Conteúdos:* Matrizes e Códigos de Hamming.
- *Competências da BNCC:* COMPETÊNCIA ESPECÍFICA 3. Utilizar estratégias, conceitos, definições e procedimentos matemáticos para interpretar, construir modelos e resolver problemas em diversos contextos, analisando a plausibilidade dos resultados e a adequação das soluções propostas, de modo a construir argumentação consistente; COMPETÊNCIA ESPECÍFICA 4. Compreender e utilizar, com flexibilidade e precisão, diferentes registros de representação matemáticos (algébrico, geométrico, estatístico, computacional etc.), na busca de solução e comunicação de resultados de problemas.
- *Habilidades da BNCC:* (EM13MAT405) Utilizar conceitos iniciais de uma linguagem de programação na implementação de algoritmos escritos em linguagem corrente e/ou matemática.
- *Objetivos:* Relacionar códigos de Hamming e matrizes; Usar as planilhas eletrônicas para realizar cálculos matematicos.

- *Tempo:* 100 minutos.

- *Desenvolvimento:*

Nesta aula o professor deve explicar aos alunos o funcionamento da máquina e como digitar os comandos na planilha eletrônica. Em seguida propor os problemas para que eles possam responder utilizando a planilha.

Problema 1: Usar a máquina para resolver os problemas propostos na Seção 2.2. Em seguida escrever as vantagens de utilizar o computador.

Máquina sem simulação de erros: *Clique aqui.*

Problema 2: Utilize a máquina com simulação de erros para responder de forma empírica aos seguintes questionamentos:

a) A máquina é capaz de corrigir qualquer palavra recebida com no máximo um erro?

b) Usando o código $ham(7,4)$, no caso das palavras recebidas com até dois erros é possível saber que ocorreu erros, mas não é possível determinar a quantidade e as posições dos erros. O que pode acontecer com o robô ao receber uma palavra com exatamente dois erros?

c) O que pode acorrer com o robô ao receber uma palavra com exatamente três erros?

Observação: Para simular um novo erro basta clicar no botão enviar.

Máquina com simulação de erros: *Clique aqui.*

- *Materiais necessários:* Computador com internet e material corriqueiro do professor.

- *Avaliação:* Atividade impressa e utilização da planilha eletrônica.

- *Referências:* [5], [1] e [3]

Capítulo 3

Conclusão

Um dos grandes desafios no processo de ensino e aprendizagem é motivar os alunos para o estudo da Matemática. Boa parte dos conteúdos matemáticos trabalhados na educação básica apresentam pouca aplicabilidade no cotidiano, desta maneira, muitos alunos questionam o porquê estudar tais conteúdos. Neste trabalho apresentamos uma possibilidade para o professor trabalhar com a Matemática de maneira contextualizada com o cotidiano e, para isto, procuramos evidenciar a necessidade de ferramentas matemáticas para o desenvolvimento de novas tecnologias. Mostramos que o estudo dos códigos corretores de erros, apesar de não ser conteúdo do Ensino Médio, pode servir como motivação para o estudo de alguns conteúdos desta etapa da educação básica.

No corpo do trabalho procuramos exibir exemplos das definições e proposições apresentadas no texto, tornando assim mais fácil o entendimento dos conceitos relativos à teoria de códigos corretores de erros e possibilitando uma abordagem mais simples do conteúdo em sala de aula. Mostramos que é possível simplificar a exposição de alguns conteúdos usando uma linguagem acessível para alunos do Ensino Médio sem perder o formalismo matemático necessário para generalizar os procedimentos.

As atividades propostas neste trabalho servem como uma introdução aos estudos dos códigos corretores de erros em sala de aula. Procuramos introduzir recursos tecnológicos nas atividades de modo que os alunos tenham um maior interesse pelo tema e consigam visualizar na prática a utilização dos códigos corretores de erros por um computador. A máquina descrita neste trabalho permite que os alunos vejam os procedimentos realizados pelo computador para identificar e corrigir erros em uma mensagem.

Referências Bibliográficas

[1] BAHIA, F. *Um primeiro curso sobre códigos corretores de erros.* ERMAC 2010: I Encontro Regional de Matemática Aplicada e Computacional, 2010. Disponível em: <http://www.ufsj.edu.br/portal2-repositorio/File/i-ermac/anais/minicursos/mc8.pdf> acesso em 22 de outubro de 2021.

[2] BRASIL . *Base Nacional Comum Curricular. Ensino Médio.* Brasília: MEC. Versão final. Disponível em: http://basenacionalcomum.mec.gov.br/. Acesso em: 11 de outubro de 2021.

[3] HEFEZ, A.; VILLELA, M. L. T. *Códigos corretores de erros.* Instituto de Matematica Pura e Aplicada, 2008.

[4] LIRA, E. H. C. de. *Códigos Corretores de Erros no Ensino Médio: um estudo sobre o Código de Hamming.* 2018. Dissertação de Mestrado. Universidade Federal Rural de Pernambuco.

[5] MILIES, C. P. *Breve introduçao a teoria dos códigos corretores de erros.* Colóquio de Matemática da Região Centro-Oeste, SBM, 2009.

[6] MILIES, C. P. *A matemática dos códigos de barras.* Artigo, USP-Departamento de Matemática, São Paulo, SP, Brasil. Recuperado em, v. 14, 2006.

Apêndice A

Construção da Máquina

Descrevemos abaixo os passos que os alunos deverão seguir para montar a máquina capaz de simular erros de transmissão e realizar o processo de codificação, correção de erros e decodificação do código de $ham(7,4)$.

Inicialmente os alunos deverão montar a parte de geração do código. Neste caso devem seguir os seguintes passos:

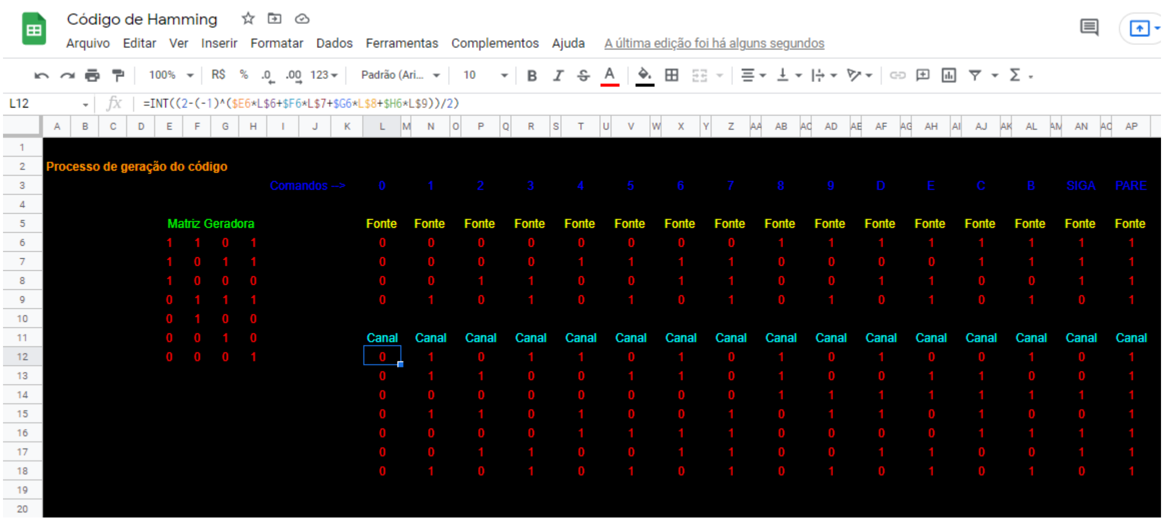


Figura A.1: Planilha: Máquina de codificação.

1º passo: Digitar a matriz $G_{7\times 4}$ da Seção C.2, que gera o código $ham(7,4)$. A primeira coluna da matriz deve iniciar na célula E6, a segunda na célula F6 e assim sucessivamente. Para identificar a matriz é possível mesclar as células E6, F6, G6 e H6 e digitar a frase: Matriz Geradora.

2º passo Na linha 3 da planilha, iniciando na célula L3 e terminando na célula AP3 e sempre saltando uma célula, digite os comandos: 0, 1, 2, $\cdots$, 8, 9, D, E, C, B, SIGA e PARE.

