UNIVERSIDADE FEDERAL DO RECÔNCAVO DA BAHIA
CENTRO DE CIÊNCIAS EXATAS E TECNOLÓGICAS
MESTRADO EM MATEMÁTICA

ARITMÉTICA MODULAR NOS CÓDIGOS DE BARRAS

LUCIANA MOTA CERQUEIRA

CRUZ DAS ALMAS
2015

ARITMÉTICA MODULAR NOS CÓDIGOS DE BARRAS

LUCIANA MOTA CERQUEIRA

Trabalho de conclusão de curso apresentado ao curso de Mestrado Profissional em Matemática do Centro de Ciências Exatas e Tecnológicas da Universidade Federal do Recôncavo da Bahia e a Sociedade Brasileira de Matemática, como parte dos requisitos para a obtenção do título de mestre.

Orientador: Profº Dr. Eleazar Gerardo Madriz Lozada

CRUZ DAS ALMAS
2015

FICHA CATALOGRÁFICA

C416a

Cerqueira, Luciana Mota.
Aritmética modular nos códigos de barras / Luciana Mota Cerqueira._ Cruz das Almas, BA, 2015.
63f.; il.

Orientador: Eleazar Gerardo Madriz Lozada.

Dissertação (Mestrado) – Universidade Federal do Recôncavo da Bahia, Centro de Ciências Exatas e Tecnológicas.

1.Matemática – Aritmética. 2.Matemática – Códigos de barras. I.Universidade Federal do Recôncavo da Bahia, Centro de Ciências Exatas e Tecnológicas. II.Título.

CDD: 510

Ficha elaborada pela Biblioteca Universitária de Cruz das Almas - UFRB.

ARITMÉTICA MODULAR NOS CÓDIGOS DE BARRAS

LUCIANA MOTA CERQUEIRA

Trabalho de conclusão de curso apresentado ao curso de Mestrado Profissional em Matemática do Centro de Ciências Exatas e Tecnológicas da Universidade Federal do Recôncavo da Bahia e a Sociedade Brasileira de Matemática, como parte dos requisitos para a obtenção do título de mestre.

Banca Examinadora:

Presidente da Banca: Eleazar

Profº Dr. Eleazar Gerardo Madriz Lozada - UFRB

Primeiro Membro: Haroldo G. Benatti

Profº Dr. Haroldo Gonçalves Benatti - UEFS

Segundo Membro: Adson Mota Rocha.

Profº MSc. Adson Mota Rocha - UFRB

Cruz das Almas, 15 de Dezembro de 2015.

Aos meus pais,
às minhas irmãs, meu irmão e à minha noiva
com muito amor.

"Agradeço todas as dificuldades que enfrentei;
não fosse por elas,
eu não teria saído do lugar.
As facilidades nos impedem de caminhar.
Mesmo as críticas,
pois nos auxiliam muito."
Chico Xavier

AGRADECIMENTOS

A Deus, fonte de Luz e Sabedoria, que sempre está ao meu lado, dando-me força e perseverança em minhas realizações.

Aos meus pais, que sempre foram alicerces em minha jornada. Às minhas irmãs Sandra e Lucélia, pelo amor e companheirismo, e ao meu irmão Luciano Júnior, por caminharmos lado-lado nessa batalha, ajudando-me e compartilhando das mesmas emoções.

À minha noiva Jamile, pela paciência, cumplicidade, companheirismo e sabedoria.

Ao meu orientador professor Eleazar Madriz, pela paciência, competência e sabedoria.

A todos os professores do PROFMAT, pólo UFRB, que participaram desse projeto, pela competência, paciência e dedicação.

Aos meus queridos colegas de turma do PROFMAT 2012, pelo incentivo, carinho, companheirismo, alegrias e pelas trocas valiosas de experiências.

A todos amigos e colegas que direta ou indiretamente contribuíram para a realização deste trabalho, os meus sinceros agradecimentos.

RESUMO

Devido às suas características de baixo custo, facilidade de implantação, utilização e, principalmente, a qualidade de informação e redução de tempo de operação dos sistemas, os códigos de barras têm sido usados desde pequenas atividades que necessitam de identificação de documentos à comercialização e controle de produtos. Daí a utilização dos códigos de barras é algo pertinente ao nosso dia-a-dia, bem como no âmbito escolar. Identificar, através do número do código de barras, a origem, o fabricante e o tipo de produto, são tarefas que podem ser aplicadas na escola. Os códigos de barras, no Brasil utilizam-se os códigos EAN-13, que contêm 13 dígitos, sendo o último dígito aquele que verifica os 12 anteriores, de modo que, se um desses for digitado incorretamente, certamente o produto não será identificado. Isso é devido ao processo de verificação do erro, onde utiliza-se a matemática, em especial a aritmética modular e o produto vetorial. Neste trabalho evidenciamos como os códigos de barras são uma aplicação da matemática interessante, que pode ser aplicada, de forma fácil e estimuladora, aos alunos dos ensinos fundamental II e médio.

Palavras-chave: Códigos de barras, Dígito de verificação, Aritmética Modular

ABSTRACT

Due to its characteristics: low price, facility of implementation and use and, mainly, the quality of information and reduction of operation time of systems, the bar codes are being used for small and big activities such as the ones that need identification documents and commercialization and control of products. This way bar codes usage is something very important to our routine and school environment. Identify the origin, producer and kind of product through bar codes are tasks that can be applied at school. The bar codes used in Brazil are codes EAN-13, with 13 digits, the last one being the digit that verifies the other 12 before, in a way that if one of these first digits is typed in a wrong way, certainly the product is not going to be identified. This is due to the verification process of the mistake, where we use the mathematics, especially the modular arithmetic and the vector product. In this project we enhance how interesting is the mathematics application of bar codes, being applied in an easy and stimulating way to elementary and high school students.

Keywords: Bar codes, Verification digit, Modular arithmetic.

SUMÁRIO

LISTA DE FIGURAS

INTRODUÇÃO

O surgimento dos códigos de barras devem-se, em especial, às atividades comerciais, onde o preço de cada produto era colocado manualmente e, quando vendido, tinha que ser digitado numa máquina pela operadora, o que ocasionava um processo lento e passível a erros. Daí, diante desse problema de logística, um dono de supermercado sugeriu a uma universidade que resolvesse esse problema, surgindo assim, nessa busca, o código de barras.

Por causa das suas características de baixo custo, facilidade de implantação, utilização e principalmente a qualidade de informação e redução de tempo de operação dos sistemas, o código de barras tem sido usado na automação desde pequenas atividades que necessitam de identificação de documentos até a comercialização e controle de produtos.

O surgimento do primeiro código de barras data de Outubro de 1949. Era formado por quatro linhas brancas sobre um fundo preto, depois convertido em círculos com mesmos centros para facilitar a leitura, a partir de qualquer ângulo. Quanto mais linhas se adicionassem, mais informação podia ser codificada. Assim, em 1952 a primeira patente de um código de barras foi registrada por Bernard Silver e Norman Joseph Woodland.

Porém, esse acontecimento só teve seu real valor reconhecido quando, após várias décadas, o aumento dos componentes eletrônicos e avanços na tecnologia a laser permitiram a produção de sistemas de leitura de baixo custo.

Dentre as diversas firmas solicitadas para que elaborassem um código adequado e viável, que pudesse ser utilizado no dia-a-dia, quem acabou apresentando o resultado mais viável foi Gerge J. Laurer, engenheiro do International Business Machines (IBM).

O primeiro código de barras foi utilizado em 1974 e foi chamado de Código Universal de Produtos, com sigla UPC (Universal Product Code).Este código é adotado atualmente nos Estados Unidos e Canadá. Ele consiste de uma sequência de 12 dígitos, traduzidos para barras da forma que estamos acostumados a ver em vários objetos ou itens.

Em decorrência do grande sucesso do código UPC, fabricantes e distribuidores de vários países da Europa formaram um conselho para estudar a possibilidade de desenvolver um sistema que padronizasse a numeração de produtos, parecido ao sistema do UPC, que é regulamentado pelo UCC (Uniform Code Council). Em 1977, formou-se uma entidade sem fins lucrativos, a EAN (European Article Numbering Association) para atender esta demanda.

A elaboração do código EAN deparou-se com um problema bastante delicado, que era o de ser necessário adicionar um dígito a cada código com a finalidade de identificar o país de origem, já que tal tecnologia seria expandida por todo o mundo. O problema era fazer isto de forma que a mesma máquina leitora pudesse ler, sem distinção, os códigos UPC e EAN.

A solução encontrada foi à seguinte: os países que utilizavam o código UPC antigo, EUA e Canadá, são identificados com um 0 (zero), na frente, resultando no novo código UPC-A (o mesmo código UPC, apenas com um zero antes para identificar os países que já utilizavam o código UPC, Estados Unidos e Canadá), e o resto da codificação é feita utilizando o sistema anterior. Para outros países, os primeiros dois ou três dígitos (da esquerda para a direita), identificam o país.

O código utilizado no Brasil é o EAN (European Article Number), que possui os primeiros três dígitos identificando o país. Todos os produtos produzidos no Brasil começam com a sequência 789. Alguns países adotam este mesmo sistema, dando-lhe outro nome. Por exemplo, no Japão o sistema é conhecido como JAN (Japanese Article Numbering).

Observando que a matemática presente nos códigos de barras é a mesma apresentada nos

ensinos fundamental e médio e acreditando que contextualizando os conceitos matemáticos com o objeto de estudo, algo que já faz parte do cotidiano do aluno, é possível por em prática, nas salas de aula, algo abstrato.

A utilização dos códigos de barras como objeto de pesquisa e de aprendizagem matemática é rica no sentido de envolver alunos em diferentes níveis de ensino.

Neste trabalho utilizaremos apenas os códigos de barras que identificam produtos, que são os códigos UPC-A e EAN-13.

Este trabalho está dividido em 4 capítulos, sendo o primeiro destinado aos conteúdos matemáticos fundamentais para o desenvolvimento desse trabalho. Tais como Divisibilidade, Máximo Divisor Comum, Números Primos, Congruências e Produto Escalar Vetorial.

O segundo capítulo é destinado aos Códigos de Barras, contendo: Definições e composição de um código de barras; Como são feitas as leituras desses códigos de barras e o processo de codificação dos códigos UPC-A e EAN-13.

O terceiro capítulo é destinado à detecção de erros, contendo os Sistemas de Identificação Modular (SIM) e sobre o Dígito Verificador .

