Aritmética modular e suas aplicações: Dos sistemas de identificação às mensagens secretas

Cláudia Fernanda Ribeiro Seabra Santos

DEPARTAMENTO DE MATEMÁTICA
FACULDADE DE CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE DE COIMBRA

Aritmética modular e suas aplicações:
Dos sistemas de identificação às mensagens secretas

Cláudia Fernanda Ribeiro Seabra Santos

Relatório para a obtenção do Grau de **Mestre em Ensino da Matemática**

no 3º ciclo do Ensino Básico e no Ensino Secundário

Júri

Presidente: Professora Doutora Natália Isabel Quadros Bebiano Pinheiro da Providência e Costa

Orientador: Professor Doutor Armando Duarte da Silva Gonçalves

Vogal: Professor Doutor Carlos Martins da Fonseca

Data: Julho de 2013

Resumo

A aritmética modular é uma ferramenta importante da teoria de números. Consiste em trabalhar com o resto da divisão inteira por um determinado número. Esta aritmética está na base da conceção de vários sistemas de identificação, presentes na nossa vida quotidiana, por exemplo em livros, cartões e artigos. É também utilizada na codificação e descodificação de mensagens secretas.

Neste relatório pretendemos fazer, numa primeira parte, uma análise de alguns sistemas de identificação modulares mais utilizados, reconhecendo as suas limitações relativamente à deteção de erros ocorridos na transmissão de números de identificação. Numa segunda parte, pretendemos apresentar, em contexto escolar, aplicações da aritmética modular no dia-a-dia, recorrendo a exemplos animados, práticos e palpáveis, bem como a programas informáticos apropriados.

Palavras-chave: Aritmética modular, Sistema de Verhoeff, Criptografia, Ensino

Abstract

The modular arithmetic is an important tool of number theory. It consists in working with the remainder of the integer division. This arithmetic is in the basis of the conception of several identification systems, and it appears, for example in books, cards and articles. It is also used in the encryption and decryption of secret messages.

In the first part of this report we intend to do an analysis of some widely used modular identification systems, recognizing their limitations regarding the detection of errors in the transmission of identification numbers. In the second part, we explain how we showed, in a school context, daily applications of modular arithmetic, using animated, practical and tangible examples, as well as the appropriate software programs.

Keywords: Modular arithmetic, Verhoeff system, Cryptography, Teaching

Agradecimentos

Cumpre-nos manifestar algumas palavras de apreço a todos aqueles que direta ou indiretamente contribuíram para a concretização deste trabalho.

Ao Professor Doutor Armando Gonçalves, agradecemos o apoio, a disponibilidade, o encorajamento, as críticas e sugestões oportunas, bem como a sugestão do tema deste trabalho.

Aos colaboradores da escola de hotelaria e turismo de Coimbra, nomeadamente à direção da escola, em particular à diretora, Dr.ª Ana Paula Pais e à coordenadora de formação Dr.ª Maria Antónia Portugal, agradecemos o apoio na conjugação de compromissos profissionais e a compreensão em momentos de menor disponibilidade. Aos colaboradores D.ª Manuela Jacinto e Sr. Paulo Ferreira agradecemos todo o apoio prestado.

Aos colegas e amigos que nos acompanharam ao longo da realização deste trabalho, agradecemos a ajuda, o incentivo, as críticas construtivas e as palavras de conforto.

Aos familiares, pais, sogros, irmão e cunhadas, agradecemos o apoio disponibilizado. Em particular aos pais, Judite e Fernando, agradecemos os valores que nos transmitiram.

Ao marido Paulo, alicerce da nossa vida, agradecemos a paciência, a compreensão e o auxílio. Não encontramos palavras suficientes para agradecer o amor incondicional. Apenas podemos retribuí-lo.

Aos filhos Francisco e Sofia, agradecemos a sua existência.

Índice

1 Introdução

"Osrebvnado anetmaente etsa fsrae, vrfieacoms que ebmrora a oderm das Lteras das plravaas não etseja crtea, cnseigumoos prebceer a msenasegm."

Numa frase deste tipo, a troca de letras nas palavras é facilmente detetável, uma vez que o sentido da mensagem dá-nos a chave para encontrar as palavras certas. No entanto, se a mensagem for constituída por números, será muito mais difícil recuperar a mensagem que pretendia ser enviada. Foram, por isso, desenvolvidas algumas técnicas para detetar interferências que modificam a mensagem inicial, quando o contexto da mesma não é suscetível de ser identificado pelo sentido.