3º passo Na linha 5 da planilha, iniciando na célula L5 e terminando na célula AP5 e sempre saltando uma célula, digite a palavra: Fonte.

4º passo Logo abaixo de cada uma das palavras fonte digite um elemento do conjunto $F_2^4 = \{0,1\}^4$, que será uma codificação do comando correspondente que se encontra acima da palavra fonte. Sugerimos que seja usada a ordem crescente dos números na base binária.

5º passo Na linha 11 da planilha, iniciando na célula L11 e terminando na célula AP11 e sempre saltando uma célula, digite a palavra: Canal.

6º passo Na célula L12 digite a seguinte fórmula:

= INT((2-(-1)ˆ ($E6*L$6+$F6*L$7+$G6*L$8+$H6*L$9))/2)

Neste caso, a expressão x_1= $E6*L$6+$F6*L$7+$G6*L$8+$H6*L$9 é referente a multiplicação entre a primeira linha da matriz geradora e a palavra do conjunto F_2^4 correspondente ao comando 1. A função INT retorna a parte inteiro do número ((2-(-1)ˆ ($E6*L$6+$F6*L$7+$G6*L$8+$H6*L$9))/2) que será 1 no caso em que x_1 é ímpar, pois neste caso temos,

$$x_1 = \frac{2-(-1)^{\text{ímpar}}}{2} = \frac{2-(-1)}{2} = \frac{2+1}{2} = \frac{3}{2} = 1,5,$$

e, 0 caso contrário, pois

$$x_1 = \frac{2-(-1)^{\text{par}}}{2} = \frac{2-1}{2} = \frac{1}{2} = 0,5.$$

6º passo Copiar a célula L12 e colar nas 6 células logo abaixo. Neste caso será codificada a primeira palavra do código.

7º passo Selecionar as células L12 até L18, copiar e colar logo abaixo de cada uma das palavras canal. Obtendo assim o código $ham(7,4)$, como mostrado na Figura A.1.

Para montar o mecanismo de codificação da mensagem a ser enviada, os alunos deverão seguir os passos abaixo:

1º passo Na linha 23 da planilha, destacar as células L23, N23, P23 e assim sucessivamente até a célula AP23. Estas células serão utilizadas para digitar a mensagem a ser enviada, sendo um comando em cada célula.

2º passo Na célula L24 digite a seguinte fórmula:

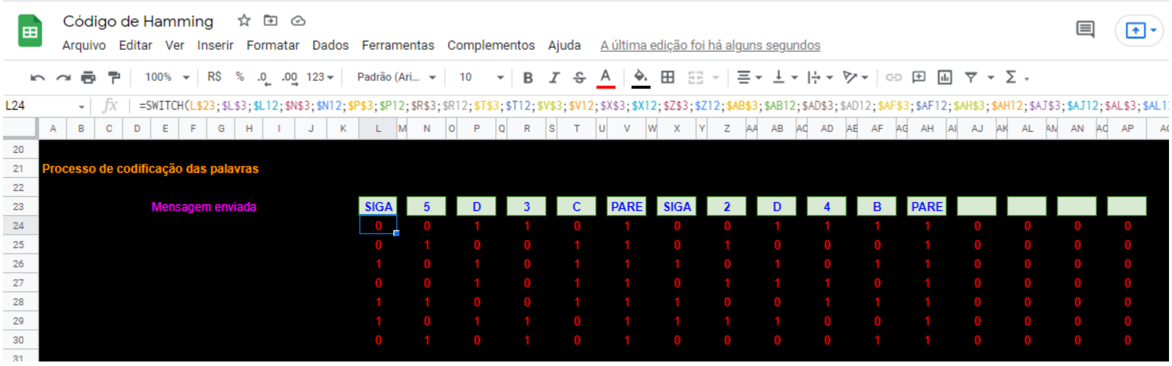


Figura A.2: Planilha: Codificação de uma mensagem.

=SWITCH(L$23; $L$3; $L12; $N$3; $N12; $P$3; $P12; $R$3; $R12; $T$3; $T12; $V$3; $V12; $X$3; $X12; $Z$3; $Z12; $AB$3; $AB12; $AD$3; $AD12; $AF$3; $AF12; $AH$3; $AH12; $AJ$3; $AJ12; $AL$3; $AL12; $AN$3; $AN12; $AP$3; $AP12)

A função SWITCH faz uma comparação entre o comando digitado na célula L23 e os comandos digitados nas células L3, N3, $\cdots$, AN3 e AP3, e retorna a primeira entrada da palavra do código $ham(7,4)$ referente ao comando encontrado.

3º passo Copiar a célula L24 e colar na 6 células imediatamente abaixo.

4º passo Selecionar as células L24 até L30, copiar e colar logo abaixo de cada uma das células destacadas para digitação do comando, finalizando a construção do mecanismo de codificação. Veja o exemplo da Figura A.2.

Para realizar o processo de simulação de erros será necessário gerar números aleatórios na planilha, neste caso os alunos deverão criar esse mecanismo utilizando a função ALEATÓRIOENTRE. Para criar esta parte da máquina os alunos deverão seguir os seguintes passos:

1º passo Na célula L33 digitar a fórmula:

$$= \text{ALEATÓRIOENTRE}(1; 100)$$

2º passo Copiar a célula L33 e colar nas células L34 e L35.

3º passo Copiar as células L33, L34 e L35 e copiar nas demais colunas que tenha o nome canal, sempre iniciando na linha 33.

Veja que para fazer a simulação de erros em uma palavra do código $ham(7,4)$, a planilha realiza os seguintes processos:

- Faz três sorteios seguidos de números em um conjunto fixado e anota os resultados.

- Forma uma sequência com os três números sorteados. Os números podem ser repetidos;
- Observa os elementos da sequência e adiciona um erro em cada uma das ‘posições’ sorteadas. Por exemplo, no caso da sequência (1, 4, 12), a palavra será recebida com erros nas posições 1 e 4, e no caso da sequência (2, 13, 2) a palavra será recebida com um erro na posição 2.

No caso da máquina que estamos usando, cada sorteio é feito com os número entre 1 e 100 e as palavras têm exatamente 7 posições. Neste caso, temos que a probabilidade de receber a palavra com três erros é de 0,021%, a probabilidade de receber com dois erros é aproximadamente 1,18% e a probabilidade de receber com uma erros é aproximadamente 18,4%. Para alterar estas probabilidades basta modificar o intervalo no qual os números são sorteados.

Para montar a parte mais importante da máquina, ou seja, para montar o mecanismo de detecção e correção erros, e decodificação da palavra os alunos deverão seguir os seguintes passos:

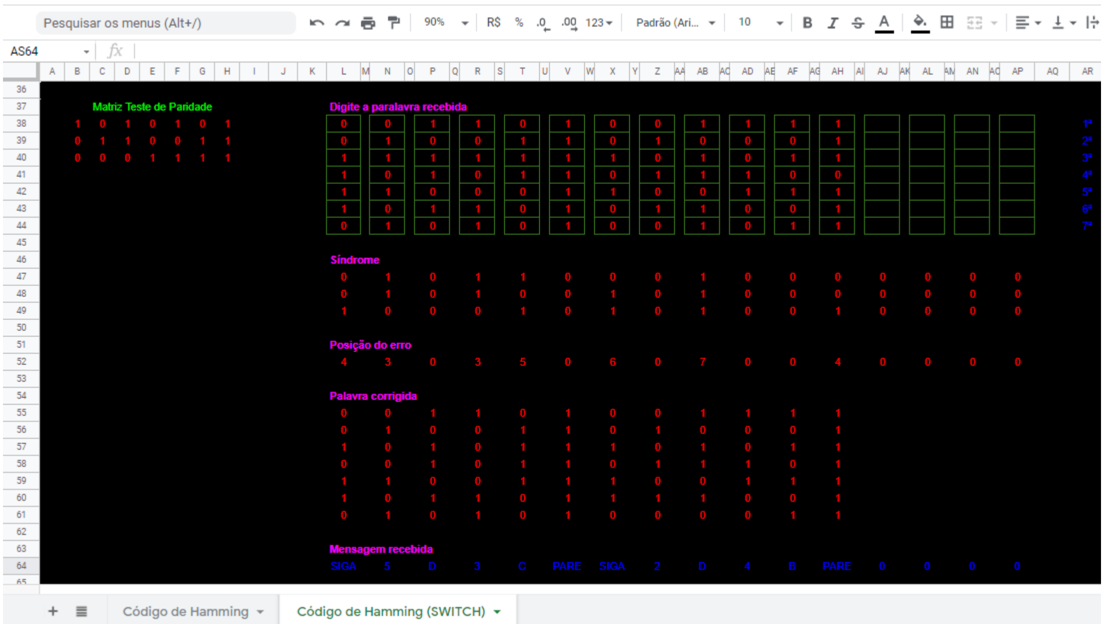


Figura A.3: Planilha: detecção e correção de erros.