No quarto capítulo são propostas atividades com o tema, incluindo sequências didáticas e suas respectivas soluções, que podem ser aplicadas para alunos do ensino fundamental II e médio.

CAPÍTULO 1

PRELIMINARES

PARA a realização deste trabalho, faz-se necessário definir alguns conceitos e propriedades da Teoria dos Números, em especial a Aritmética Modular. As definições e propriedades aqui definidas e demonstradas estão de acordo com [1] , [2], [3] e [4]. Denotaremos por $\mathbb{Z}$ o conjunto dos números inteiros e consideremos suas operações de adição e multiplicação, com sua relação de ordem e propriedades.

1.1 Divisibilidade

O conceito de divisibilidade e suas propriedades serão fundamentais para o desenvolvimento deste trabalho.

Definição 1.1. *(Módulo ou Valor Absoluto) Para todo $m \in \mathbb{Z}$, define-se módulo ou valor absoluto de m, denota-se $|m|$, através das seguintes condições:*

$$|m| = m, \text{ se } m \geqslant 0$$

$$e$$

$$|m| = -m, \text{ se } m < 0$$

Em particular, $|m| = 0$ se, e somente se, $m = 0$.

Definição 1.2. *Sejam dois números inteiros x e y, com $x \neq 0$, diz-se que x divide y, escreve-se $x|y$, se existe um inteiro z tal que $y = xz$. Podemos escrever, também, que x é divisor de y ou y é divisível por x.*

No caso em que x não divide y, escreve-se $x \nmid y$.

Exemplo 1.1. *Podemos afirmar que $3|6$ pois, $6 = 2.3$, enquanto $3 \nmid 7$ pois não existe nenhum inteiro z que satisfaça a igualdade $7 = 3.z$.*

Nas proposições a seguir, apresentamos as propriedades da divisibilidade que serão utilizadas nesse trabalho.

Proposição 1.1. *Sejam os inteiros x, y e z, com $x \neq 0$. Tem-se que:*

- *i) $x|x$*
- *ii) se $x|y$ e $y|x$, então $|x| = |y|$*
- *iii) se $x|y$ e $y|z$, então $x|z$*
- *iV) se $x|y$ e $x|z$, então $x|(y + z)$*
- *v) se $x|y$, $x|z$ e $y \geqslant z$, então $x|(y - z)$*

Demonstração:
i), ii) e iii) Ver propriedades d_1,d_2 e d_3 em [1], página 31.

iv) e v) Ver proposição 3.1.3 e 3.1.4 em [2] páginas 31 e 32.

■

Proposição 1.2. *Sejam os inteiros a, b, x, y e z, com $x \neq 0$. Tem-se que:*

- *i) Se $x|y$, então $x|yz$.*
- *ii) Se $x|y$ e $x|z$, então $x|(ay + bz)$.*

Demonstração:

i) Se $x|y$, implica que existe um inteiro k tal que $y = x.k$. Multiplicando-se a igualdade por z, teremos:

$$y.z = x.k.z \Rightarrow y.z = x.(k.z) \Rightarrow x|yz$$

ii) Pela hipótese, $x|y$ implica que existe um inteiro k_1 tal que $y = k_1.x$ e $x|z$ implica que existe um inteiro k_2 tal que $z = k_2.x$. Daí, temos:

$$y = k_1.x \Rightarrow ay = ak_1x$$

e

$$z = k_2.x \Rightarrow bz = bk_2x$$

Somando-se as igualdades, teremos:

$$ay + bz = ak_1x + bk_2x = x(ak_1 + bk_2) \Rightarrow x|(ay + bz)$$

■

Proposição 1.3. *Sejam os inteiros x, y, z e w, com $x \neq 0$ e $z \neq 0$. Tem-se que, se $x|y$ e $z|w$, então $(x.z)|(y.w)$.*

Demonstração: Ver em proposição 3.1.2 em [2], página 31.

Proposição 1.4. *Sejam os inteiros x e y, com $y \neq 0$. Tem-se que se $x|y$, então $|x| \leqslant |y|$.*

Demonstração: Ver proposição 3.1.6 em [2], página 32.

■

Nem sempre um inteiro x divide a outro inteiro y, porém com o próximo teorema, que é conhecido como Algoritmo da divisão ou Algoritmo de Euclides, veremos como trabalhar quando temos dois números e que um não é divisível pelo outro.

Teorema 1.1. *Sejam x e y dois inteiros, com $y > 0$. Existem dois únicos números inteiros q e r, tais que*

$$y = x.q + r \quad , com \quad 0 \leqslant r < x$$

Os elementos q e r são denominados, respectivamente, de *quociente* e *resto* da divisão de x por y.

Demonstração:

Ver Teorema 1, páginas 102 e 103 em [1] ou Teorema 3.2.1, páginas 35 e 36 em [2].

■

1.2 Máximo Divisor Comum

Definição 1.3. *Sejam os inteiros x e y. Um número inteiro positivo d diz-se máximo divisor comum de x e y, denotado por $mdc(x, y)$, se verifica:*

i) $d|a$ e $d|b$;

ii) Se existe um inteiro z, de modo que $z|x$ e $z|y$, então $z|d$, donde $z \leqslant d$.

A partir da Definição 1.3, podemos definir quando dois números serão denominados primos entre si.

Definição 1.4. *Dois números inteiros x e y são denominados primos entre si se $mdc(x, y) = 1$.*

Proposição 1.5. *Sejam dois inteiros x e y, temos que se $x = yq + r$ e $d = mdc(x, y)$, então $d = mdc(y, r)$.*

Demonstração:

Por hipótese, temos que $d = mdc(x, y)$, logo $d|x$ e $d|y$.

Como $d|y$, implica que $d|yq$.

Note que $x = yq + r$ é o mesmo que $x - yq = r$

Portanto, como $d|x$ e $d|yq$, pela Proposição 1.1, item (v), temos que $d|(x - yq)$, ou seja, $d|r$.

Proposição 1.6. *Se $d = mdc(x, y)$, então $mdc(ax, ay) = ad$, para todo $a \in \mathbb{N}$.*

Demonstração: Aplicando o Teorema 1.1 a x e y, e fazendo divisões sucessivas até obtermos resto zero, temos que:

$$x = yq_1 + r_1$$

$$y = r_1q_2 + r_2$$

$$\cdots$$

$$r_{n-2} = (r_{n-1})q_n + r_n$$

$$r_{n-1} = (r_n)q_{n+1}$$

Multiplicando-se as equações anteriores por a, membro-a-membro, temos:

$$ax = ayq_1 + ar_1$$

$$ay = ar_1q_2 + ar_2$$

$$\cdots$$

$$ar_{n-2} = (ar_{n-1})q_n + ar_n$$

$$ar_{n-1} = (ar_n)q_{n+1}$$

Então, pela Proposição 1.5, temos que:

$$ad = ar_n = mdc(ar_{n-1}, ar_n) = \cdots = mdc(ay, ar_1) = mdc(ax, ay)$$

■

Proposição 1.7. *Sejam x, y e $z \in \mathbb{Z}$. Se $x|yz$ e $mdc(x, y) = 1$, então $x|z$.*

Demonstração:

Como, por hipótese, temos que $mdc(x, y) = 1$, pela Proposição 1.6, temos que $mdc(xz, yz) = z$. Por outro lado, temos que $x|yz$, logo $x|xz$. Daí, $x|mdc(xz, yz)$, o que implica $x|z$.

■

1.3 Números Primos

A importância dos números primos reside em um resultado central na teoria de números: afirmar que todo número inteiro natural, maior do que 1, pode ser escrito como um produto de fatores primos.

Os números primos possuem significativa parcela de importância nos teoremas envolvendo os erros de verificação dos códigos de barras.

Definição 1.5. *Um número inteiro positivo p, diferente de 0 e 1, ou seja, $p \geqslant 2$ e que é divisível apenas por 1 e por si próprio é chamado número primo.*

No caso em que o número não seja primo, dizemos que ele é composto.

Proposição 1.8. *Seja p primo e x e y inteiros. Se $p \mid xy$ então $p \mid x$ ou $p \mid y$.*

Demonstração:

Vamos supor que $x \neq 0$ e $y \neq 0$. No caso em que $x = 0$ ou $y = 0$, o resultado é imediato, pois $p|0$.

Admitiremos que $p \nmid x$. De fato, dado um inteiro c, tem-se que $c|x$ e $c|p$, então $c = 1$ ou $c = p$, pois p é primo.

Porém, como $p \nmid x$, temos que $c = 1$, pois $c = p$ implica $p \mid x$. A Proposição 1.7 nos garante que $p|y$.

■

1.4 Congruências

Esta seção é de suma importância para o desenvolvimento tema do trabalho, em especial o sistema de identificação de erros, que é baseado na aritmética modular. Daí a finalidade de um estudo das congruências e suas propriedades.

Definição 1.6. *Sejam x, y e m números inteiros, com $m > 0$. Dizemos que x e y são congruentes módulo m, se $m|(x - y)$. Denota-se por $x \equiv y(mod\ m)$.*

No caso em que x não é congruente a y módulo m, ou seja, $m \nmid (x - y)$, escreve-se $x \not\equiv y(mod\ m)$.

1.4.1 Propriedades das Congruências

Proposição 1.9. *Sejam x e y números inteiros, temos que $x \equiv y(mod\ m)$ se, e somente se, os restos das divisões de x e y por m são iguais.*

Demonstração:

$(\Rightarrow)$

Pelo algoritmo da divisão, Teorema 1.1, existem $q_1, q_2, r_1, r_2 \in \mathbb{Z}$, com $0 \leqslant r_1, r_2 < m$, de modo que $x = m.q_1 + r_1$ e $y = m.q_2 + r_2$.

Subtraindo-se x e y temos:

$$x - y = (m.q_1 + r_1) - (m.q_2 + r_2)$$

$$= mq_1 - mq_2 + r_1 - r_2$$

$$m.(q_1 - q_2) - (r_2 - r_1)$$

Então, como $x \equiv y(mod\ m)$ implica $m|(x - y)$. Note que $m|m(q_1 - q_2)$. Consequentemente, pela Proposição 1.2, $m|[(x - y) - m(q_1 - q_2)]$. Daí, teremos $m|(r_2 - r_1)$. Porém, como $|r_1 - r_2| < m$, só será possível se $r_2 - r_1 = 0$, ou seja, se $r_1 = r_2$.