Neste relatório apresentamos, numa primeira parte, o desenvolvimento de alguns sistemas de identificação detetores de erros, que podem ocorrer na transmissão de uma mensagem. Numa segunda parte, apresentamos algumas aplicações, na vida quotidiana, dos sistemas abordados na primeira parte, bem como exemplos elementares cuja escolha foi motivada para posterior apresentação em sala de aula.

2 Projeto educacional I

No projeto educacional I fazemos uma abordagem aos sistemas de identificação de erros que podem ocorrer no processo de transmissão de uma informação numérica, partindo de um emissor para um recetor. Identificamos os erros mais comuns e referenciamos os sistemas de deteção de erros mais conhecidos, os mais e menos eficientes e os mais e menos utilizados. Relativamente aos sistemas mais utilizados, mas menos eficientes, os sistemas de identificação modulares, nomeadamente o sistema ISBN, os sistemas UPC e EAN, o sistema dos cheques bancários, o sistema de identificação do bilhete de identidade e o sistema de identificação do cartão de cidadão, fazemos uma análise mais aprofundada.

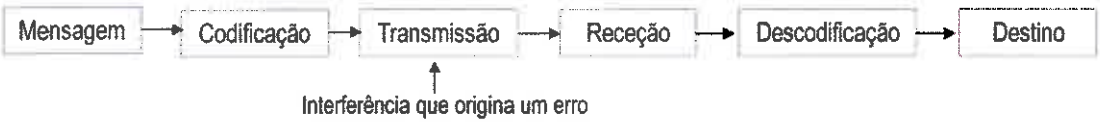
2.1. Referência histórica

A necessidade de comunicar foi um fator determinante para o desenvolvimento civilizacional uma vez que as civilizações se estabelecem com base na comunicação, ou seja, na partilha de ideias e informações. Ora, a capacidade de comunicar, é a capacidade de transmitir e receber mensagens. Durante o processo de comunicação, para além de um emissor e de um recetor, há mais quatro fases: a codificação, a transmissão, a receção e a descodificação. Na primeira, a informação contida na mensagem é codificada numa linguagem ou em séries de símbolos. Na

segunda, a linguagem, processada para viajar através de um meio, é transmitida. Na terceira, a mensagem transmitida é recebida e convertida novamente em linguagem. Por fim, a linguagem é descodificada na informação original.

Uma vez que a linguagem dos números é universal, a sua transmissão e descodificação torna-se mais simples. Contudo, se durante esse processo ocorrerem interferências que modifiquem a mensagem original, também será mais complicado recuperá-la. A figura 2.1.1 ilustra as várias fases do processo de comunicação:

Figura 2.1.1.



Com a crescente necessidade de transmitir grandes quantidades de informação numérica, surgiu a importância de desenvolver uma teoria que otimizasse a codificação de uma informação, de modo a que a sua receção fosse descodificada sem erros, ou que, pelo menos, os detetasse. Nalguns sistemas de identificação numéricos, a deteção desses erros é feita acrescentando uma informação adicional à mensagem original, que permita verificar a existência dos erros ocorridos na transmissão da mensagem. Estes sistemas são normalmente chamados de sistemas ou códigos de identificação detetores de erros, ou mesmo sistemas de identificação com algarismos de teste.

Estas questões começaram a ser tratadas na teoria de informação de Claude Shannon (1916 – 2001), considerado o "pai da teoria da informação", uma vez que levantou o problema de encontrar a melhor forma de codificar uma informação, que um remetente deseja transmitir a um recetor, no seu clássico artigo "*A Mathematical Theory of Communication*", publicado em 1948 no Bell System Technical Journal, nos Estados Unidos.

Figura 2.1.2. Claude Shanon 1916 - 2001

Outros cientistas como Richard W. Hamming, Marcel Golay, Claude Berrou entre outros, continuaram o trabalho de Shannon, desenvolvendo-o até chegar à teoria de informação tal como hoje é conhecida (Anexo 1).

2.2. Tipos de erros

A conceção dos sistemas de identificação com algarismos de teste, tem por base a identificação dos erros ocorridos na transmissão de uma mensagem, e a frequência com que ocorrem.