1º passo Digitar a matriz teste de paridade $H_{3\times7}$ da Seção C.2, nas linhas 38, 39, 40, iniciando na célula B38 da planilha.

2º passo Destacar as células L38 até L44, N38 até N44, P38 até P44 e assim sucessivamente até a coluna AP. Esta células servirão como entrada para as palavras recebidas.

3º passo Na célula L38 digitar a seguinte fórmula:

=SE(OU(L$33=1;L$34=1;L$35=1);ABS(L24-1);L24)

Veja que neste caso a palavra terá uma erro na primeira entrada se, no sorteio aleatório, aparecer um número igual a 1 em qualquer uma das células L33, L34 ou L35.

4º passo Na célula L39 digitar a seguinte fórmula:

=SE(OU(L$33=2;L$34=2;L$35=2);ABS(L24-1);L24)

e assim sucessivamente até a célula L44, na qual será digitada a fórmula

=SE(OU(L$33=7;L$34=7;L$35=7);ABS(L24-1);L24)

5º passo Selecionar as células L38 até L44, copiar e colar nas demais colunas onde aparece o nome canal, sempre iniciando na linha 38.

6º passo Digitar a palavra 'síndrome' na célula L46.

7º passo Na célula L47, digitar a seguinte fórmula:

=INT((2-(-1)^ ($B38*L$38 + $C38*L$39 + $D38*L$40 + $E38*L$41 + $F38*L$42 + $G38*L$43 + $H38*L$44))/2)

8º passo Copiar a célula L47 e colar nas células L48 e L49.

9º passo Selecionar as células L46, L47 E L48, copiar e colar nas demais colunas, como mostrado na Figura A.3.

10º passo Digitar a frase 'posição do erro' na célula L51.

11º passo Na célula L52, digitar a seguinte fórmula:

=L47 + L48*2 + L49*4

Esta fórmula faz a mudança de base do número binário (síndrome) para um número decimal (posição do erro).

12º passo Copiar a célula L52 e colar nas células N52, P52, $\cdots$, AN52 e AP52.

13º passo Digitar a frase 'palavra corrigida' na célula L54.

14º passo Nas células L55, L56, L57, L58, L59, L60 e L61, digitar, respectivamente, as seguintes fórmulas:

=SE(L$52=1;INT((2+(-1)^ (L38))/2);L38)
=SE(L$52=2;INT((2+(-1)^ (L39))/2);L39)
=SE(L$52=3;INT((2+(-1)^ (L40))/2);L40)
=SE(L$52=4;INT((2+(-1)^ (L41))/2);L41)
=SE(L$52=5;INT((2+(-1)^ (L42))/2);L42)
=SE(L$52=6;INT((2+(-1)^ (L43))/2);L43)
=SE(L$52=7;INT((2+(-1)^ (L44))/2);L44)

14º passo Selecionar as células L55, L56, L57, L58, L59, L60 e L61, copiar e colar nas células N55, P55, $\cdots$, AN55 e AP55.

15º passo Digitar a frase ‘palavra recebida e corrigida’ na célula L63.

16º passo Digitar na célula L64, a seguinte fórmula:

=SE(E(L55=$L$12; L56=$L$13; L57=$L$14; L58=$L$15; L59=$L$16; L60=$L$17; L61=$L$18); $L$3; SE(E(L55=$N$12; L56=$N$13; L57=$N$14; L58=$N$15; L59=$N$16; L60=$N$17; L61=$N$18); $N$3; SE(E(L55=$P$12; L56=$P$13; L57=$P$14; L58=$P$15; L59=$P$16; L60=$P$17; L61=$P$18); $P$3; SE(E(L55=$R$12; L56=$R$13; L57=$R$14; L58=$R$15; L59=$R$16; L60=$R$17; L61=$R$18); $R$3; SE(E(L55=$T$12; L56=$T$13; L57=$T$14; L58=$T$15; L59=$T$16; L60=$T$17; L61=$T$18); $T$3; SE(E(L55=$V$12; L56=$V$13; L57=$V$14; L58=$V$15; L59=$V$16; L60=$V$17; L61=$V$18); $V$3; SE(E(L55=$X$12; L56=$X$13; L57=$X$14; L58=$X$15; L59=$X$16; L60=$X$17; L61=$X$18); $X$3; SE(E(L55=$Z$12; L56=$Z$13; L57=$Z$14; L58=$Z$15; L59=$Z$16; L60=$Z$17; L61=$Z$18); $Z$3; SE(E(L55=$AB$12; L56=$AB$13; L57=$AB$14; L58=$AB$15; L59=$AB$16; L60=$AB$17; L61=$AB$18); $AB$3; SE(E(L55=$AD$12; L56=$AD$13; L57=$AD$14; L58=$AD$15; L59=$AD$16; L60=$AD$17; L61=$AD$18); $AD$3; SE(E(L55=$AF$12; L56=$AF$13; L57=$AF$14; L58=$AF$15; L59=$AF$16; L60=$AF$17; L61=$AF$18); $AF$3; SE(E(L55=$AH$12; L56=$AH$13; L57=$AH$14; L58=$AH$15; L59=$AH$16; L60=$AH$17; L61=$AH$18); $AH$3; SE(E(L55=$AJ$12; L56=$AJ$13; L57=$AJ$14; L58=$AJ$15; L59=$AJ$16; L60=$AJ$17; L61=$AJ$18); $AJ$3; SE(E(L55=$AL$12; L56=$AL$13; L57=$AL$14; L58=$AL$15; L59=$AL$16; L60=$AL$17; L61=$AL$18); $AL$3; SE(E(L55=$AN$12; L56=$AN$13; L57=$AN$14; L58=$AN$15; L59=$AN$16; L60=$AN$17; L61=$AN$18); $AN$3; SE(E(L55=$AP$12; L56=$AP$13; L57=$AP$14; L58=$AP$15; L59=$AP$16; L60=$AP$17; L61=$AP$18); $AP$3; ”?”))))))))))))))))

Esta função faz uma comparação entre a palavra corrigida e as palavras do código $ham(7,4)$, e retorna o comando referente a palavra escolhida.

17º passo Copiar a célula L64 e colar nas células N64, P64, $\cdots$, AN64 e AP64.

Veja que no exemplo da Figura A.3, várias palavras apresentaram um erro, e mesmo assim a máquina conseguiu recuperar a mensagem.

Por fim, os alunos deverão montar a parte em que será apresentada algumas posições do robô no tabuleiro, podendo assim descrever o caminho percorrido pelo mesmo de maneira visual. O alunos deverão proceder da seguinte maneira:

1º passo Na célula L69 digitar a fórmula:

=SE(L64="SIGA";"-";"Erro")

2º passo Na célula N69 digitar a fórmula:

=SE(OU($L$69="Erro";L$64="SIGA";L$64="PARE";L$64="D";L$64="E";
L$64="B";L$64="C";L$64=0);"-";N$64)

3º passo Nas células N70, N71, $\cdots$, N77, respectivamente, digitar as fórmulas

=SE(OU(L$64=2;L$64=3;L$64=4;L$64=5;L$64=6;L$64=7;L$64=8;L$64=9);
N$69;"-")
=SE(OU(L$64=3;L$64=4;L$64=5;L$64=6;L$64=7;L$64=8;L$64=9);N$69;"-")
=SE(OU(L$64=4;L$64=5;L$64=6;L$64=7;L$64=8;L$64=9);N$69;"-")
=SE(OU(L$64=5;L$64=6;L$64=7;L$64=8;L$64=9);N$69;"-")
=SE(OU(L$64=6;L$64=7;L$64=8;L$64=9);N$69;"-")
=SE(OU(L$64=7;L$64=8;L$64=9);N$69;"-")
=SE(OU(L$64=8;L$64=9);N$69;"-")
=SE(L$64=9;N$69;"-")

4º passo Selecionar as células N70, N71, $\cdots$, N77, copiar e colar nas demais colunas em que aparece o palavra canal, sempre iniciando na linha 70.