$(\Leftarrow)$

Por hipótese, os restos das divisões de x e y são iguais, logo, pelo Teorema 1.1 podemos escrever:

$x = mq_1 + r$ e $y = mq_2 + r$, donde q_1, q_2 e r são inteiros, com $0 \leqslant r < m$.

Daí, subtraindo-se x e y, temos:

$$x - y = (mq_1 + r) - (mq_2 + r) = mq_1 + r - mq_2 - r = mq_1 - mq_2 = m.(q_1 - q_2)$$

Logo $m|(x - y)$, ou seja, $x \equiv y(mod\ m)$

■

A proposição a seguir contém algumas propriedades das congruências módulo m.

Proposição 1.10. *Sejam x, y, z e m números inteiros, com $m > 0$. Então:*

- *(i)* $x \equiv x(mod\ m)$
- *(ii) Se* $x \equiv y(mod\ m)$ *então* $y \equiv x(mod\ m)$.
- *(iii) Se* $x \equiv y(mod\ m)$ *e* $y \equiv z(mod\ m)$, *então* $x \equiv z(mod\ m)$

Demonstrações:
(i) De fato, como $m|0$, implica $m|(x - x)$.

(ii) Se $x \equiv y(mod\ m)$, então $m|(x - y)$, ou seja, existe um inteiro a tal que $m = (x - y).a$. Daí, podemos escrever $m = -(y - x).a$, pois $-(y - x) = (x - y)$.Logo, temos que: $m|(y - x)$, o que implica $y \equiv x(modm)$.

(iii) Sejam $a, b \in \mathbb{Z}$. Se $x \equiv y(mod\ m)$, então $m|(x - y)$. Logo, $m = (x - y).a$. De modo análogo, podemos escrever $y \equiv z(mod\ m)$. Daí $m|(y - z)$ e, por fim, $m = (y - z).b$
Note que, pela Proposição 1.2, $m|[(x - y) + (y - z)]$, ou seja, $m|(x - z)$. Portanto,
$x \equiv z(mod\ m)$.

■

A noção de congruência torna-se mais útil e valiosa pelo fato de ser uma relação que é compatível com as operações de adição e multiplicação no conjunto $\mathbb{Z}$, conforme as proposições a seguir:

Proposição 1.11. *Sejam os inteiros x, y, z e m, de modo que $m > 1$. Se $x \equiv y(mod\ m)$, então:*

i) $(x + z) \equiv (y + z)\ (mod\ m)$

ii) $(x - z) \equiv (y - z)\ (mod\ m)$

iii) $xz \equiv yz\ (mod\ m)$

Demonstração:
(i) Como $x \equiv y\ (mod\ m)$, pela Definição 1.6, então $m|(x - y)$, o que implica em
$m|(x + z - z - y)$, ou seja, $m|[(x + z) - (y + z)]$. Logo $(x + z) \equiv (y + z)\ (mod\ m)$.

(ii) Já que $x \equiv y \ (mod\ m)$, pela Definição 1.6, temos que $m|(x-y) = (x-z+z-y)$, ou seja, $m|[(x-z)-(y-z)]$. Portanto, $(x-z) \equiv (y-z) \ (mod\ m)$.

(iii) Como $x \equiv y \ (mod\ m)$, pela Definição 1.6, temos que $m|(x-y)$. Pela Proposição 1.2, item (i), temos que se $m|(x-y)$, então $m|(x-y)z$, isso é, $m|(xz-yz)$. Daí, pela definição 1.6, temos que $xz \equiv yz \ (mod\ m)$.

■

Proposição 1.12. *Sejam os inteiros x , y , z , a , b e m , com $m > 1$. Se $x \equiv y \ (mod\ m)$ e $a \equiv b \ (mod\ m)$, então:*

i) $(x+a) \equiv (y+b)(mod\ m)$

ii) $xa \equiv yb \ (mod\ m)$

Demonstração:
(i) Sabemos que $x \equiv y \ (mod\ m)$ e $a \equiv b \ (mod\ m)$, logo $m|(x-y)$ e $m|(a-b)$, logo, pela proposição 1.1 item (v), temos que $m|[(x-y)+(a-b)]$ e assim $m|[(x+a)-(y+b)]$. Portanto, $(x+a) \equiv (y+b)(mod\ m)$.

(ii) Agora, como $x \equiv y \ (mod\ m)$ e $a \equiv b \ (mod\ m)$, $m|(x-y)$ e $m|(a-b)$, assim pela proposição 1.2, temos que $m|[a(x-y)+y(a-b)]$, ou seja,$m|(xa-ya+ya-yb)$. Logo, $m|(xa-yb)$, o que implica $xa \equiv yb \ (mod\ m)$.

■

Teorema 1.2. *Sejam os inteiros a e m. Temos que, $ab \not\equiv 0(mod\ m)$, para todo $b \in \{1, ..., m-1\}$ se, e somente se, $mdc(a,m) = 1$.*

Demonstração:
($\Rightarrow$)
Vamos supor que $mdc(a,m) = d > 1$. Então, $d|a$ e $d|m$. Portanto, existem os inteiros x e y, tais que $a = dx$ e $m = dy$, de modo que $1 \leqslant y < m$, pois se $y = m$, implica que $ab \equiv 0(mod\ m)$,

o que é contrário à hipótese.

Fazendo $b = y$, temos que:

$$ab = ay = dxy = xdy = xm \equiv 0(mod\ m)$$

Ou seja, como $m|xm$, implica que $xm \equiv 0(mod\ m)$. Logo, $ay \equiv 0(mod\ m)$.

Porém, como $1 \leqslant b < m$, pois $ab \ncong 0(mod\ m)$, teremos uma contradição.

Portanto, $d = 1$, isso é, $mdc(a, m) = 1$.

($\Leftarrow$)
Como $mdc(a, m) = 1$, então temos que $m \nmid a$.

Seja $1 \leqslant b < m$, tal que $ab \equiv 0(mod\ m)$. Daí, temos que $m|ab$ e $mdc(a, m) = 1$.

Pela proposição 1.7, temos que $m|b$. Um absurdo, pois $1 \leqslant b < m$. Logo, $ab \ncong 0(mod\ m)$.

■

Como os sistemas de identificação utilizam, em sua maioria, o produto escalar entre dois vetores, aqui neste trabalho usaremos a definição de acordo com [5].

Definição 1.7. *Sejam dois vetores α e β, com $\alpha = (a_1, a_2, a_3)$ e $\beta = (b_1, b_2, b_3)$. O produto escalar entre os vetores α e β, denotado por $\alpha \cdot \beta$, define-se por:*

$$\alpha \cdot \beta = a_1b_1 + a_2b_2 + a_3b_3$$

Em suma, os conteúdos aqui vistos serão de grande importância para melhor entendimento do tema do trabalho, que são os códigos de barras e detecção de erros de digitação.

CAPÍTULO 2

OS CÓDIGOS DE BARRAS

Neste capítulo veremos como reconhecer um código de barras, bem como ele é constituído, os tipos de códigos de barras existentes e os mais utilizados. Destina-se, também, o modo como os códigos de barras são lidos e como são codificados os códigos EAN-13 e UPC-A. As referências utilizadas neste capítulo estão de acordo com [6] e [7], basicamente, bem como de [8] a [13].

2.1 O que são códigos de barras?

Segundo [7], os códigos de barras são composições de linhas paralelas e verticais, pretas e brancas e de espessuras e alturas diferentes. Essas linhas, que são as barras, contêm dados do produto. Os códigos de barras são representados por números apenas ou por números e letras, a depender do tipo do código de barras adotado.

Existem dois tipos de códigos de barras: o *código bidimensional* e os *códigos lineares*. A principal diferença entre o código bidimensional (2D) do código de barras linear (1D) é que os lineares guardam em seus códigos apenas números e referências, pequenas palavras, enquanto o código 2D tem capacidade de armazenar diversos dados. As diferenças, com relação à capacidade de armazenação de dados, entre os códigos 1D e 2D podem, também, serem vistas na Figura 2.1 (disponível em [11]):

→ **Capacidade**: 20 dígitos apenas

QR Code – Capacidade máxima de dados	
Numérico	7,089 caracteres
Alfanumérico	4,296 caracteres
Binário (8 bits)	2,953 bytes
Kanji, full-width Kana	1,817 caracteres

Figura 2.1: Códigos 1D e 2D

A Figura 2.2, disponível em [10], contém os principais tipos de códigos de barras bidimensionais:

Figura 2.2: Códigos Bidimensionais

Na Figura 2.3, que encontra-se em [8], temos os códigos de barras lineares mais utilizados:

Figura 2.3: Códigos Lineares

2.2 Composição de um Código de Barras Linear

O código de barras é composto por várias barras paralelas e verticais, contendo uma sequência de números, de modo a permitir a leitura pela máquina e pelo homem. A Figura 2.4, disponível em [12], contém um exemplo de um código de barras linear:

Figura 2.4: Exemplo de Códigos Lineares

2.2.1 As Partes do Código de Barras

Na Figura 2.5, temos as partes de um código, de acordo com as **barras**:

Os *separadores* servem, basicamente, para indicar a extremidade do código. Notemos, pela Figura 2.5, que existem três separadores, e que o separador "central" delimita o código em duas partes: lado direito e lado esquerdo. Daí, a principal função dos separadores é permitir que a máquina leitora reconheça de qual lado o código está sendo lido.

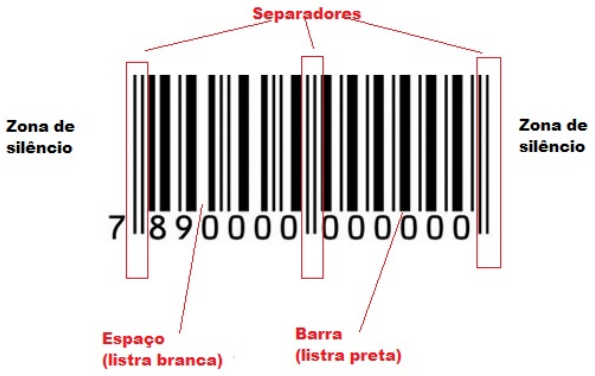

Figura 2.5: Composição do Código de Barras 1

As *zonas de silêncio* são as margens ou os espaços antes do dígito inicial e final. Estas não podem ser excluídas pois a interpretação do código será impossibilitada.

Na Figura 2.6, temos as partes de um código, de acordo com os **números** que aparecem abaixo das barras:

Figura 2.6: Composição do Código de Barras 2

Então:

- Os três primeiros dígitos do código de barras representam o país onde o produto foi fabricado. Cada país tem a sua numeração correspondente. No caso do Brasil, esses três números são 789.