Cerca de 90% dos erros que ocorrem na transmissão de dados numéricos são de dois tipos:

Erros singulares, em que existe a alteração de apenas um algarismo, como por exemplo,

$$2345 \rightarrow 2348.$$

Transposições de algarismos adjacentes, em que se verifica a troca de pares de algarismos adjacentes, como por exemplo,

$$2345 \rightarrow 2354.$$

Existem outros tipos de erros, como as transposições intercaladas, os erros gémeos, os erros fonéticos, os erros gémeos intercalados e outros erros aleatórios, que são, no entanto, menos frequentes conforme se pode verificar na tabela 2.2.1:

Tabela 2.2.1. Tipos de erros mais comuns:

Tipos de erros		Frequência relativa (%)
Erros singulares	$a \rightarrow b$	79,1
Transposições de algarismos adjacentes	$ab \rightarrow ba$	10,2
Transposições intercaladas	$acb \rightarrow bac$	0,8
Erros gémeos	$aa \rightarrow bb$	0,5
Erros fonéticos	$a0 \rightarrow 1a\ (a = 2,3,\cdots,9)$	0,5
Erros gémeos intercalados	$aca \rightarrow bcb$	0,3
Erros aleatórios		8,6

Após a perceção dos erros mais frequentes na utilização sistemática de grandes quantidades de números, surgiu a necessidade de encontrar uma forma de detetar, assim que o número é escrito, a presença ou ausência daqueles erros, concebendo algoritmos eficientes. Grande parte desses algoritmos assenta na aritmética modular, pelo que apresentaremos algumas noções elementares relativas a essa aritmética.

2.3. Noções elementares de aritmética modular

A aritmética modular é uma ferramenta da teoria de números que envolve o conceito de congruência. Consiste em trabalhar com o resto da divisão inteira por um determinado número.

Definição 2.1 Relação de congruência *módulo k*

Dois números inteiros a e b dizem-se congruentes, *módulo k*, se tiverem o mesmo resto quando divididos pelo mesmo número inteiro k $(k > 0)$. Essa relação representa-se normalmente por:

$$a \equiv b \bmod k$$

Outro modo equivalente de dizer que a e b são congruentes *módulo k*, é verificar que a diferença $a - b$ *ou* $b - a$ é divisível por *k* (ou que *k é* divisor dessa diferença).

A relação de congruência satisfaz as seguintes **propriedades**:

P1) É uma relação de equivalência, isto é, para quaisquer $a, b, c \in \mathbb{Z}$:

- $a \equiv a \bmod k$ (reflexiva)
- *Se* $a \equiv b \bmod k$, então $b \equiv a \bmod k$ (simétrica)
- *Se* $a \equiv b \bmod k$ e $b \equiv c \bmod k$, então $a \equiv c \bmod k$ (transitiva)

P2) Se $a \equiv b \bmod k$ e $c \equiv d \bmod k$, então $a + c \equiv b + d \bmod k$

P3) Se $a \equiv b \bmod k$ e $c \equiv d \bmod k$, então $a - c \equiv b - d \bmod k$

P4) Se $a \equiv b \bmod k$ e $c \equiv d \bmod k$, então $a \times c \equiv b \times d \bmod k$

As propriedades anteriores são facilmente demonstráveis. Por exemplo, na propriedade P2 pretende-se provar que $a + c \equiv b + d \bmod k$, ou seja, que

$$k \Big| \big((a + c) - (b + d)\big).$$

Ora, $(a + c) - (b + d) = (a - b) + (c - d)$. Por hipótese $k \Big| (a - b)$ e $k \Big| (c - d)$, logo $k \Big| \big((a - b) + (c - d)\big) \Leftrightarrow k \Big| \big((a + c) - (b + d)\big)$ e portanto

$$a + c \equiv b + d \bmod k. \quad \blacksquare$$

2.4. Sistemas de identificação modulares

Como já referimos, a ideia base na verificação da fiabilidade de um código numérico consiste em acrescentar uma informação adicional, redundante, à informação inicial. Partindo desta ideia, nos sistemas de identificação modulares, ao número que se quer verificar se existe erro, é acrescentado um algarismo suplementar, chamado algarismo de teste ou dígito de controlo, que se obtém realizando uma operação com os algarismos do número original, através da aritmética modular. Se tal não acontecer, significa que ocorreu um erro na transmissão da escrita do número original. Estes sistemas, também chamados sistemas *módulo k*, não corrigem os erros ocorridos, apenas os detetam e com uma fiabilidade relativa. Um sistema *módulo k* será mais eficiente se *k* for um número primo, como verificaremos na análise de alguns dos sistemas de identificação modulares apresentados.

2.4.1. Sistema ISBN (International Standard Book Number)

O sistema ISBN surgiu com a necessidade que as livrarias tiveram em catalogar os seus livros e informatizar o sistema de encomendas.

O ISBN é constituído por 10 algarismos que identificam o livro, do seguinte modo:

$$\underbrace{a_1\ a_2\ a_3\ a_4\ a_5\ a_6\ a_7\ a_8\ a_9}_{\text{Identificação do livro}}\ \boxed{a_{10}} \rightarrow \text{Algarismo de teste ou de controlo}$$

a_{10} é escolhido de modo a que a soma de teste

$$S = (10a_1 + 9a_2 + 8a_3 + \cdots + a_{10}),$$

seja congruente com $0\ mod\ 11$, ou seja, que S seja divisível por 11.