5º passo Nas células K81, K82, K83 E K84, respectivamente, digitar as letras maiúsculas D, E, C e B.

6º passo Nas colunas L81, L82, L83 E L84, respectivamente, digitar as fórmulas:

=CONT.SE(L$69:L77;"D")
=CONT.SE(L$69:L77;"E")
=CONT.SE(L$69:L77;"C")
=CONT.SE(L$69:L77;"B")

7º passo Selecionar as células L81, L82, L83, L84, copiar e colar nas demais colunas em que aparece o palavra canal, sempre iniciando na linha 81.

8º passo Digitar a letra X na célula K66 e Y na célula K67.

9º passo Nas células L66 e L67, respectivamente, digitar as fórmulas:

=0,5+L81+L82

=0,5+L83+L84

10º passo Nas células N66 e N67, respectivamente, digitar as fórmulas:

=L66+N81+N82

=L67+N83+N84

11º passo Selecionar as células N66 e N67 copiar e colar nas demais colunas em que aparece o palavra canal, sempre iniciando na linha 66.

12º passo Digitar o número 12 nas células AR66 e AR67.

13º passo Inserir uma imagem com o tabuleiro representado o salão e as mesas, conforme Figura 2.1.

14º passo Selecionar as células K66 até AR67 e inserir um gráfico de dispersão com esses dados.

15º passo formatar o gráfico para que apareça apenas os pontos de coordenadas (x, y).

16º passo ajustar o gráfico e a imagem do tabuleiro para que os pontos apareçam exatamente nas casas do tabuleiro conforme a Figura {graficotabuleiro}.

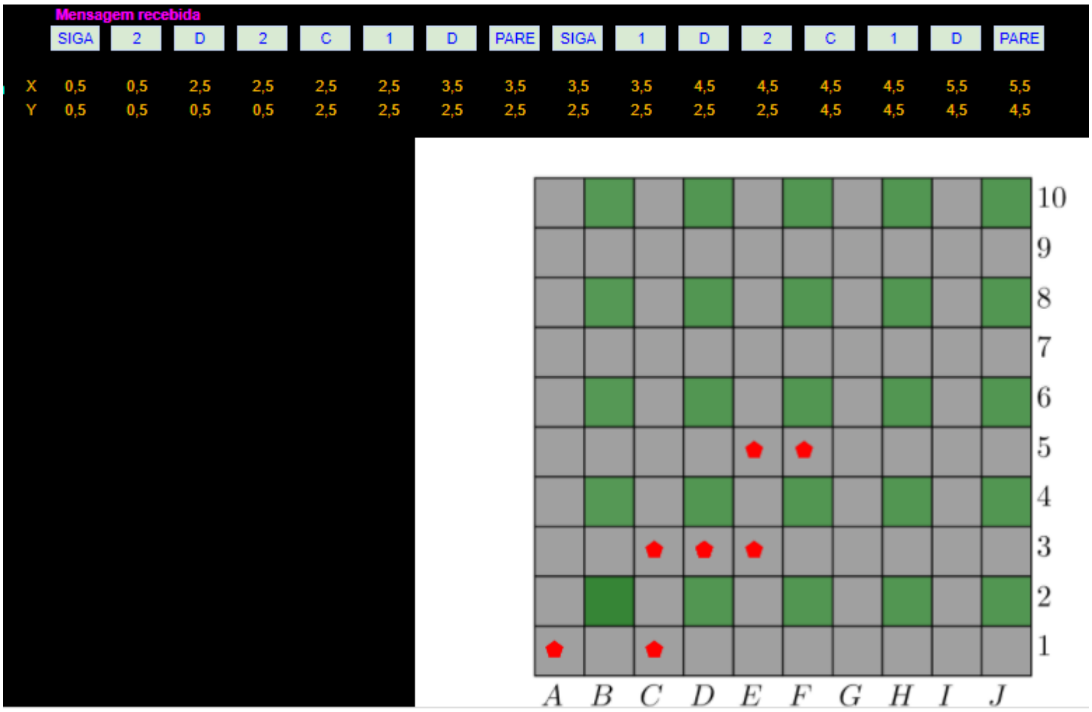
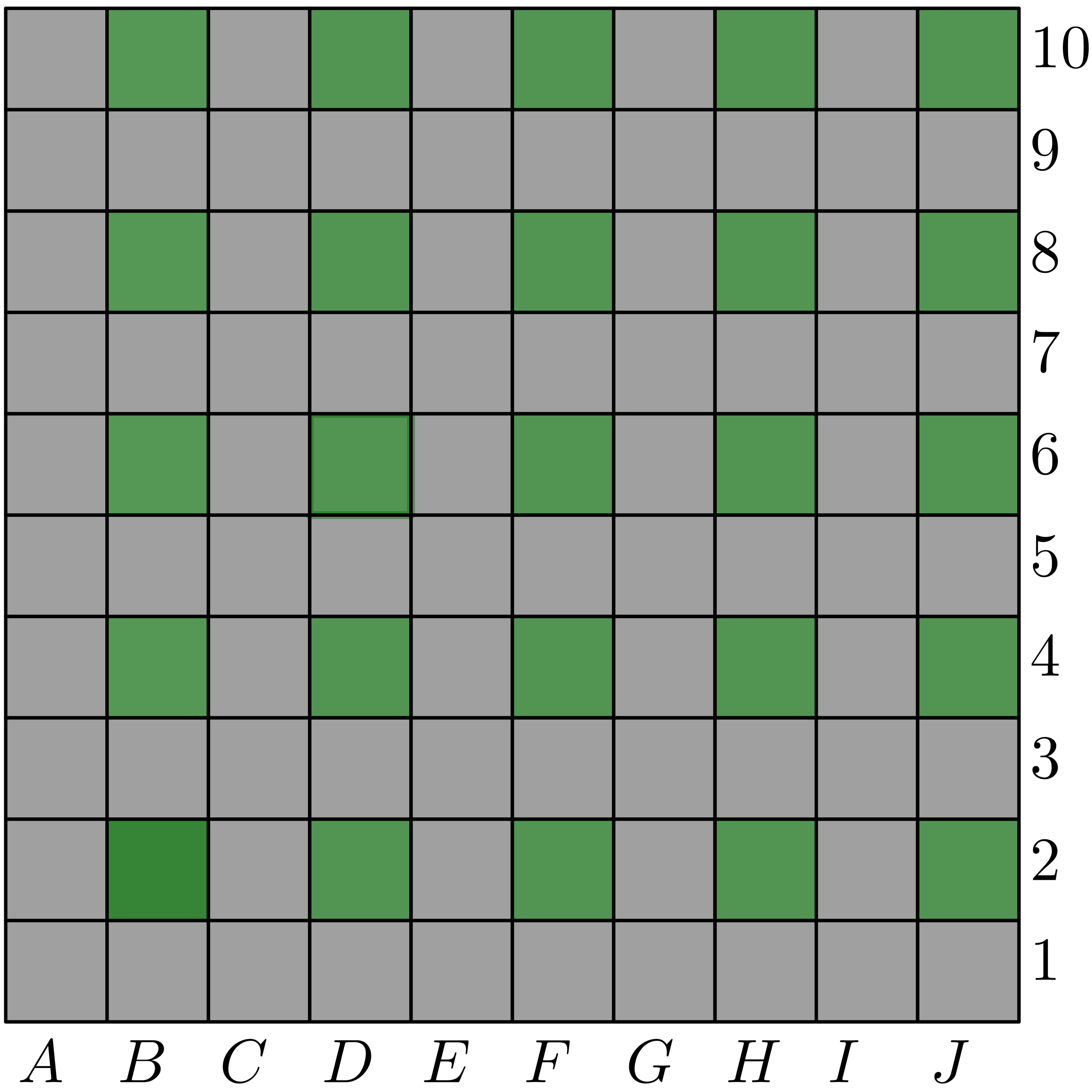


Figura A.4: Representação do movimento do robô.