- Os quatro dígitos seguintes, determinam a empresa que fabricou o produto.
- Os cinco dígitos, seguintes aos que caracterizam a empresa, são os que caracterizam o produto.
- O último dígito tem uma função diferenciada, pois permite verificar possíveis erros de digitação, denominado *dígito verificador*.

As posições acima descritas, são determinadas segundo [9], que organiza e padroniza os códigos.

2.3 O Código UPC-A

Segundo [6], o código UPC foi aceito formalmente em maio de 1973. Inicialmente chamava-se de UPC (Universal Product Code), fora adotado nos Estados Unidos e Canadá e continha apenas 12 dígitos, conforme o exemplo da Figura 2.7, disponível em [13]:

Figura 2.7: Código UPC-A

2.4 O Código EAN-13

Baseado no UPC-A, o código EAN-13 (European Article Numbering) foi elaborado a partir da necessidade de se adicionar um dígito a cada código, de modo a permitir a identificação do país de origem do produto, de forma que a mesma máquina leitora pudesse ler códigos UPC e EAN.

Na Figura 2.8, temos um exemplo de um código de barras EAN-13:

Figura 2.8: Código EAN-13

2.5 Como é feita a leitura do Código de Barras

O código de barras é lido através de equipamento apropriado chamado *scanner*, ver Figura 2.9, disponível em <http://www.megsystem.com.sg/BarCodeScanner-1000CCD.html>

Figura 2.9: Exemplo de Scanner

2.6 Codificação do Código de Barras UPC-A

Inicialmente, vamos analisar a estrutura dos códigos UPC-A pela Figura 2.10:

Figura 2.10: Exemplo de código UPC-A

Segundo [6] e [7], as barras representam, cada uma, uma sequência de sete dígitos, que alternam entre 0 e 1, que é a linguagem utilizada pelo computador, que são associados de acordo com a cor e espessura da barra.

Existem quatro tipo de espessuras para as duas cores das listras (pretas e brancas), que são: fina, média, grossa e muito grossa. Portanto, para cada cor e espessura da barra é associado um dígito, conforme a **Tabela 1** a seguir:

Espessura	Barra Branca	Barra Preta
Fina	0	1
Média	00	11
Grossa	000	111
Muito Grossa	0000	1111

A Figura 2.11, disponível em [7], contém uma situação que auxilia numa melhor compreensão dos fatos citados acima:

Figura 2.11: Codificação dos códigos

No exemplo dado no início dessa seção, iniciando após as barras limites (separadores), as quatro primeiras listras: uma **listra branca grossa**, uma **preta média**, uma **branca fina** e uma **preta fina** são representadas, respectivamente, pela sequência **0001101**. Esta sequência representa o número 0, o primeiro do código da figura.

Procedendo de modo análogo, verificamos que o dígito seguinte, o **1**, é representado pela sequência de listras: **branca média**, **preta média**, **branca média** e **preta fina**, o que corresponde à sequência: **0011001**.

O scanner consegue distinguir a direita da esquerda. Isto ocorre porque os dígitos são codificados de maneira diferente quando estão do lado direito ou do lado esquerdo do código de barras. Isso é feito conforme à seguinte **Tabela 2**, adaptada de [6]:

Dígito	do lado esquerdo	do 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

A codificação de um certo número à direita é obtida de sua codificação à esquerda, quando trocamos cada 0 por 1 e vice versa.

O processo de identificação fica mais claro quando entendemos que, cada sequência do lado esquerdo tem um número ímpar de dígitos iguais a 1 e, logo, cada uma das que estão à direita tem um número par.

Daí, verificando a paridade de cada sequência de sete dígitos, a máquina identifica imediatamente de que lado está lendo o código.

2.7 Codificação do Código de Barras EAN-13

A codificação dos Códigos de Barras EAN-13 é análoga à codificação dos Códigos de Barras UPC-A.

Na elaboração do código EAN fora encontrada uma dificuldade: a necessidade de adicionar um dígito a cada código de tal forma que a mesma leitora ótica pudesse ler, sem distinção, os códigos EAN e UPC. Daí, a solução encontrada foi: Os países que utilizam o código UPC antigo, que são EUA e Canadá, são identificados com um 0 na frente, e o resto da codificação é feita utilizando-se o sistema anterior.

Para os demais países, os primeiros dois ou três dígitos identificam o país. No caso do Brasil, os códigos de barras de produtos produzidos começam com a sequência 789. Como era necessário adicionar um dígito e, também, manter o mesmo padrão de tamanho do código de barras, para não ter que modificar todas as leitoras, a ideia utilizada foi fazer com que o novo dígito estivesse implícito na forma de escrita de todos os outros.

Logo, não foi modificada a codificação do lado direito, permitindo assim que todas as leitoras continuassem a identificar o lado correspondente, porém, a codificação do lado esquerdo varia de acordo com o dígito inicial. Um dígito do lado esquerdo pode ser agora codificado com um número par ou número ímpar de dígitos iguais a 1, de acordo a **Tabela 3**, disponível em [6]:

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

Por fim, para cada dígito inicial, escolhe-se uma alternância diferente de pares e ímpares, de acordo com o seguinte critério contido na **Tabela 4**, segundo [6]:

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

Veremos um exemplo, uma pipoca doce produzida no Brasil, que é identificada pelo código **7898927019426**. Como começa com **789**, o primeiro dígito, que estará implícito na codificação dos demais, é o **7**.

Por consequência, deve-se usar do lado esquerdo, a seguinte ordem de codificação (obtida na Tabela 4):

(8) ímpar - **(9)** par - **(8)** ímpar - **(9)** ímpar - **(2)** ímpar - **(7)** par

Consultando a Tabela 3, temos:

$$8 \rightarrow 0110111$$
$$9 \rightarrow 0010111$$
$$8 \rightarrow 0110111$$
$$9 \rightarrow 0010111$$
$$2 \rightarrow 0010011$$
$$7 \rightarrow 0010001$$

Para os dígitos do lado direito não é preciso nos preocuparmos com a paridade, e diretamente da Tabela 2, obtemos a seguinte codificação:

$$0 \rightarrow 1110010$$

$$1 \rightarrow 1100110$$
$$9 \rightarrow 1110100$$
$$4 \rightarrow 1011100$$
$$2 \rightarrow 1101100$$
$$6 \rightarrow 1010000$$

Portanto, o código de barras correspondente é:

Figura 2.12: Codificação do EAN-13

Sumariando, o código de barras nada mais é do que um número identificador do produto. Apenas um número escrito em uma linguagem diferente: uma linguagem que se comunica com barras pretas e brancas.

CAPÍTULO 3

DETECÇÃO DE ERROS

Os códigos de barras podem e devem ser controlados para tornar possível a detecção de erros de codificação. Isso significa que quando, por exemplo, um caixa do supermercado digitar um número errado do código que identifica o produto, esse erro de codificação deve ser "avisado". Para isso, criou-se um grupo de algarismos, denominados *algarismo de testes*, *algarismo de controle* ou *dígito de verificação*, e que representam o último dígito do código. O conteúdo aqui contido, está de acordo com [6], [7] e [14].

3.1 Sistemas de Identificação Modular (SIM)

Para identificarmos, de maneira rápida, um produto, livro ou pessoa, utilizamos um número. Esses números são chamados de números de identificação, tais como o Registro de Identidade (RG), o Cadastro de Pessoas Físicas (CPF), identificação de livros, cuja sigla é ISBN(International Standard Book Number)), códigos de barras, contas bancárias e outras situações. Em geral, esses números de identificação contêm apenas algarismos (são os códigos numéricos) ou algarismos e letras (são os códigos alfanuméricos).

De modo a evitar e detectar fraudes ou que um código seja transmitido de forma errada, os sistemas de identificação utilizam o dígito verificador para tal. Geralmente, o dígito verificador é o último dígito do código. Como seu valor é calculado utilizando Aritmética Modular, daí entitulam-se esses sistemas como Sistemas de Identificação Modular.

Em um sistema que utiliza-se a aritmética modular, um número de identificação é escrito da forma:

$$a_1a_2a_3...a_nA,$$

onde A é o dígito verificador, a_i o dígito do número de identificação, com $a_i \in \{0,1,2,...,9\}$, $i \in \{1,2,...,n\}$ e $A \in \{1,2,...,9\}$

Tomando-se p_i com $p_i \in \{0,1,2,...,k-1\}$ e $i \in \{1,2,...,n\}$, temos que o valor de A é calculado através da seguinte congruência:

$$p_1a_1 + p_2a_2 + ... + p_na_n + A \equiv 0 \ (mod\ k)$$

onde os elementos $p_1, p_2, ..., p_n$ são os pesos, que são escolhidos inicialmente.

Esse sistema é denominado de *sistema módulo k* e a soma $p_1a_1 + p_2a_2 + ... + p_na_n + A$ de *soma teste*, que representaremos por S, isso é,

$$S = p_1a_1 + p_2a_2 + ... + p_na_n + A$$

No exemplo a seguir, teremos um melhor entendimento sobre o funcionamento do sistema de identificação modular.

Exemplo 3.1. *Uma empresa utiliza três dígitos, $a_1a_2a_3$, para identificar cada produto que é vendido. Para certificar-se que estes números serão transmitidos de maneira correta, acrescenta-se um quarto dígito em cada número, gerando assim o código de identificação $a_1a_2a_3A$. A empresa utiliza um sistema módulo 10 com pesos 3,1,3. Portanto, para calcularmos o dígito verificador devemos resolver a equação $3a_1 + a_2 + 3a_3 + A \equiv 0(mod 10)$. Então, um produto identificado pelo número 123, temos que A = 6 , pois A deve satisfazer à congruência:*

$$3.1 + 1.2 + 3.3 + A \equiv 0(mod\ 10)$$

$$3 + 2 + 9 + A \equiv 0(mod\ 10)$$

$$14 + A \equiv 0(mod\ 10)$$

Logo, $A = 6$ pois $14 + 6 = 20$ e $20 \equiv 0(mod\ 10)$. Daí, o código de identificação desse produto é 1236. Porém, o código 2468 é um código inválido, pois $3.2 + 1.4 + 3.6 + 8 = 6 + 4 + 18 + 8 = 36$ e $36 \not\equiv 0(mod\ 10)$.