Assim,

$$\begin{aligned} a_{10} &= -(10a_1 + 9a_2 + 8a_3 + \cdots 2a_9) \\ &= -[(11-1)a_1 + (11-2)a_2 + \cdots + (11-9)a_9] \Leftrightarrow \end{aligned}$$

$$a_{10} = -\sum_{i=1}^{9}(11-i)a_i$$

Se existir um erro singular, de transposição ou outro semelhante na.transmissão do número, o resultado não deverá ser divisível por 11.

Considere-se o seguinte exemplo.

Exemplo 2.4.1

O livro "*Introduction to finite fields and their applications*" [4], tem o seguinte ISBN:

$$ISBN\ 0-521-46094-8$$

Neste caso, o algarismo de teste é o 8.

Verifiquemos se é correto:

$$-(10\times 0+9\times 5+8\times 2+7\times 1+6\times 4+5\times 6+4\times 0+3\times 9+2\times 4)=-157$$

Ora $-157\equiv 8\ mod\ 11$, pelo que podemos concluir que não ocorreu um erro singular, de transposição ou outro semelhante na transmissão do número.

Este sistema foi concebido de modo a detetar os erros singulares e os de transposição, que são os normalmente cometidos por operadores humanos, aquando da comunicação de números longos. De facto, a seguinte proposição demonstra isso mesmo:

Proposição 2.4.1

Suponhamos que na leitura de um dado número ISBN, ocorre um e um só dos dois seguintes erros: um erro singular **ou** uma transposição. Então a soma de teste não é um múltiplo de 11.

Demonstração:

1.º caso: "ocorre um erro singular"

Seja $a_1\ a_2\ a_3\ a_4\ a_5\ a_6\ a_7\ a_8\ a_9\ a_{10}$ um número ISBN com a ocorrência de um erro singular na i-ésima posição, com $i\epsilon\{1,\dots 10\}$, tal que $a_1\dots a_i'\dots a_{10}$. → Algarismo incorreto

Sejam S e S' as somas de teste do número correto e do número errado, respetivamente.

Ora, $S\equiv 0\ (mod\ 11)$, por definição de S, sendo que

$$S=10a_1+9a_2+\cdots+(11-i)a_i+\cdots+a_{10}\ \text{e}$$
$$S'=10a_1+9a_2+\cdots+(11-i)a_i'+\cdots+a_{10}.$$

Assim,

$$S-S'=10a_1+9a_2+\cdots+(11-i)a_i+\cdots+a_{10}-(10a_1+9a_2+\cdots+(11-i)a_i'+\cdots+a_{10})$$
$$S-S'=(11-i)a_i+(11-i)a_i'=(11-i)(a_i-a_i').$$

É necessário provar que a soma de teste S' não é um múltiplo de 11.

De facto, se S' fosse um múltiplo de 11, 11 dividiria S', com

$$S'=S+(11-i)(a_i-a_i'),$$

ou seja, $S' = 11 \times n, (n \in \mathbb{N}) \Leftrightarrow n = \frac{S'}{11} \Leftrightarrow n = \frac{S+(11-i)(a_i-a_i')}{11} \Leftrightarrow n = \frac{S}{11} + \frac{(11-i)(a_i-a_i')}{11}$.

n.º inteiro pois $S \equiv 0 \ (mod\ 11)$

Logo $\frac{(11-i)(a_i-a_i')}{11}$ teria de ser um número inteiro e portanto

$$\underbrace{11 \mid (11-i) \text{ ou } 11 \mid (a_i' - a_i)}.$$

Absurdo pois $11 - i$ é um n.º inteiro entre 1 e 10 e $a_i' - a_i$ é um n.º inteiro não nulo entre -10 e 10

Portanto S' não é um múltiplo de 11.

2.º caso: "ocorre uma transposição"

Seja $a_1 \dots a_i \dots a_j \dots a_{10}$ um número ISBN com a ocorrência de uma transposição dos algarismos a_i e a_j nas posições i e j, com $i, j \in \{1, \cdots, 10\}$ e $i \neq j$.