Apêndice B

Representação do Salão



Apêndice C

Referencial teórico

Neste capítulo apresentaremos uma revisão sobre códigos corretores de erros, códigos de Hamming e alguns conceitos algébricos sobre anéis, corpos e espaços vetoriais. Utilizamos como referências [3] e [5].

C.1 Códigos Corretores de Erros

Imagine que você seja professor de uma turma de $6^{\underline{o}}$ ano e esteja tentando explicar algum conteúdo para os alunos, mas, um grupo de três ou quatro crianças estão conversando baixinho no canto da sala e acabe tirando um pouco da atenção dos colegas. Nessa situação todos nós professores gostaríamos que por meio de alguma mágica fosse possível retirar o som da conversa de modo que apenas a nossa voz chegasse até os ouvidos dos alunos. Poderíamos imaginar que retirar os alunos da sala resolveria o problema e teríamos o caminho livre para o som da nossa fala. No entanto, mesmo tomando esta atitude, provavelmente chegaria ruídos dos corredores, da sala vizinha, dos carros na rua, dentre várias outras fontes de ruídos. De maneira semelhante, quando queremos enviar uma mensagem, seja por sinal analógico ou por sinal digital, existem interferências causadas pelos aparelhos usados na transmissão, pelo meio físico utilizado ou por outras fontes, de modo que é impossível que uma mensagem seja enviada e recebida sem algum tipo de problema causado por algum ruído.

Os ruídos são causados por fontes naturais, como por exemplo o sol, e também por fontes não naturais, como a turbina de um avião, o uso de um aparelho de micro-ondas, e até mesmo pelos próprios aparelhos eletrônicos utilizados na transmissão e recepção dos dados. Ao tentar sintonizar um televisor analógico ou o rádio de um carro em uma determinada frequência, é possível perceber o ruído chegando até o receptor dos aparelhos. Os pequenos chuviscos na TV ou o chiado no rádio, representam a captação de "pequenas" ondas eletromagnéticas que muitas vezes não faziam parte do sinal que foi enviado. Diferentemente do que acontece em sala de aula em muitos dos casos conseguimos fazer com que apenas o sinal enviado chegue até os nossos olhos ou ouvidos. Muitas vezes este problema é resolvido porque o sinal enviado é mais forte que o ruído, de modo que podemos corrigir os erros. O bom desempenho dos aparelhos de correção dos sinais recebidos depende da criação de algoritmos e estes dependem da existência de bons códigos corretores de erros. Um dos exemplos mais simples e corriqueiro é o código de barras. Muitas vezes em um supermercado ao aproximar o código de barras de um leitor óptico o aparelho emite um som informando algum tipo de erro e automaticamente aproximamos o produto novamente. O mesmo acontece quando por algum motivo estamos preenchendo um formulário digital e acabamos informando um dígito do nosso CPF errado. Seria um incomodo ter que pagar mais caro por um produto ou usar um CPF diferente em uma compra pela internet. Esse tipo de situação não ocorre devido ao acréscimo de redundância no código de barras e também no CPF, neste caso a redundância é chamada de dígito verificador. Um código corretor de erros, em poucas palavras e de maneira simples, nada mais é que uma maneira organizada de acrescentar redundâncias a cada informação que queremos transmitir e/ou armazenar, de modo que seja possível utilizar os dados adicionais para recuperar a informação, detectar e corrigir os erros quando possível. Assim como no nosso sistema de escrita, os códigos corretores de erros são formados por um alfabeto, por palavras e por regras de composição das palavras. Normalmente o alfabeto é formado pelos elementos de um corpo finito e as palavras são sequências finitas destes elementos.

De acordo com [4], a Teoria dos Códigos, assim como muitos outros estudos nos diversos campos da Matemática, surgiu em meio a um conflito entre duas grandes nações, a União Soviética e o Estados Unidos durante a Guerra Fria. Em 1948 o Matemático estadunidense Claude Elwood Shannon publicou o primeiro trabalho sobre códigos corretores de erros e ficou conhecido como o pai da teoria da informação. Um outro nome ligado ao inicio da Teoria de Códigos é do Matemático estadunidense Richard Wesley Hamming, que trabalhava com um grande computador e tinha seu trabalho perdido sempre que a máquina cometia algum tipo de erro durante o armazeno de informações. Motivado pela frustração de perder todo seu trabalho sempre que ocorria um erro desse tipo, Hamming desenvolveu um dos primeiros códigos corretores de erros da história e publicou um trabalho em 1950 com conceitos fundamentais como métrica, redundância, equivalência de códigos e códigos sistemáticos. Neste trabalho veremos este código com mais detalhes. A partir da década de 70 a Teoria de Códigos passou a ser de interesse de vários engenheiros que estavam envolvidos na corrida espacial e surgiram vários outros códigos.

No trabalho publicado em 1948, Shannon definiu a unidade de medida de informação, que chamou de bit (binary digits) e um sistema de comunicação, formado basicamente por cinco componentes: uma fonte de informação, um transmissor, um canal, um receptor e um destino, como apresentado no esquema da Figura C.1. O código fonte, ou simplesmente a fonte, produz a mensagem a ser transmitida e envia para o transmissor que transforma a mensagem em um sinal passível de ser transmitido pelo canal, que é o meio utilizado para transmitir o sinal do transmissor até o receptor que decodifica o sinal recebido equivalente a mensagem enviada pelo transmissor, e por fim a informação é visualizada pela pessoa ou equipamento no destino final.

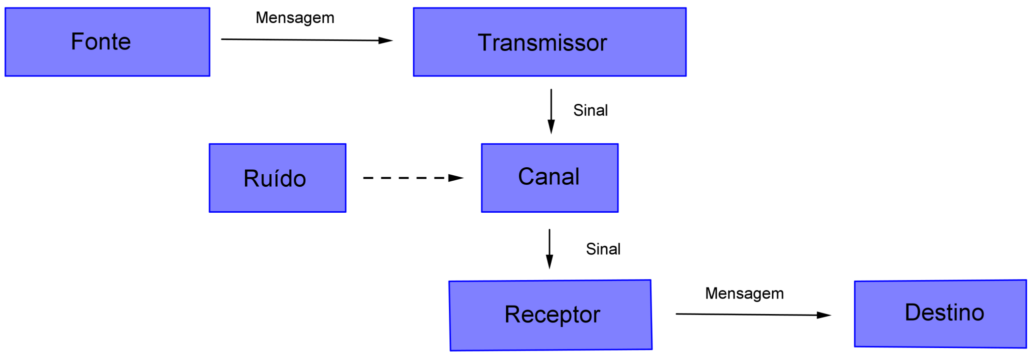


Figura C.1: Esquema do sistema de transmissão.

Um exemplo simples de código corretor de erros é o nosso idioma. Considerando o alfabeto F formado por 23 letras mais o espaço em branco, temos uma total de 24 elementos. A maior palavra do nosso alfabeto é pneumoultramicroscopicossilicovulcanoconiótico, que possui 46 letras, assim, podemos completar as demais palavras com espaços em branco, de modo que todas elas tenham exatamente 46 elementos. Neste caso o nosso código C é um subconjunto de F^{46}, ou seja, é um subconjunto do conjunto formado por todas as sequências com 46 elementos do alfabeto F. Este código não é muito eficiente por ter palavras muito próximas umas das outras, por exemplo, as palavras BALA e MALA diferem de apenas um elemento. Assim o transmissor pode enviar a palavra BALA e, devido a interferência causada por algum ruído, chegar ao destino a palavra MALA, como representado na Figura C.2. Como as duas palavras pertencem ao código o receptor não identificará o erro.