3.2 O Dígito Verificador

O dígito verificador não é um número aleatório, tampouco faz parte da sequência dos dígitos que contém informações sobre o produto. Ele é obtido através de um algoritmo, que obtém-se a partir dos dígitos anteriores, seguindo os padrões do sistema adotado, com a finalidade de certificar a validade de um código numérico.

3.2.1 O Dígito Verificador do Código EAN-13

Todo código de barras possui um dígito de verificação, que funciona como um detector de erros. Esse dígito normalmente é o último algarismo da sequência. Para descobrir qual é esse dígito, utilizaremos o algoritmo a seguir:

Algoritmo:

1) Suponhamos que um determinado produto está identificado, no sistema EAN-13, por uma dada sequência de dígitos $a_1a_2a_3...a_{12}a_{13}$.

   Como os primeiros dígitos identificam o país de origem, o fabricante e o produto específico, os primeiros doze dígitos da sequência estão determinados naturalmente por um método padrão, a cargo de uma autoridade classificadora em cada país. Assim sendo, chamaremos o décimo terceiro dígito de verificação, por x.

2) Para facilitar o entendimento, escrevemos esta sequência como um vetor:

$$\alpha = (a_1, a_2, a_3, ..., a_{12}, x)$$

   Os códigos de formato EAN-13 utilizam-se de um vetor fixo, denominado *vetor de pesos*, identificado por:

$$\beta = (1, 3, 1, 3, 1, 3, 1, 3, 1, 3, 1, 3, 1).$$

3) Calcula-se, então, o produto escalar entre esses dois vetores:

$$\alpha \cdot \beta = (a_1, a_2, a_3, ..., a_{12}x) \cdot (1, 3, 1, 3, 1, 3, 1, 3, 1, 3, 1, 3, 1)$$

$$\alpha \cdot \beta = a_1 + 3a_2 + a_3 + 3a_4 + a_5 + 3a_6 + a_7 + 3a_8 + +a_9 + 3a_{10} + a_{11} + 3a_{12} + x$$

.

4) Agora, o dígito verificador x é determinado de tal forma que a soma obtida é um múltiplo de 10 (pois trata-se de uma operação feita no sistema decimal de base 10), isto é,

$$\alpha \cdot \beta \equiv 0(mod\ 10)$$

Ou também,

$$a_1 + 3a_2 + a_3 + 3a_4 + a_5 + 3a_6 + a_7 + 3a_8 + +a_9 + 3a_{10} + a_{11} + 3a_{12} + x \equiv 0(mod 10)$$

Por exemplo, no caso do código da Figura 3.1, os números que indicam o país de origem, o fabricante e o produto são **789**, **6422** e **51322**, respectivamente e como pode se notar, o dígito verificador é **7**.

Figura 3.1: Cálculo do Dígito Verificador no Código EAN-13

Através da aplicação do algoritmo, citado anteriormente, faz-se a confirmação desse fato.

Sendo $\alpha = (7, 8, 9, 6, 4, 2, 2, 5, 1, 3, 2, 2, x)$ e $\beta = (1, 3, 1, 3, 1, 3, 1, 3, 1, 3, 1, 3, 1)$, então:

$$\alpha \cdot \beta = (7.1) + (8.3) + (9.1) + (6.3) + (4.1) + (2.3) + (2.1) + (5.3) + (1.1) + (3.3) + (2.1) + (2.3) + (x.1)$$

$$\alpha \cdot \beta = 7 + 24 + 9 + 18 + 4 + 6 + 2 + 15 + 1 + 9 + 2 + 6 + x$$

$$\alpha \cdot \beta = 103 + x$$

Como $103 + x \equiv 0(mod\ 10)$, logo $x = 7$. O que mostra que a identificação do produto está correta, pois 7 é o menor inteiro que torna esta soma um múltiplo de 10.

Agora, se ao digitar esse código de barras, houvesse um erro de digitação humano no quarto dígito e o código fosse assim transmitido: 789**1**422513227. O computador, programado para fazer a leitura utilizando o algoritmo, detectaria o erro, pois:

$$\alpha \cdot \beta = (7 \cdot 1) + (8 \cdot 3) + (9 \cdot 1) + (1 \cdot 3) + (4 \cdot 1) + (2 \cdot 3) + (2 \cdot 1) + (5 \cdot 3) + (1 \cdot 1) + (3 \cdot 3) + (2 \cdot 1) + (2 \cdot 3) +$$

$$\alpha \cdot \beta = 7 + 24 + 9 + 3 + 4 + 6 + 2 + 15 + 1 + 9 + 2 + 6 + 7$$

$$\alpha \cdot \beta = 95$$

e

$$95 \ncong 0(mod\ 10)$$

Logo, a leitura desse código pelo computador não seria feita, pois, como visto, fora cometido um erro, não sendo identificado pelo computador programado para aplicar o algoritmo de verificação de códigos de barras EAN-13.

3.2.2 O Dígito Verificador do Código UPC-A

Com o código UPC-A é muito semelhante. Como utiliza apenas 12 dígitos (pois usa apenas um para identificar o país de origem do artigo), o *vetor de pesos* utilizado pelo UPC apresenta um dígito a menos:

$$\beta = (3,1,3,1,3,1,3,1,3,1,3,1)$$

Por exemplo, vamos utilizar o código UPC-A da Figura 3.2 (disponível em: <http://www.manutencaoesuprimentos.com.br/conteudo/5585-codigo-de-barras-universal/>)

Figura 3.2: Dígito Verificador do Código UPC-A

Como $\alpha = (0,2,1,2,0,0,9,7,8,1,5,9)$ e $\beta = (3,1,3,1,3,1,3,1,3,1,3,1)$, então:

$$\alpha \cdot \beta = (0\cdot 3) + (2\cdot 1) + (1\cdot 3) + (2\cdot 1) + (0\cdot 3) + (0\cdot 1) + (9\cdot 3) + (7\cdot 1) + (8\cdot 3) + (1\cdot 1) + (5\cdot 3) + (9\cdot 1)$$

$$\alpha \cdot \beta = 0 + 2 + 3 + 2 + 0 + 0 + 27 + 7 + 24 + 1 + 15 + 9$$

$$\alpha \cdot \beta = 90$$

Como $90 \equiv 0 (mod\ 10)$, tem-se que a identificação do produto está correta.

Porém, se houvesse um erro de digitação humana, uma "permutação de posições"entre os 2^o e 3^o dígitos, e o código fosse assim transmitido: 0**12**200978159, o computador, através do algoritmo que fora programado para fazer a leitura dos códigos, acusaria que o código foi digitado errado, pois:

$$\alpha \cdot \beta = (0\cdot 3) + (1\cdot 1) + (2\cdot 3) + (2\cdot 1) + (0\cdot 3) + (0\cdot 1) + (9\cdot 3) + (7\cdot 1) + (8\cdot 3) + (1\cdot 1) + (5\cdot 3) + (9\cdot 1)$$

$$\alpha \cdot \beta = 0 + 1 + 6 + 2 + 0 + 0 + 27 + 7 + 24 + 1 + 15 + 9$$

$$\alpha \cdot \beta = 92$$

e

$$92 \ncong 0 (mod\ 10)$$

3.3 Detectando Erros

Segundo [6], existem diversos tipos de erros que podem ser cometidos ao digitar um vetor de identificação, ao qual denominamos de α na seção anterior, tais como a embalagem do produto estar úmida ou amassada.

Os erros num único dígito e as transposições são os mais frequentes. O *erro único* ou *singular* é o mais comum, pois representa mais de 79% dos erros cometidos. Esse erro acontece quando, por exemplo, tenta-se digitar um código 2**3**56 e, ao invés disso, digita-se 2**4**56.

Já os erros de *transposição adjacente*, que correspondem a pouco mais de 10% do total de erros cometidos ao digitar acontecem quando, por exemplo, ao invés de digitar o código 2**46**8, digita-se 2**64**8.

Com uma porcentagem de quase 1%, estão os erros de **transposição alternada**, ou seja, acontecem quando ao tentar digitar, por exemplo o código 1**3**57**9**, é digitado 1**7**5**3**9.

Ainda segundo esses estudos, a chance de ocorrer mais de um erro ao digitar um número é muito pequena.

Na **Tabela 5** abaixo, abreviando a tabela publicada em [6], estão os tipos de erros e as frequências relativas de cada um:

TIPO DE ERRO		Frequência relativa (%)
erro único	...a...↦...b...	79,1
erro de transposição adjacente	...ab...↦...ba...	10,2
erro de transposição alternada	...abc...↦...cba...	0,8
outros erros		9,9

Por meio de alguns exemplos, obteremos uma melhor compreensão desses erros.

Exemplo 3.2. *Considere o código de barras 7898532698238. Suponha que houve um erro de digitação na posição a_1, e que o código foi assim transmitido: 8898532698238. Sendo $\alpha = (8,8,9,8,5,3,2,6,9,8,2,3,8)$ o vetor que representa esse código e $\beta = (1,3,1,3,1,3,1,3,1,3,1,3,1)$ o vetor de pesos, temos que o computador detectaria o erro, pois:*

$$\alpha \cdot \beta = 8 + 24 + 9 + 24 + 5 + 9 + 2 + 18 + 9 + 24 + 2 + 9 + 8$$

$$\alpha \cdot \beta = 151 \ncong 0 (mod\ 10)$$

Porém, se mais de um erro for cometido na digitação, o erro provavelmente ainda será detectado, mas já não se pode ter certeza, pois eles poderiam se "compensar mutuamente"e a soma poderia ainda continuar sendo um múltiplo de 10.

Supondo que além do erro cometido no dígito da posição a_1, ocorresse outro erro com o dígito da posição a_6 , e o código fosse assim transmitido: **8**8985**9**2698238. Da mesma forma o computador teria detectado o erro, pois:

$$\alpha \cdot \beta = 8 + 24 + 9 + 24 + 5 + 27 + 2 + 18 + 9 + 24 + 2 + 9 + 8$$

$$\alpha \cdot \beta = 169 \ncong 0 (mod\ 10)$$

Agora, vamos supor que os erros foram cometidos com os dígitos da posições a_7 e a_{13}, e o código foi digitado da seguinte forma: 7898534698236. Pelo algoritmo do dígito verificador, teríamos:

$$\alpha \cdot \beta = 7 + 24 + 9 + 24 + 5 + 9 + 4 + 18 + 9 + 24 + 2 + 9 + 6$$

$$\alpha \cdot \beta = 150 \equiv 0(mod\ 10)$$

Portanto, o erro não seria detectado pelo computador.