Neste caso, o número errado será $a_1 \dots a_j \dots a_i \dots a_{10}$ e pretendemos provar que a sua soma de teste, S', não é um múltiplo de 11, ou seja, que

$$S' = 10a_1 + 9a_2 + \cdots + (11-i)a_j + \cdots + (11-j)a_i + \cdots + a_{10}$$

não é um múltiplo de 11. Ora,

$S = 10a_1 + 9a_2 + \cdots + (11-i)a_i + \cdots + (11-j)a_j + \cdots + a_{10}$, pelo que

$$S - S' = 10a_1 + \cdots + (11-i)a_i + \cdots + (11-j)a_j + \cdots + a_{10}$$
$$-(10a_1 + \cdots + (11-i)a_j + \cdots + (11-j)a_i + \cdots + a_{10}) \Leftrightarrow$$
$$S - S' = (a_j - a_i)(j-i).$$

De facto, se S' fosse um múltiplo de 11, 11 dividiria S', com

$S' = S + (a_j - a_i)(j-i)$, ou seja,

$S' = 11 \times n, (n \in \mathbb{N}) \Leftrightarrow n = \frac{S'}{11} \Leftrightarrow n = \frac{S+(a_j-a_i)(j-i)}{11} \Leftrightarrow n = \frac{S}{11} + \frac{(a_j-a_i)(j-i)}{11}$

n.º inteiro pois $S \equiv 0\ mod\ 11$

Logo $\frac{(a_j-a_i)(j-i)}{11}$ teria de ser um número inteiro e portanto

$$\underbrace{11 \mid (a_j - a_i) \text{ ou } 11 \mid (j-i)}$$

Absurdo pois $a_j - a_i$ é um n.º inteiro não nulo entre -10 e 10, pois $a_j \neq a_i$ e $(j-i)$ é um n.º inteiro não nulo entre -10 e 10, pois $(j \neq i)$

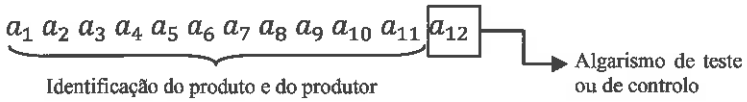
Portanto S' não é um múltiplo de 11.

Em cada um dos dois casos verificou-se que a soma de teste, S', onde ocorre o erro, não pode ser um número múltiplo de 11. ■

2.4.2. Sistemas UPC (Universal Product Code) e EAN (European Article Number)

Estes sistemas foram concebidos para que os números fossem lidos e comunicados por leitores óticos. Segundo os testes de qualidade realizados, estes leitores são tão precisos que, quando muito, cometem erros singulares, pelo que o sistema foi concebido para detetar apenas este tipo de erros.

O UPC é um número constituído por 12 algarismos:



O último algarismo é o dígito de controlo, que permite ao leitor ótico verificar se o número está correto.

Nota 2.4.1

No sistema UPC, o último algarismo do número de identificação, o dígito de controlo a_{12}, é escolhido de modo a que a soma de teste

$$S = (3a_1 + a_2 + 3a_3 + \cdots + 3a_{11} + a_{12}) \equiv 0\ mod\ 10,$$

ou seja, que S seja divisível por 10. Assim,

$$a_{12} \equiv -(3a_1 + a_2 + 3a_3 + \cdots + 3a_{11})\ mod\ 10.$$

No caso do sistema EAN, os números de identificação são constituídos por 13 algarismos, em que o décimo terceiro algarismo será, novamente, o algarismo de teste, escolhido da seguinte forma:

$$a_{13} \equiv -(a_1 + 3a_2 + a_3 + \cdots + a_{11} + 3a_{12})\ mod\ 10,$$

tal que a soma de teste $\boldsymbol{S = (a_1 + 3a_2 + a_3 + \cdots + a_{11} + 3a_{12} + a_{13}) \equiv 0\ mod\ 10.}$

Proposição 2.4.2.

Os sistemas UPC e EAN detetam todos os erros singulares.

Demonstração:

Para provar esse facto, serão analisadas duas situações do sistema UPC:

1.º caso: quando o erro ocorre num algarismo de posição par

(o coeficiente desse algarismo em S é 1)

$$S = (3a_1 + a_2 + 3a_3 + \cdots + a_i + \cdots + 3a_{11} + a_{12}) \quad (i \text{ par})$$

$$S' = (3a_1 + a_2 + 3a_3 + \cdots + a_i' + \cdots + 3a_{11} + a_{12})$$

Algarismo incorreto

Então, $S' - S = a_i' - a_i$, caso i seja par.

Ora S é múltiplo de 10, isto é $10 \mid S$ (10 divide S).

Se S' fosse múltiplo de 10, então $10 \mid S'$ e portanto $10 \mid (S' - S) \Longrightarrow 10 \mid (a_i' - a_i)$, o que é absurdo pois $a_i' - a_i$ é um n.º inteiro não nulo entre -9 e 9.