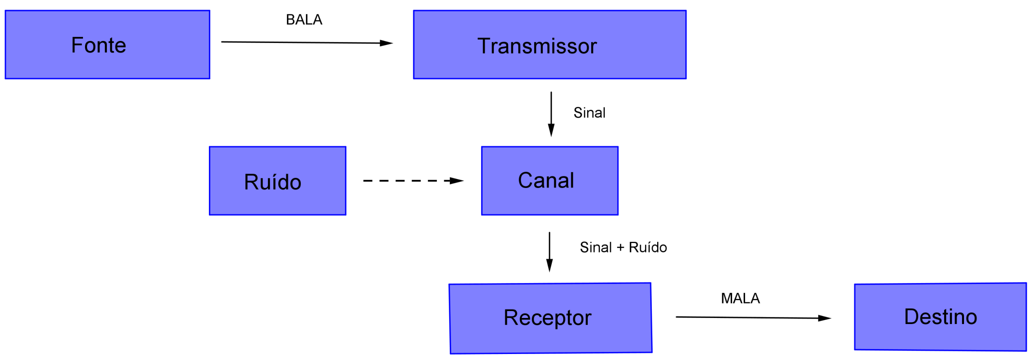


Figura C.2: Palavra recebida com erro.

Existe também a possibilidade de ser enviada a palavra BALA e por algum motivo o receptor detectar um sinal equivalente a palavra QALA. Como esta palavra não existe em nosso idioma, o receptor saberá que a palavra está errada. Mas existem várias outras palavras que estão próxima desta, por exemplo BALA, FALA, MALA e TALA, assim o receptor não saberá qual foi a palavra enviada e não será possível corrigir o erro. Este tipo de impossibilidade é causada devido a distância entre as palavras e a falta de informações adicionais. Veja o esquema da Figura C.2.

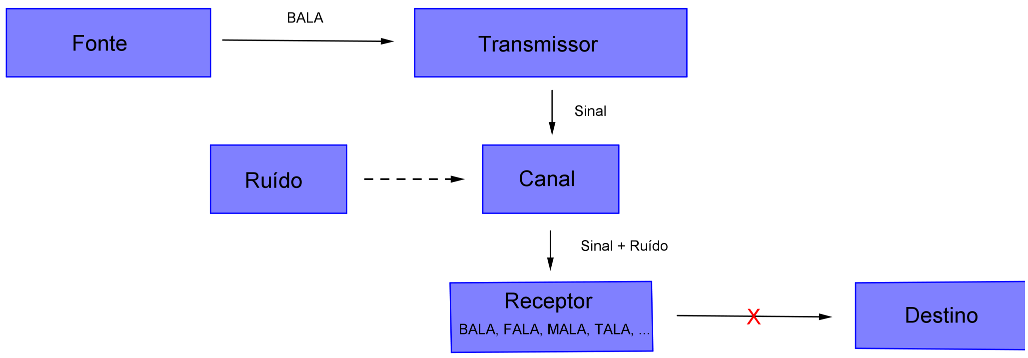


Figura C.3: Erro sem correção.

Para resolver o problema das múltiplas substituições para palavra QALA recebida, poderíamos ter acrescentado algum tipo de redundância no sinal enviado, por exemplo, ter enviado o sinal da palavra CARAMELO junto com o sinal da palavra BALA, desta maneira, como mostra o esquema da Figura C.4, o receptor iria detectar o sinal das palavras QALA e CARAMELO. Como as demais palavras possíveis para a correção não tem nenhum tipo de relação com a palavra CARAMELO, o receptor iria decidir corretamente por enviar a palavra BALA para o destino final. Mas note que poderíamos ter escolhido a palavra DOCE como redundância, aumentando assim apenas quatro elementos na sequência da palavra BALA. Quanto maior o comprimento das palavras, maior o custo computacional para enviar e receber as mesmas, por isso devemos nos preocupar em como acrescentar redundância nas palavras mantendo o comprimento razoavelmente pequeno.

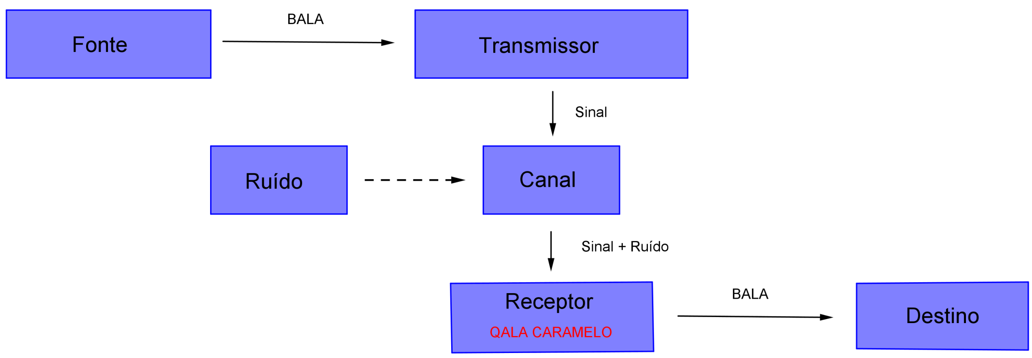


Figura C.4: Acréscimo de redundância e correção do erro.

Um outro exemplo clássico é o código fonte $F_2^2 = \{(0,0),(0,1),(1,0),(11)\}$ de um robô que se move sobre um tabuleiro quadriculado de modo que a cada comando dado o robô se desloca de uma casa para a casa vizinha. Por simplicidade vamos representar o par ordenado (a,b) simplesmente por ab, aqui cada palavra ab representa um comando, como descrito abaixo:

$$\begin{array}{rcl@{\qquad\qquad}rcl} \text{Leste} & \longrightarrow & 00 & \text{Norte} & \longrightarrow & 10 \\ \text{Oeste} & \longrightarrow & 01 & \text{Sul} & \longrightarrow & 11 \end{array}$$

Neste caso não temos um bom código, pois ao enviar a palavra 11 o robô poderia receber a palavra 10 e não perceber que foi cometido um erro, como no caso das palavras BALA e MALA. Precisamos então acrescentar redundâncias a cada palavra deste código de maneira organizada para que seja possível identificar e corrigir erros. Ao fazer esse acréscimo criamos uma novo código conhecido como código do canal. Podemos então, fazer a seguinte modificação no código F_2^2:

$$\begin{array}{rcl@{\qquad\qquad}rcl} 00 & \longrightarrow & 00000 & 10 & \longrightarrow & 10110 \\ 01 & \longrightarrow & 01011 & 11 & \longrightarrow & 11101 \end{array}$$

Neste caso os dois primeiros elementos são idênticos ao que já tínhamos no código F_2^2 e os últimos são redundâncias. Assim, o código do canal é $\{00000,\ 01011,\ 10110,\ 11101\}$.

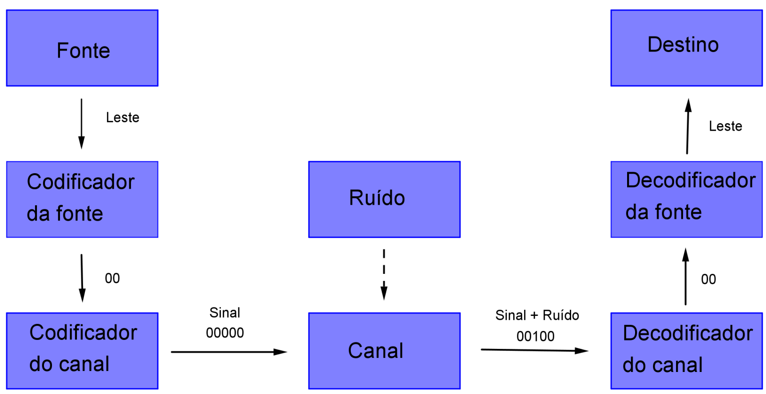


Figura C.5: Transmissão do comando Leste com correção.

Com esse novo código fica mais fácil de corrigir possíveis erros. Por exemplo, suponha que o comando Leste deva ser enviado para o robô, primeiramente o codificador da fonte gera a palavra 00 que será transformada na palavra 00000 pelo codificador do canal e em seguida enviada. Assim, o decodificador do canal receberá um palavra que pode ser a palavra correta ou uma palavra com erro, vamos supor que o decodificador receba a palavra 00100. Ao comparar esta palavra com as demais palavras do código, o decodificador perceberá que a mesma não pertence ao código do canal, logo, concluir que ocorreu algum tipo de erro na transmissão e analisar se é possível fazer a correção. Procurando pelas palavras mais próximas, em relação a diferença entre os elementos de mesma posição, o decodificador chegará a conclusão de que a palavra enviado foi 00000 e desta maneira o robô receberá corretamente o comando Leste, como representado na Figura C.1.