Exemplo 3.3. *Consideremos o código de barras 9788531404580. Ao digitar o código 9788531404580 ocorreu um erro de transposição adjacente entre os elementos a_5 e a_6, e que o código digitado foi 9788**35**1404580.*

Sendo $\alpha = (9,7,8,8,5,3,1,4,0,4,5,8,0)$ o vetor que representa esse código e $\beta = (1,3,1,3,1,3,1,3,1,3,1,3$ o vetor de pesos, temos que o computador teria detectado o erro, pois:

$$\alpha \cdot \beta = 9 + 21 + 8 + 24 + 3 + 15 + 1 + 12 + 12 + 5 + 24$$

$$\alpha \cdot \beta = 134 \ncong 0(mod\ 10)$$

Exemplo 3.4. *Consideremos o código de barras 9781402002380.*

Vamos supor que, ao digitar o código 9781402002380, o erro cometido foi entre os elementos a_{11} e a_{12}, e que o número de fato digitado foi 978140200283**0**. Ao efetuar a verificação, teríamos:

$$\alpha \cdot \beta = (9,7,8,1,4,0,2,0,0,2,8,3,0) \cdot (1,3,1,3,1,3,1,3,1,3,1,3,1)$$

$$\alpha \cdot \beta = (9\cdot 1) + (7\cdot 3) + (8\cdot 1) + (1\cdot 3) + (4\cdot 1) + (0\cdot 3) + (2\cdot 1) + (0\cdot 3) + (0\cdot 1) + (2\cdot 3) + (8\cdot 1) + (3\cdot 3) +$$

$$\alpha \cdot \beta = 9 + 21 + 8 + 3 + 4 + 0 + 2 + 0 + 0 + 6 + 8 + 9 + 0$$

$$\alpha \cdot \beta = 70 \equiv 0(mod\ 10)$$

E como 70 é um múltiplo de 10, o sistema não teria detectado o erro.

Pelos exemplos vistos anteriormente, percebemos que o algoritmo utilizado não detecta 100% dos erros de transposições e possui algumas falhas de segurança. Pela proposição a seguir, é possível demonstrar que a transposição de dois dígitos consecutivos a_i e a_{i+1} não é detectada se, e somente se, $5|(|a_i - a_{i+1}|)$.

Proposição 3.1. *A transposição de dois dígitos consecutivos a_i e a_{i+1} não é detectada, neste sistema de codificação, se e somente se,* $5|(|a_i - a_{i+1}|)$.

Demonstração

Vamos supor que o código

$$\alpha = (a_1, a_2, ..., a_i, a_{i+1}, ..., a_{12}, x)$$

foi digitado como

$$\alpha' = (a_1, a_2, ..., a_{i+1}, a_i, ..., a_{12}, x)$$

e que não tenha sido detectado o erro. Assim, teríamos duas possibilidades:

- se i for ímpar, implica que devemos multiplicar a_i por 1
- se i for par, implica que devemos multiplicar a_i por 3

Seja β o vetor de pesos adotado, ou seja $\beta = (1, 3, 1, 3, 1, 3, 1, 3, 1, 3, 1, 3, 1)$. Fazendo o produto entre os vetores $\alpha \cdot \beta$ e $\alpha' \cdot \beta$, temos que se $\alpha \cdot \beta \equiv 0(mod\ 10)$ e $\alpha' \cdot \beta \equiv 0(mod\ 10)$, então:

$$a_1 + 3a_2 + a_3 + 3a_4 + ... + a_i + 3a_{i+1} + ... + 3a_{12} + x \equiv 0(mod\ 10)$$

e

$$a_1 + 3a_2 + a_3 + 3a_4 + ... + a_{i+1} + 3a_i + ... + 3a_{12} + x \equiv 0(mod\ 10)$$

Fazendo $\alpha \cdot \beta - \alpha' \cdot \beta$, temos:

$$\alpha \cdot \beta - \alpha' \cdot \beta = a_i + 3a_{i+1} - a_{i+1} - 3a_i \equiv 0(mod\ 10)$$

$$\alpha \cdot \beta - \alpha' \cdot \beta = 2a_{i+1} - 2a_i \equiv 0(mod\ 10)$$

$$\alpha \cdot \beta - \alpha' \cdot \beta = 2(a_{i+1} - a_i) \equiv 0(mod\ 10)$$

Portanto,

$$10|2(a_{i+1} - a_i)$$

O que implica

$$5|(|a_i - a_{i+1}|)$$

3.4 Generalização dos Detectores de Erros

Nesta seção estudaremos outros detectores de erros, mas para tal, adotaremos a seguinte linguagem:

- Chamaremos de D será o conjunto que contém os valores que os dígitos utilizados no sistema de identificação podem assumir na codificação.
  Temos que $D = \{x \in \mathbb{Z} \,|0 \leqslant x \leqslant k-1\}$, donde k é um número inteiro positivo e $k > 1$.
- $\alpha = (a_1, a_2, a_3, ..., a_{n-1})$, donde n é um número inteiro positivo, será o **vetor de informação**. No caso em que $\alpha = (a_1, a_2, a_3, ..., a_{n-1}, a_n)$, ou seja, acrescido do dígito verificador a_n, será o **vetor de identificação**.

Definição 3.1. *Consideremos $\beta = (p_1, ..., p_n)$ com $p_i \in D$, $1 \leqslant i \leqslant n$ um vetor de pesos e $c \in D$ um número inteiro fixado. Dados dois inteiros positivos k e n e um conjunto de números $a_1, a_2, .., a_{n-1}$ de modo que $a_i \in D$, $1 \leqslant i \leqslant n$, define-se o número de verificação a_n como o único elemento de D que verifica a equação:*

$$a_1p_1 + a_2p_2 + ... + a_ip_i + ... + a_np_n \equiv c(mod\ k)$$

ou, simplesmente:

$$\sum_{i=1}^{n} a_ip_i \equiv c(mod\ k)$$

Daí, um sistema de codificação de acordo com a Definição 3.1 será denotado por:

$$M = (D, k, n, c, \beta)$$

Donde $D = \{0, 1, ..., k-1\}$.

Como exemplo, temos o sistema utilizado por bancos (alguns) para codificar o número da conta de seus clientes que é composto por 9 dígitos, sendo que o último é o dígito verificador.

Consideremos um banco com o sistema de codificação $M = (D, 10, 9, 0, \beta)$ onde D é o conjunto dos dígitos de 0 a 9 e $\beta = (7, 3, 9, 7, 3, 9, 7, 3, 9)$ é o vetor de pesos previamente definido pelo banco. Se tomarmos o número de certa conta bancária como $67 - 127345 - 7$, usando o algoritmo para verificação temos que:

$$(6, 7, 1, 2, 7, 3, 4, 5, 7).(7, 3, 9, 7, 3, 9, 7, 3, 9) = 6.7 + 7.3 + 1.9 + 2.7 + 7.3 + 3.9 + 4.7 + 5.3 + 7.9$$

$$= 42 + 21 + 9 + 14 + 21 + 27 + 28 + 15 + 63 = 240 \equiv 0(mod\ 10)$$

■

O teorema a seguir descreve a capacidade que tem um sistema definido da forma $M = (D, k, n, c, \beta)$, para detectar os erros mais comuns.

Teorema 3.1. *(Capacidade de detecção de erros) Consideremos k um inteiro positivo e $\beta = (p_1, ..., p_n)$ um vetor de pesos. Suponhamos que um vetor de identificação $\alpha = (a_1, ..., a_n)$ (onde tem-se que $a_i \in \{0, 1, ..., k\}$, para todo índice $i \in \{1, 2, ..., n\}$ satisfaz a condição:*

$$\alpha \cdot \beta = a_1 p_1 + ... + a_n.p_n \equiv c(mod\ k)$$

Então,

- *(1) Todo erro consistente numa única alteração (erro singular) na posição i-ésima será detectado se, e somente se, $mdc(p_i, k) = 1$.*
- *(2) Todo erro de transposição da forma $...a_i...a_j... \longrightarrow ...a_j...a_i...$ será detectado se, e somente se, $mdc(p_i - p_j, k) = 1$, (comi $\neq$ j)*

Demonstração:
(1)
Vamos supor que a_i , na posição i, foi trocado por b_i. Denotaremos α' como vetor resultante desse erro e $\beta = \{p_1, ..., p_n\}$ o vetor de pesos. Daí, o erro não será detectado se, e somente, se

$$\alpha \cdot \beta \equiv c\ (mod\ k)$$

e

$$\alpha' \cdot \beta \equiv c \ (mod\ k)$$

Logo,

$$\alpha \cdot \beta - \alpha' \cdot \beta \equiv 0 \ (mod\ k)$$

Mas

$$\alpha \cdot \beta - \alpha' \cdot \beta = (ai - b_i)p_i \equiv 0 \ (mod\ k)$$

De modo que o erro será detectado se, e somente,

$$k|(ai - b_i)p_i$$

($\Leftarrow$) Vamos supor que $mdc(p_i, k) = 1$ e que o erro não foi detectado. Daí, $k|(ai - b_i)p_i$.

Como, por hipótese, p_i e k são primos, temos que $k|(ai - b_i)$.

Mas, $0 \leqslant a_i, b_i < k$, ou seja, $ai - b_i = 0 \Rightarrow ai = b_i$. O que é um absurdo, pois, por hipótese, $ai \neq b_i$

($\Rightarrow$) Vamos supor que todo erro é detectado e que $mdc(p_i, k) = d \neq 1$.

Sejam $b_i = a_i + \frac{k}{d}$ ou $b_i = a_i - \frac{k}{d}$.

Afirmação: $0 \leqslant b_i < k$

De fato, se $k \leqslant b_i$ e $b_i < 0 \Rightarrow k \leqslant a_i \pm \frac{k}{d}$. Daí:

$$k \leqslant a_i + \frac{k}{d} < 0 \Rightarrow k - \frac{k}{d} < \frac{k}{d} \Rightarrow k < \frac{k}{d} + \frac{k}{d} \Rightarrow k < \frac{2k}{d} \Rightarrow dk < 2k \Rightarrow d < 2$$

O que é um absurdo, pois d é um número inteiro e $d \neq 1$.

Então, $a_i - b_i = (\pm\frac{k}{d})p_i = k \cdot (\pm\frac{p_i}{d})$, isto é, o erro que substitui a_i por b_i não é detectado, o que é um absurdo!