Portanto S' não é múltiplo de 10. ■

2.º caso: quando o erro ocorre num algarismo de posição ímpar

(o coeficiente desse algarismo em S é 3)

$$S = (3a_1 + a_2 + 3a_3 + \cdots + 3a_i + \cdots + 3a_{11} + a_{12}) \quad (i \text{ ímpar})$$

$$S' = (3a_1 + a_2 + 3a_3 + \cdots + 3a_i' + \cdots + 3a_{11} + a_{12})$$

Algarismo incorreto

Então, $S' - S = 3(a_i' - a_i)$, caso i seja ímpar.

S é múltiplo de 10, isto é, $10 \mid S$.

Se S' fosse múltiplo de 10, então $10 \mid S'$ e portanto $10 \mid (S' - S)$, com

$$S' - S = 3(a_i' - a_i).$$

Assim, $3(a_i' - a_i) = 10 \times s' - 10 \times s$, com s' e s números inteiros pois, tínhamos por hipótese $10 \mid S'$ e $10 \mid S$. Ora,

$$3(a_i' - a_i) = 10(s' - s) \Rightarrow 10 \mid 3(a_i' - a_i) \xrightarrow[\text{10 não divide 3}]{} 10 \mid (a_i' - a_i),$$

o que é absurdo pois $a_i' - a_i$ é um número inteiro não nulo entre -9 e 9.

Portanto S' não é múltiplo de 10. ■

Como já foi descrito anteriormente, os sistemas UPC ou EAN foram concebidos apenas para detetar erros singulares pelo que não detetam todas as transposições de algarismos adjacentes.

Proposição 2.4.3.

Os sistemas UPC e EAN, não detetam as transposições de algarismos adjacentes sempre que $|a_{i+1} - a_i| = 5$.

Demonstração:

Considere-se dois números UPC.

1.º caso: quando a transposição ocorre entre os algarismos a_i e a_{i+1} para i um número par:

$$S = (3a_1 + a_2 + 3a_3 + \cdots + \overset{i\ par}{\overleftarrow{a_i}} + 3a_{i+1} + \cdots + 3a_{11} + a_{12})$$

$$S' = (3a_1 + a_2 + 3a_3 + \cdots + \underbrace{a_{i+1} + 3a_i}_{\text{Ocorreu uma transposição dos algarismos adjacentes } a_i \text{ e } a_{i+1}} + \cdots + 3a_{11} + a_{12})$$

$$\begin{aligned} S' - S &= a_{i+1} + 3a_i - (a_i + 3a_{i+1}) \\ &= a_{i+1} + 3a_i - a_i - 3a_{i+1}) \\ &= a_{i+1} - a_i + 3a_i - 3a_{i+1}) \\ &= a_{i+1} - a_i + 3(a_i - a_{i+1}) \\ &= a_{i+1} - a_i - 3(a_{i+1} - a_i) \\ &= -2(a_{i+1} - a_i) \\ &= 2(-a_{i+1} + a_i). \end{aligned}$$

No caso de i ser par, tem-se $S' - S = 2(-a_{i+1} + a_i)$.

No entanto, caso se verifique $-a_{i+1} + a_i = 5$ ou $-a_{i+1} + a_i = -5$, tem-se que $S' - S$ é múltiplo de 10 e portanto

$$10 \mid (S' - S) \Leftrightarrow 10 \mid 2(-a_{i+1} + a_i) \underbrace{\Longrightarrow}_{\overbrace{10\ não\ divide\ 2}^{\downarrow}} 10 \mid (-a_{i+1} + a_i).$$

2.º caso: quando a transposição ocorre entre os algarismos a_i e a_{i+1} para i número ímpar:

$$S = (3a_1 + a_2 + 3a_3 + \cdots + \overset{i\ ímpar}{\overleftarrow{3a_i}} + a_{i+1} + \cdots + 3a_{11} + a_{12})$$

$$S' = (3a_1 + a_2 + 3a_3 + \cdots + \underbrace{3a_{i+1} + a_i}_{\text{Ocorreu uma transposição dos algarismos adjacentes } a_i \text{ e } a_{i+1}} + \cdots + 3a_{11} + a_{12})$$

Assim,
$$\begin{aligned} S' - S &= 3a_{i+1} + a_i - (3a_i + a_{i+1}) \\ &= 3a_{i+1} + a_i - 3a_i - a_{i+1} \\ &= 3a_{i+1} - 3a_i - a_{i+1} + a_i \\ &= 3(a_{i+1} - a_i) - (a_{i+1} - a_i) \\ &= 2(a_{i+1} - a_i). \end{aligned}$$

Note-se ainda que $S' - S = 2(a_{i+1} - a_i)$, sempre que i é ímpar.