Dado um alfabeto F, isto é, um conjunto com finitos elementos, denotamos o número de elementos do conjunto F por $|F|$. Em muitos dos casos o conjunto F é um corpo. Um código corretor de erros é qualquer subconjunto próprio de F^n, ou seja, é um conjunto formado por sequência com exatamente n termos, todos pertencentes ao alfabeto F.

Definição C.1. Dados dois elementos quaisquer $u, v \in F^n$, a distância de Hamming entre $u = (u_1, u_2, ...u_i, ..., u_n)$ e $v = (v_1, v_2, ..., v_i, ..., v_n)$ é definida como

$$d(u, v) = |\{i;\ u_i \neq v_i,\ 1 \leq i \leq n\}|.$$

Exemplo C.2. No código $\{00000,\ 01011,\ 10110,\ 11101\} \subset \{0, 1\}^5$, temos que

$$\begin{array}{rclcrcl} d(00000, 01011) & = & 3, & \qquad & d(01011, 10110) & = & 4, \\ d(00000, 10110) & = & 3, & & d(01011, 11101) & = & 3, \\ d(00000, 11101) & = & 4, & & d(10110, 11101) & = & 3. \end{array}$$

A distância de Hamming satisfaz as três propriedades de métrica, como veremos na proposição a seguir, por esse motivo a distância entre duas palavras como definida acima, também é conhecida como métrica de Hamming.

Proposição C.3. *Dados $u,\ v,\ w \in F^n$, temos que*

i) Positividade: $d(u, v) \geqslant 0$, valendo a igualde se, e somente se, $u = v$;

ii) Simetria: $d(u, v) = d(v, u)$;

iii) Desigualdade triangular: $d(u, w) \leqslant d(u, v) + d(v, w)$

Demonstração. O item *i)* é claro, pois o número de elementos de um conjunto é não negativo. O item *ii)* segue do fato de que $u_i \neq v_i$ se, e somente se, $v_i \neq u_i$.

Para demonstrar o item *iii)*, vamos analisar a contribuição da i-ésima coordenada para $d(u, w)$ e para a soma $d(u, v) + d(v, w)$. Observe que se $u_i \neq w_i$, então a contribuição desta coordenada para $d(u, w)$ é 1 e para $d(u, v) + d(v, w)$ é 1 ou 2, pois não podemos ter $u_i = v_i = w_i$. No caso em que $u_i = w_i$ a contribuição para $d(u, w)$ é 0 e para $d(u, v) + d(v, w)$ pode ser 0 ou 2, pois, ou $u_i = w_i = v_i$ ou $v_i \neq u_i$ e $v_i \neq w_i$. Portanto, a contribuição da i-ésima coordenada para $d(u, w)$ é menor ou igual a contribuição para $d(u, v) + d(v, w)$. Como essas distâncias podem ser obtidos somando as contribuições de cada coordenada, chegamos a conclusão que $d(u, w) \leqslant d(u, v) + d(v, w)$. $\square$

Muitos dos métodos de correção de erros são baseados na distância de Hamming entre duas palavras. Um aspecto que pode tornar o processo de correção mais eficiente é a distância mínima do código. Esta característica está intimamente ligada com a quantidade de erros que é possível corrigir. Por exemplo, no código do Exemplo C.2, temos que a distância mínima é 3 e como veremos mais na frente, neste caso é possível corrigir no máximo um erro.

Definição C.4. Seja $\mathcal{C}$ um código, a distância miníma d de $\mathcal{C}$ é definida por

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

Observe que no Exemplo C.2 calculamos a distância entre todas as palavras diferentes, fizemos um total de $\binom{4}{2} = 6$ operações. Assim, pode-se imaginar que para todo código $\mathcal{C}$, é necessário realizar $\binom{|\mathcal{C}|}{2}$ cálculos para determinar a distância mínima d. Mas, como veremos mais adiante, no caso de códigos lineares a quantidade de operações é certamente menor.

Definição C.5. Dado um código $\mathcal{C}$ com distância mínima d definimos a cota de correção t como sendo o número natural

$$t = \left[\frac{d-1}{2}\right],$$

onde $[k]$ representa a parte inteira de um elemento k.

A proposição a seguir fornece dados numéricos relacionados a capacidade de correção de um código corretor de erros.

Proposição C.6. *Seja $\mathcal{C}$ um código com distância mínima d. Então $\mathcal{C}$ pode corrigir no máximo $t = \left[\frac{d-1}{2}\right]$ erros e detectar $d-1$ erros.*

Veja que de acordo esta proposição, para maximizar a capacidade de correção de um código $\mathcal{C}$ precisamos escolher as palavras de modo que a distância mínima d seja maximizada, e desta maneira aumentar a capacidade de correção do código. Portanto, é de fundamental importância determinar a distância mínima de um código ou ter pelo menos uma estimativa para a mesma.

Exemplo C.7. Considere o código $\mathcal{C} = \{000, 111\} \subset \{0, 1\}^3$. Veja que neste código temos distância mínima $d = 3$, capacidade de correção $t = \left[\frac{3-1}{2}\right] = 1$,

Observe que a eficiência deste método de correção de erros depende muito do canal que será utilizado para enviar o sinal e da distância mínima entre as palavras do código. Caso a probabilidade de erros seja pequena, podemos receber palavras com no máximo t erros, assim seremos capazes de fazer a correção do erro sem incertezas, no entanto, caso a quantidade de erros seja maior que t podemos corrigir a palavra de forma incorreta. Portanto, nunca teremos certeza que não houve erro no processo de correção, o máximo que teremos é uma probabilidade de acerto.

C.2 Código de Hamming

Os códigos de Hamming foram desenvolvidos por Richard W. Hamming por volta de 1950. Trata-se de uma família de códigos binários que são utilizados até hoje devido a sua baixa complexidade e é utilizado para a correção de erros simples. Nos códigos de Hamming temos os seguintes parâmetros:

- O comprimento n das palavras é dado por $n = 2^r - 1$, com $r > 1$;
- O número de bits de informação é $k = 2^r - (r + 1)$;
- O número de bits de paridade (redundância) é $r = n - k$.

Desta maneira, ao falar do código de Hamming (n, k), ou simplesmente $ham(n, k)$, estamos nos referindo ao código cujas palavras têm comprimento n, sendo k dígitos de informação e $n - k$ dígitos de paridade.

O código de Hamming (n, k) é um subconjunto de $F_2^n = \{0, 1\}^n$. Assim, por exemplo, uma palavra será representada por $(0, 0, 1, 1, 0, 0, 1)$, ou simplesmente por 0011001.

Dada uma palavra $x = (x_1, x_2, x_3, \cdots, x_n) \in ham(n, k)$, temos que a i-ésima coordenada será um dígito de paridade P_α se $i = 2^{\alpha-1}$, com $\alpha \in \mathbb{N}$, e será um dígito de informação caso contrário. Por exemplo, no caso da palavra $x = (x_1, x_2, x_3, x_4, x_5, x_6, x_7) \in ham(7, 4)$, temos que x_1, x_2 e x_4 são os dígitos de paridades P_1, P_2 e P_3, respectivamente, e x_3, x_5, x_6 e x_7 são os dígitos de informação D_1, D_2, D_3 e D_4, respectivamente.

Considere a palavra $x = (x_1, x_2, x_3, \cdots, x_n) \in ham(n, k)$, temos que cada posição i, com $i \in \{1, 2, 3, \cdots, n\}$, tem uma representação binária, ou ainda, cada uma dessas posições podem ser representadas por uma sequência de r dígitos. No caso da representação das potências de 2, nesta sequência existe uma posição em que o dígito é igual a 1 e os demais são todos iguais a zero, isto é, dada a potência $2^{\alpha-1}$, temos que, em sua representação binária, o dígito da $(r-\alpha+1)$-ésima posição será igual a 1 e todos as demais iguais a 0. Por exemplo, no caso em que $n = 7$, temos $r = 3$, assim, o número $4 = 2^{3-1}$ pode ser representado por 100_2.