(2)

Sejam $\alpha = (a_0, ..., a_i, ..., a_j, ..., a_n)$ e $\alpha' = (a_0, ..., a_j, ..., a_i, ..., a_n)$ os vetores resultantes da troca e $\beta = \{p_1, ..., p_n\}$ o vetor de pesos. Neste caso, o erro não será detectado se, e somente se,

$$\alpha \cdot \beta \equiv \alpha' \cdot \beta (mod\ k)$$

isto é

$$k|(a_i - a_j)(p_i - p_j)$$

($\Leftarrow$) Vamos supor que $mdc(p_i - p_j, k) = 1$ e que o erro não foi detectado.

Então, temos que $k|(a_i - a_j)(p_i - p_j)$.

Como, por hipótese, $p_i - p_j$ e k são primos, segue que $k|(a_i - a_j)$.

Mas, $0 \leqslant a_i, a_j < m \Rightarrow a_i - a_j \Rightarrow a_i = a_j$, o que é um absurdo!!!

($\Rightarrow$) Vamos supor que todo erro é detectado e $mdc(p_i - p_j, k) = d \neq 1$.

Usando as ideias anteriores, temos que:

$$a_j = a_i \pm \frac{k}{d}$$

Daí, o erro de transposição cometido trocando-se a_i por a_j não é detectado, chegando assim a um absurdo.

■

Como consequência do resultado encontrado, temos que a melhor forma de ter-se total certeza que o sistema de codificação será capaz de detectar os erros únicos e todos os erros de transposições é tomarmos k um número primo.

Um exemplo dessa consequência é o sistema universal adotado para a classificação de livros **ISBN** (International Standard Book Number). O módulo adotado neste sistema é 11, mas afim de simplificar a notação, utiliza-se como conjunto de valores D os dígitos de 0 a 9. Os vetores de identificação têm 10 componentes. $\beta = (10, 9, 8, 7, 6, 5, 4, 3, 2, 1)$ será o vetor de pesos. Assim, na notação em que estamos utilizando, este sistema será $(D, 11, 10, 0, \beta)$.

Por exemplo, vejamos o código da Figura 3.3:

Figura 3.3: Código ISBN

O livro possui o número ISBN 85-904198-1-9. Sendo $\alpha = (8, 5, 9, 0, 4, 1, 9, 8, 1, 9)$ o vetor que representa o código desse livro e $\beta = (10, 9, 8, 7, 6, 5, 4, 3, 2, 1)$ o vetor de peso do ISBN, temos que o dígito final de verificação é 9 porque:

$$\alpha \cdot \beta = (8, 5, 9, 0, 4, 1, 9, 8, 1, 9) \cdot (10, 9, 8, 7, 6, 5, 4, 3, 2, 1)$$

$$= 8.10 + 5.9 + 9.8 + 0.7 + 4.6 + 1.5 + 9.4 + 8.3 + 1.2 + 9.1$$

$$= 80 + 45 + 72 + 0 + 24 + 5 + 36 + 24 + 2 + 9$$

$$= 297 \equiv 0(mod\ 11)$$

Porém, ao determinarmos o dígito verificador de um livro cujo código é 0387960535, nos depararemos com um pequeno problema. Para isso, fazendo $\alpha = (0, 3, 8, 7, 9, 6, 0, 3, 5, f)$ e

$\beta = (10, 9, 8, 7, 6, 5, 4, 3, 2, 1)$ chamaremos o dígito verificador de um número $f \in D$, ou seja, $f \in \{0, 1, ..., 9\}$ Daí:

$$\alpha \cdot \beta = (0, 3, 8, 7, 9, 6, 0, 3, 5, f) \cdot (10, 9, 8, 7, 6, 5, 4, 3, 2, 1)$$

$$0.10 + 3.9 + 8.8 + 7.7 + 9.6 + 6.5 + 0.4 + 3.3 + 5.2 + f$$

$$0 + 27 + 64 + 49 + 54 + 30 + 0 + 9 + 10 + f$$

$$243 + f \equiv 0 (mod\ 11)$$

Ou seja, como $11|243 + f$, implica que existe um número inteiro g tal que:

$$243 + f = 11.g \Rightarrow f = 11.g - 243$$

Portanto, g deve ser escolhido de tal forma que torne verdadeira a igualdade. Em particular, temos que se $g = 23$, implica $f = 10$. Porém, como $f \in D$, associaremos o dígito 10 ao símbolo X.

Daí, o código do livro citado será ISBN 0-387-96035-X.

É possível perceber que, se atribuirmos um número primo ao número k e o conjunto D formado por inteiros menores que k tendo todos os componentes p_i do vetor de peso β primos com k , multiplicar por p_i , módulo k , terá com resultado uma permutação do conjunto D, ou seja, uma bijeção de D em si mesmo. Com isso, temos um método mais geral para definir o vetor de pesos β.

Consideremos um vetor de informação $\alpha = (a_1, ..., a_{n-1})$, sem o dígito verificador a_n. Podemos escolher n permutações $d_1, ..., d_n$ do conjunto D e definir um vetor de pesos por $\omega = d_1, ..., d_n$, fixarmos um número $c \in D$ e escolhermos um dígito de verificação de forma que satisfaça a igualdade:

$$\omega(\alpha) = d_1(a_1) + ... + d_n(a_n) \equiv c (mod\ k)$$

Isso é, o dígito verificador a_n será definido pela igualdade:

$$a_n = d_n^{-1}\left(c - \sum_{i=1}^{n=1} d_i(a_i)\right)$$

Como exemplo dessa codificação, que encontra-se em [6], paginas 12-13, se um código usado pela IBM (International Business Machines) que utiliza como conjunto D os números de 0 a 9, $k = 10$, um valor qualquer $c \in D$ e a permutação

$$\delta = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ 0 & 2 & 4 & 6 & 8 & 1 & 3 & 5 & 7 & 9 \end{pmatrix}$$

Quando o número de componentes de α for **ímpar** utiliza-se o vetor de pesos: $\gamma = (I, \delta, I, \delta, ..., \delta, I)$.

Daí, a equação verificação será:

$$a_n + \delta(a_{n-1}) + a_{n-2} + \delta(a_{n-3}) + ... \equiv c(mod 10)$$

$$a_n = (c - (\delta(a_{n-1}) - a_{n-2} - \delta(a_{n-3}) - ...)(mod 10)$$

Quando o número de componentes de α é *par* utiliza-se o vetor de pesos: $\gamma = (\delta, I, \delta, I, ..., \delta, I)$.

Logo, a equação que determina o dígito verificador será idêntica e com fórmula parecida com o caso anterior.

Como aplicação desse sistema, temos o método de determinação dos **cartões de crédito**. Ainda em MILIES 2009, páginas 12-13, tomemos como exemplo, um determinado cartão tem o número 5745 5195 0431 5412. Aplicando o sistema de verificação IBM teremos:

5	7	4	5	5	1	9	5	0	4	3	1	5	4	1	2
$\downarrow \sigma$	$\downarrow I$	$\downarrow \sigma$	$\downarrow I$	$\downarrow \sigma$	$\downarrow I$	$\downarrow \sigma$	$\downarrow I$	$\downarrow \sigma$	$\downarrow I$	$\downarrow \sigma$	$\downarrow I$	$\downarrow \sigma$	$\downarrow I$	$\downarrow \sigma$	$\downarrow I$
1	7	8	7	1	1	9	1	0	4	6	1	1	9	2	2

Efetuando os cálculos teremos:

$$1 + 7 + 8 + 5 + 1 + 1 + 9 + 5 + 0 + 4 + 6 + 1 + 1 + 4 + 2 + 2 = 60 \equiv 0(mod\ 10)$$

De forma geral, o código IBM generalizado utiliza a mesma permutação com o vetor pesos:

$$\gamma = \left(\delta^{n-1}, \delta^{n-2}, ..., \delta, \delta^0\right)$$

onde $\delta^0 = I$. A equação é então:

$$\sum_{i-1}^{n} \delta^{i-1}(a_{n+1-i}) \equiv c(mod\ 10)$$

e

$$a_n = \left(c - \sum_{i-1}^{n} \delta^{i-1}\right)(a_{n+1-i})(mod\ 10)$$

■

O próximo teorema servirá de resposta para seguinte pergunta: será que existe algum código, trabalhando módulo 10, que identifique tanto os erros singulares e de transposições? A resposta para tal pergunta será negativa, como veremos a seguir.

Teorema 3.2. *Se um sistema numérico de detecção de erros, com um módulo par, detecta todo erro único de digitação, então para todo par de índices i, j existe um erro de transposição entre as posições i e j que não é detectada pelo sistema.*

Demonstração:
Já que o módulo é par, então iremos trabalhar com os números de 0 a $2k-1$ e com congruências em módulo $2k$. Considerando os dígitos como elementos de Z_{2k} (com o objetivo de simplificar os argumentos).

Suponha que o sistema transforma o vetor $(a_1, ..., a_n)$ num outro vetor, cuja notação será

$$(\sigma_1(a_1), ..., \sigma_n(a_n))$$

Tendo a capacidade de detecção do erro único de digitação, então a aplicação na posição i-ésima $x \mapsto \sigma_i(x)$ deve ser uma permutação de Z_{2k}

Para que o sistema detecte todo erro de transposição entre as posições i e j é necessário que

$$\sigma_i(a) + \sigma_i(b) \neq \sigma_j(a) + \sigma_i(b)$$

para todo par de elementos diferentes a , $b \in Z_{2k}$, ou seja, esta aplicação $\sigma = \sigma_i - \sigma_j$ é uma permutação em Z_{2k}.

Porém, como $k \in [0, 2k-1]$, temos que:

$$0 + 1 + 2 + ... + 2k - 1 = \frac{(2k-1)2k}{2} = 2k.k - k \equiv k(mod\ 2k)$$

Como $2k.k - k \equiv k(mod\ 2k)$, então $0 + 1 + 2 + ... + 2k - 1 \equiv k(mod\ 2k)$.

Logo

$$k = \sum_{x \in Z_{2k}} \cdot x =$$

$$k = \sum_{x \in Z_{2k}} \cdot \sigma(x)$$

$$k = b \cdot \sum_{x \in Z_{2k}} \cdot (\sigma_i(x) - \sigma_j(x))$$

$$k = \sum_{x \in Z_{2k}} \cdot \sigma_i(x) - \sum_{x \in Z_{2k}} \cdot \sigma_j(x)$$

$$k = k - k$$

$$k = 0$$

Mas, isso é uma contradição, pois k é um inteiro positivo.