Do mesmo modo, caso se verifique $a_{i+1} - a_i = 5$ ou $a_{i+1} - a_i = -5$, tem-se que $S' - S$ é múltiplo de 10 e portanto

$$10 \mid (S' - S) \Leftrightarrow 10 \mid 2(a_{i+1} - a_i) \xrightarrow[\text{10 não divide 2}]{} 10 \mid (a_{i+1} - a_i).$$

Concluímos que, tanto para i par como para i ímpar, se tem $S' - S = 2|a_{i+1} - a_i|$ e quando $|a_{i+1} - a_i| = 5$, $S' - S$ é um múltiplo de 10, o que significa que S' também é um múltiplo de 10, pelo que a transposição não é detetada. ■

A demonstração para o sistema EAN é análoga.

O exemplo 2.4.2. ilustra a proposição anterior:

Exemplo 2.4.2. Transposição de algarismos adjacentes

Sejam dois números UPC, com os seguintes números de identificação:

$(8,4,3,2,5,1,\mathbf{1},\mathbf{6},7,2,0,3) \rightarrow n.º\ correto$ e $(8,4,3,2,5,1,\mathbf{6},\mathbf{1},7,2,0,3) \rightarrow n.º\ errado$

Ocorreu uma transposição dos algarismos adjacentes a_7 e a_8

Assim, $S = 3 \times \underline{8} + \underline{4} + 3 \times \underline{3} + \underline{2} + 3 \times \underline{5} + \underline{1} + 3 \times \underset{a_7}{\underline{\mathbf{1}}} + \underset{a_8}{\underline{\mathbf{6}}} + 3 \times \underline{7} + \underline{2} + 3 \times \underline{0} + \underline{3}$

$S' = 3 \times \underline{8} + \underline{4} + 3 \times \underline{3} + \underline{2} + 3 \times \underline{5} + \underline{1} + 3 \times \underset{a_8}{\underline{\mathbf{6}}} + \underset{a_7}{\underline{\mathbf{1}}} + 3 \times \underline{7} + \underline{2} + 3 \times \underline{0} + \underline{3}$

$S' - S = 3 \times a_8 + a_7 - (3 \times a_7 + a_8)$

$= 3 \times \mathbf{6} + \mathbf{1} - (3 \times \mathbf{1} + \mathbf{6})$

$= 3 \times \mathbf{6} + \mathbf{1} - 3 \times \mathbf{1} - \mathbf{6}$

$= 3 \times \mathbf{6} - 3 \times \mathbf{1} - \mathbf{6} + \mathbf{1}$

$= 3 \times (\mathbf{6} - \mathbf{1}) - (\mathbf{6} - \mathbf{1})$

$= 2(\boldsymbol{a_8} - \boldsymbol{a_7})$

$= 2 \times (\mathbf{6} - \mathbf{1}) = 2 \times 5 = 10 \rightarrow$ múltiplo de 10,

pelo que neste caso $10 \mid (S' - S)$.

Note-se ainda que $S' - S = 2(a_{i+1} - a_i)$, sempre que i é ímpar (neste caso $i = 7$).

Ora, $S' - S$ é um número múltiplo de 10, mas $a_8 - a_7 = 5$, não o é, pelo que o sistema não deteta esta troca de algarismos.

Reparemos que neste exemplo, tanto S como S' são divisíveis por 10.

No caso de i ser par, tem-se $S' - S = 2(-\boldsymbol{a_{i+1}} + a_i)$, como se pode verificar através de um exemplo análogo ao anterior em que $i = 8$:

$(8,4,3,2,5,1,7,\mathbf{1},\mathbf{6},2,0,3) \rightarrow n.º\ correto$ e $(8,4,3,2,5,1,7,\mathbf{6},\mathbf{1},2,0,3) \rightarrow n.º\ errado$

Ocorreu uma transposição dos algarismos adjacentes a_8 e a_9

Com efeito,

$$S = 3\times\underline{8}+\underline{4}+3\times\underline{3}+\underline{2}+3\times\underline{5}+\underline{1}+3\times\underline{7}+\underbrace{\underline{1}}_{a_8}+3\times\underbrace{\underline{6}}_{a_9}+\underline{2}+3\times\underline{0}+\underline{3}$$

$$S' = 3\times\underline{8}+\underline{4}+3\times\underline{3}+\underline{2}+3\times\underline{5}+\underline{1}+3\times\underline{7}+\underbrace{\underline{6}}_{a_9}+3\times\underbrace{\underline{1}}_{a_8}+\underline{2}+3\times\underline{0}+\underline{3}$$