Seja $i = 2^{\alpha-1}$, com $\alpha \in \{1, 2, \cdots, r\}$ e $B_{j1}B_{j2}B_{j3} \cdots B_{jr}$ a representação binária da posição j. O dígito de paridade P_α que aparece na posição i da palavra v é dado pela soma de todos os dígitos de informação D_θ, que aparecem numa posição j tal que $B_{j(r-\alpha+1)} = 1$.

Exemplo C.8. Considere a palavra $x = (x_1, x_2, x_3, x_4, x_5, x_6, x_7) \in ham(7,4)$. Escrevendo os índices na base binária temos que,

Posição na base decimal	1	2	3	4	5	6	7
Posição na base binária	001_2	010_2	011_2	100_2	101_2	110_2	111_2
Dígito	x_1	x_2	x_3	x_4	x_5	x_6	x_7
Tipo de dígito	P_1	P_2	D_1	P_3	D_2	D_3	D_4

.

Observe que $1 = 00\mathbf{1}_2$, assim o primeiro dígito de paridade P_1 é dado pela soma de todas os dígitos de informação que aparecem em posições cuja representação na base binária apresenta o dígito 1 na posição mais a direita, que neste caso são, $3 = 01\mathbf{1}_2$, $5 = 10\mathbf{1}_2$ e $7 = 11\mathbf{1}_2$. Temos então que,

$$P_1 = x_3 + x_5 + x_7.$$

De modo análogo, temos

$$\begin{aligned} P_2 &= x_3 + x_6 + x_7, \\ P_3 &= x_5 + x_6 + x_7. \end{aligned}$$

Assim, para enviar a mensagem 1001 com 4 bits de informação usando $ham(7,4)$, temos que a palavra a ser enviada é dada por

x_1	x_2	x_3	x_4	x_5	x_6	x_7
P_1	P_2	1	P_3	0	0	1

.

Neste caso temos,

$$\begin{aligned} P_1 &= 1 + 0 + 1 = 0; \\ P_2 &= 1 + 0 + 1 = 0; \\ P_3 &= 0 + 0 + 1 = 1. \end{aligned}$$

De onde segue que a palavra a ser enviada é dada por,

x_1	x_2	x_3	x_4	x_5	x_6	x_7
0	0	1	1	0	0	1

.

No exemplo anterior, uma maneira fácil e prática de obter a palavra codificada 0011001 seria por meio da matriz da multiplicação de matrizes. Por exemplo, dada uma palavra qualquer $D_1D_2D_3D_4$ a ser transmitida, temos que a palavra $x = (x_1, x_2, x_3, x_4, x_5, x_6, x_7) \in ham(7,4)$ é dada por,

$$\underset{X_{7\times1}}{\begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \\ v_7 \end{bmatrix}} = \underset{X_{7\times1}}{\begin{bmatrix} 1\cdot D_1 & + & 1\cdot D_2 & + & 0\cdot D_3 & + & 1\cdot D_4 \\ 1\cdot D_1 & + & 0\cdot D_2 & + & 1\cdot D_3 & + & 1\cdot D_4 \\ 1\cdot D_1 & + & 0\cdot D_2 & + & 0\cdot D_3 & + & 0\cdot D_4 \\ 0\cdot D_1 & + & 1\cdot D_2 & + & 1\cdot D_3 & + & 1\cdot D_4 \\ 0\cdot D_1 & + & 1\cdot D_2 & + & 0\cdot D_3 & + & 0\cdot D_4 \\ 0\cdot D_1 & + & 0\cdot D_2 & + & 1\cdot D_3 & + & 0\cdot D_4 \\ 0\cdot D_1 & + & 0\cdot D_2 & + & 0\cdot D_3 & + & 1\cdot D_4 \end{bmatrix}} = \underset{G_{7\times4}}{\begin{bmatrix} 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}} \cdot \underset{D_{4\times1}}{\begin{bmatrix} D_1 \\ D_2 \\ D_3 \\ D_4 \end{bmatrix}}.$$

Veja que qualquer palavra $D_1D_2D_3D_4$ pode ser codificada apenas multiplicando a matriz G pela matriz D. Dizemos que G é a matriz geradora do código $ham(7,4)$.

Em geral, os códigos de Hamming tem distância mínima 3, assim, pela Proposição C.6, temos que o código é capaz de identificar até 2 erros e corrigir no máximo 1. No caso em que ocorre apenas um erro durante a transmissão, o código é capaz de detectar a posição exata onde ocorreu o erro, permitindo assim que seja feita a correção.

Para detectar o erro, no processo de decodificação é realizado o teste de verificação dos bits de paridade da palavra recebida. Sabendo exatamente como foi obtido cada um dos dígitos de paridades P_α, o decodificador consegue testar a paridade do dígito recebido. Como os dígitos são elementos do conjunto F_2, para verificar a paridade basta somar o dígito recebido com o dígito gerado a partir da palavra recebida.

Por exemplo, para verificar a paridade do dígito P_1, o decodificador soma o dígito x_1 da palavra recebida com todos os dígitos das posições i para as quais a representação binária apresenta 1 no dígito mais a direita. Assim, caso tenha ocorrido um erro em uma das posições i o resultado será diferente de 0. E desta maneira, o erro terá ocorrido em uma posição na qual o último dígito da representação decimal é igual a 1. De modo análogo, o dígito P_2 pode ser verificado e caso o resultado seja diferente de 0, saberemos que o penúltimo dígito da representação binária da posição onde ocorreu o erro também é igual a 1. Seguindo esse processo é possível encontrar a representação binária exata da posição onde o erro ocorreu.

Exemplo C.9. Suponha que, ao enviar a palavra 0011001 tenha ocorrido um erro, de modo que a palavra recebida foi 0011011. Para detectar o erro é necessário realizar o teste de verificação de bits paridade. Neste caso, temos

	x_1		x_2		x_3		x_4		x_5		x_6		x_7		
Palavra recebida	0		0		1		1		0		1		1		
Teste P_1	$1 \cdot 0$	$+$	$0 \cdot 0$	$+$	$1 \cdot 1$	$+$	$0 \cdot 1$	$+$	$1 \cdot 0$	$+$	$0 \cdot 1$	$+$	$1 \cdot 1$	$=$	0
Teste P_2	$0 \cdot 0$	$+$	$1 \cdot 0$	$+$	$1 \cdot 1$	$+$	$0 \cdot 1$	$+$	$0 \cdot 0$	$+$	$1 \cdot 1$	$+$	$1 \cdot 1$	$=$	1
Teste P_3	$0 \cdot 0$	$+$	$0 \cdot 0$	$+$	$0 \cdot 1$	$+$	$1 \cdot 1$	$+$	$1 \cdot 0$	$+$	$1 \cdot 1$	$+$	$1 \cdot 1$	$=$	1

.

Assim, identificando por T_α o resultado do teste de paridade para o dígito P_α, temos que a palavra $T_3T_2T_1$, conhecida como síndrome, determina a posição em que ocorreu o erro, ou seja, o número binário $T_3T_2T_1$ representa exatamente a posição onde existe um erro. Neste exemplo, temos que $110_2 = 6$, assim, o decodificador fará uma alteração no dígito x_6 e concluirá corretamente que a palavra enviada foi 0011001.

No exemplo, poderíamos ter obtido a síndrome $T_3T_2T_1$ por meio da multiplicação de matrizes. Em geral, dada a palavra $x = (x_1, x_2, x_3, x_4, x_5, x_6, x_7) \in ham(7,4)$, temos que

$$\underbrace{\begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}}_{H_{7\times 3}} \cdot \underbrace{\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix}}_{X_{7\times 1}} = \underbrace{\begin{bmatrix} T_1 \\ T_2 \\ T_3 \end{bmatrix}}_{T_{3\times 1}}.$$

Assim, dada qualquer palavra x, podemos obter a síndrome multiplicando a matriz H pela matriz X. Dizemos que H é a matriz teste de paridade do código $ham(7,4)$.