■

Nos últimos anos, o desenvolvimento dos sistemas automáticos para leitura de números, rápidos, confiáveis e relativamente baratos, permitiu a justaposição dos algarismos de controle ao número de um código, para detectar erros mais comuns. Os sistemas não corrigem os erros,

porém "avisam´´ que eles foram cometidos.

Quando um código é digitado, o computador no qual está instalado o sistema de identificação aplica o algoritmo teste, para verificar se o último algarismo é de fato o mesmo algarismo que o algoritmo aplica ao código sem proteção.

CAPÍTULO 4

ATIVIDADES PROPOSTAS

Afim de aplicar o conteúdo visto nesse trabalho, as atividades a seguir terão como finalidade fazer com que a matemática tenha maior aplicabilidade no dia-a-dia dos alunos. Com essa proposta, espera-se que os mesmos consigam associar o conteúdo visto em sala de aula com situações do seu cotidiano.

4.1 Atividade 1 - Reconhecendo um Código de Barras

Nessa atividade iremos reconhecer um código de barras, bem como entender a sua estrutura e funcionalidade.

4.1.1 Objetivo Geral

Reconhecer a importância dos códigos de barras em sala de aula, afim de mostrar o quanto seria mais lento o reconhecimento de um produto sem que utilizasse um código para identificá-lo.

4.1.2 Objetivos Específicos

- Identificar um código de barras
- Reconhecer e classificar os números ali contidos

- Compreender a sua funcionalidade no cotidiano

4.1.3 Público Alvo

Alunos do 6º ano do ensino fundamental II

4.1.4 Pré-Requisitos

Os alunos deverão saber interpretar informações, associando-as ao contexto do trabalho.

4.1.5 Materiais Necessários

Os materiais a serem utilizados nessa atividade são:

- lápis e borracha - para anotar (e apagar, se for o caso) as informações necessárias;
- papel ofício (para colar os códigos de barras selecionados)
- tesoura (para fazer os recortes de códigos de barras);)
- cola (para colar os códigos de barras)

4.1.6 Proposta da Atividade

Inicialmente, temos que entender o que é um código de barras e qual a sua funcionalidade. Portanto, para isso usaremos um exemplo e faremos passo-a-passo a atividade proposta.

Solicitar que os alunos tragam códigos de barras, em seguida que eles identifiquem os números que caracterizam o país de origem, a empresa fabricante, o produto e o dígito verificador.

1º passo: Solicitar os materiais necessários para realização da atividade.

2º passo: Deverá ser feita uma explanação, pelo docente, sobre o que são os códigos de barras, para que servem, sua estrutura e onde são utilizados.

3º passo: Dividir a turma em grupos com, no máximo 5 alunos.

4º passo: Será feita a interpretação dos códigos trazidos por cada aluno. Os alunos devem recortar os códigos trazidos, colá-los nas folhas de papel ofício e anotar todas as informações contidas nos códigos por eles trazidos, tais como país de origem, os números que representam a empresa, o produto e o dígito verificador.

Exemplo 4.1. *Considere o código de barras da Figura 4.1:*

Figura 4.1: Exemplo da Atividade I

- 789 são os dígitos que caracterizam o país, nesse caso o Brasil;
- 8357 representam os dígitos que caracterizam a empresa;
- 4100 caracterizam o produto;
- 5 é o dígito verificador

4.2 Atividade 2 - Cálculo do Dígito Verificador

Nesta atividade, iremos entender o que é o dígito verificador, sua importância e como determiná-lo.

Como sugestão, o docente deverá aplicar a atividade anterior, afim de explicar o que são códigos de barras e sua funcionalidade.

4.2.1 Objetivo Geral

Reconhecer a importância do dígito verificador, que é fator determinante para a veracidade de um código de barras, bem como determiná-lo por um algoritmo.

4.2.2 Objetivos Específicos

- Reconhecer o dígito verificador em um código de barras;
- Compreender a funcionalidade do dígito verificador;
- Entender e aplicar o algoritmo utilizado para determinação do dígito verificador.

4.2.3 Público Alvo

Alunos do 7º ano do ensino fundamental II.

4.2.4 Pré-Requisitos

- Realizar as operações com números naturais, tais como adição, subtração, multiplicação e divisão;
- Reconhecer quando um número é divisível pelo outro, aplicando os critérios de divisibilidade;
- Compreender informações contidas nos códigos de barras;
- Entender e aplicar um algoritmo.

4.2.5 Materiais Necessários

Lápis, borracha e uma folha contendo os códigos de barras.

4.2.6 Proposta da Atividade

Os alunos irão reconhecer a importância do dígito verificador e determiná-lo através de um algoritmo. Para isso, veremos o exemplo a seguir, passo-a-passo, de como deve ser feita a atividade.

Exemplo 4.2. *Consideremos o código 789162731405. Determine o dígito verificador desse código.*

Sugere-se que a atividade seja feita em duplas, afim de proporcionar uma discussão acerca do resultado encontrado. Solicitar códigos de barras, que podem ser trazidos pelo professor

ou solicitado, previamente, aos alunos.

Deve-se explicar como é feita a verificação da validade do código, isso é, explicando como é feito o algoritmo para a determinação do mesmo.

Segundo [7] , **algoritmo** é uma sequência de instruções que podem ser executadas mecanicamente, por uma pessoa ou uma máquina (computador).

1º passo: Suponhamos que estamos usando o código de barras dado no exemplo e queremos saber qual é o dígito final (verificador).

2º passo: Somamos todos os dígitos das posições ímpares (primeira, terceira, quinta, etc). Portanto:

$$7 + 9 + 6 + 7 + 1 + 0 = 30$$

3º passo: Multiplicamos todos os dígitos das posições pares (segunda, quarta, sexta, etc) por 3 e somamos esses produtos:

$$8x3 + 1x3 + 2x3 + 3x3 + 4x3 + 5x3 = 69$$

4º passo: Somamos os dois resultados das etapas anteriores. Daí:

$$30 + 69 = 99$$

5º passo: Determine o número que deve ser adicionado ao resultado da soma para se criar um múltiplo de 10. Portanto:

$$99 + 1 = 100$$

Portanto, o dígito verificador é **1**.

Afim de dinamizar mais a atividade, seria interessante que o professor solicitasse que eles troquem um dos dígitos do código de barras e depois verifiquem se o resultado obtido coincide com o dígito verificador.

CONCLUSÃO

Indubitavelmente, a presença da matemática é notória em vários segmentos da tecnologia que surgiu ou que ainda irá surgir. O surgimento dos códigos de barras foi de suma importância para que uma ida ao supermercado, por exemplo, e registrar suas compras, deixasse de ser algo tão lento, contribuindo, então, para o progresso, em particular, do comércio mundial.

Como fora exposto neste trabalho, a presença do código de barras tornou-se uma tecnologia indispensável nos dias atuais, auxiliando as empresas no controle e identificação de produtos, bem como na agilização do atendimento ao cliente, atendimento bancário e entre diversos setores.

Os códigos de barras devem ser controlados para tornar possível a detecção de erros de codificação. Isso significa que quando, por exemplo, um caixa do supermercado digitar um número errado do código que identifica o produto, esse erro de codificação deve ser "avisado". Para isso, criou-se um grupo de algarismos, denominados *dígitos verificadores*.

Neste sentido, ao abordar o tema Aritmética Modular nos Códigos de Barras, reiteramos conceitos matemáticos imprescindíveis para a realização do tema, sendo aplicados nos Sistemas de Identificação Modular (SIM) e na Detecção de Erros.

Por fim, foram propostas duas situações de aprendizagem articulando o real e o fictício, onde além de proporcionar um ambiente investigativo e criativo, será possível fazer conexões

entre diversos conceitos matemáticos, suas diferentes formas de pensamento, e ainda, promover a relação da Matemática com outras áreas do saber e da atualidade, valorizando a tendência atual.

REFERÊNCIAS BIBLIOGRÁFICAS

[1] DOMINGUES; H. Hygino:**Fundamentos de aritmética**; São Paulo, Atual, 1991.

[2] HEFEZ, Abramo: **Elementos de Aritmética**; 2ª ed, Rio de Janeiro, SBM, 2011.

[3] HEFEZ, Abramo: **Iniciação à Aritmética** - Programa de Iniciação Científica, OBMEP, Ed. da SBM, Rio de Janeiro-RJ, 2012.

[4] OLIVEIRA, Krerley Irraciel Martins: FERNANDEZ, Adán José Carcho: **Iniciação à Matemática: um curso com problemas e soluções**, 2ª ed., Rio de Janeiro, SBM, 2010.

[5] AVRITZER, Dan: Geometria Analítica e Álgebra Linear: uma visão geométrica. Belo Horizonte. Editora UFMG, 2009.

[6] POLCINO MILIES, C. **A matemática dos códigos de barras**. Programa de Iniciação Científica da OBMEP. Rio de Janeiro: OBMEP, 2009, v., p. 131-179.

[7] FINI, Maria Inês: **Controle dos Códigos de Identificação**. Revista do Professor - Atualidades, SEESP, Edição nº2, p. 70 - 75, 2009.

[8] BARCODE ISLAND. Disponível em: <http://www.barcodeisland.com/ean13.phtml>. Acesso em: 01 de dezembro de 2015.

[9] GS1 BRASIL (Associação Brasileira de Automação). Disponível em: <http://www.gs1br.org>. Acesso em: 01 de dezembro de 2015.

[10] Portal ItDirect. Disponível em: <http://itdirectbrasil.blogspot.com.br/2013/01/codigos-2d-ou-qr-code.html>. Acesso em: 01 de dezembro de 2015.

[11] QR codes e Realidade Aumentada, por Martha Gabriel. Disponível em: <http://pt.slideshare.net/marthagabriel/qr-codes-realidade-aumentada-por-martha-gabriel>. Acesso em: 01 de dezembro de 2015.

[12] Código de Barras Code 128, Código 128, GS1 128, EAN 128, UPC 128. Disponível em: <http://www.code128.com.br/tag/codigos-de-barras>. Acesso em: 01 de dezembro de 2015.

[13] Código de Barras EAN. Disponível em: <https://www.codigodebarrasean.com/codigo-de-barras-imagem-ean.html>. Acesso em: 01 de dezembro de 2015.

[14] COSTA, F. R. A. **Sistemas de Identificação Modular: uma aplicação no ensino fundamental**. 2014. 22 f. Trabalho de Conclusão de Curso do Mestrado Profissional em Matemática - PROFMAT. Ouro Branco, MG: Universidade Federal de São João del-Rei.