$$S' - S = a_9 + 3\times a_8 - (a_8 + 3\times a_9)$$

$$= \mathbf{6} + 3\times\mathbf{1} - (\mathbf{1} + 3\times\mathbf{6})$$

$$= \mathbf{6} + 3\times\mathbf{1} - \mathbf{1} - 3\times\mathbf{6}$$

$$= \mathbf{6} - \mathbf{1} + 3\times(\mathbf{1} - \mathbf{6})$$

$$= (\mathbf{6} - \mathbf{1}) - 3\times(\mathbf{6} - \mathbf{1})$$

$$= 2\times(-\boldsymbol{a_9} + \boldsymbol{a_8})$$

$$= 2\times(\mathbf{1} - \mathbf{6}) = 2\times(-5) = -10 \rightarrow \text{divisível por 10, pelo que } 10 \mid (S' - S).$$

Também neste caso $S' - S$ é múltiplo de 10, mas $a_9 - a_8 = 5$ não o é, pelo que o sistema não deteta esta troca de algarismos, pois tanto S como S' são divisíveis por 10.

Então, neste exemplo, tanto para i ímpar como para i par, tem-se que

$$|a_{i+1} - a_i| = 5.$$

Corolário 2.4.1.

No sistema UPC, a soma de teste S', de um número onde houve uma transposição de dois algarismos adjacentes, é um múltiplo de 10 se e só se $|a_{i+1} - a_i| = 5$.

Demonstração:

$$S' \text{múltiplo de } 10 \Leftrightarrow 10 \mid S' \Leftrightarrow 10 \mid (S' - S) \Leftrightarrow 10 \mid (2|a_{i+1} - a_i|) \Leftrightarrow |a_{i+1} - a_i| = 5,$$

pois $a_{i+1} - a_i$ é um número inteiro não nulo entre -9 e 9.

Podemos concluir que quando S' é múltiplo de 10, as trocas de algarismos adjacentes não são detetadas, ou seja, quando $|a_{i+1} - a_i| = 5$. ■

Reparemos que, das ${}^{10}A_2 = \frac{10!}{8!} = 90$ transposições possíveis de algarismos adjacentes o sistema apenas não deteta 10, que são os casos "05", "50", "16", "61", "27", "72", "38", "83", "49" e "94". Portanto o sistema apenas tem uma falibilidade de cerca de 11, 1%, ou seja, uma eficiência de cerca de 88,9%.

2.4.3. Sistema de identificação dos cheques bancários

Um cheque bancário tem três números legíveis por computador impressos na sua parte inferior. Um desses números refere-se ao número do cheque, outro ao número da conta e um terceiro número que identifica o banco emissor.

Em alguns bancos esse terceiro número é constituído por uma sequência de nove algarismos da seguinte forma:

$\underbrace{a_1\ a_2\ a_3\ a_4\ a_5\ a_6\ a_7\ a_8}_{\text{Identificação do banco}}\boxed{a_9}$ → Algarismo de teste ou de controlo

sendo que o algarismo de teste $a_9 \in \{0, 1, \ldots, 9\}$ e é calculado de modo que a soma de teste:

$$S = (7a_1 + 3a_2 + 9a_3 + 7a_4 + 3a_5 + 9a_6 + 7a_7 + 3a_8 - a_9) \equiv 0\ mod\ 10,$$

ou seja, que o resto da divisão de S por 10 seja 0, tal como no sistema UPC.

Assim, $a_9 \equiv (7a_1 + 3a_2 + 9a_3 + 7a_4 + 3a_5 + 9a_6 + 7a_7 + 3a_8)\ mod\ 10.$

O sistema dos cheques bancários deteta todos os erros singulares, mas não deteta todas as transposições de algarismos adjacentes, mais precisamente quando a diferença entre esses algarismos é, em valor absoluto, igual a 5.

Proposição 2.4.4.

O sistema dos cheques bancários deteta todos os erros singulares.

Demonstração:

Será demonstrado apenas o caso em que ocorre um erro singular num algarismo cujo coeficiente é 7 na soma de teste, uma vez que para a ocorrência do mesmo tipo de erro num algarismo cujo coeficientes seja $-1, 3$ ou 9, a prova é análoga.

1.º caso: ocorre um erro singular num algarismo de coeficiente 7

$S = (7a_1 + 3a_2 + 9a_3 + \cdots + 3a_8 - a_9) S' = (7\underbrace{a_1'}_{\text{Algarismo incorreto}} + 3a_2 + 9a_3 + \cdots + 3a_8 - a_9).$

Então, $S' - S = 7(a_i' - a_i)$, caso $i \in \{1, 4, 7\}$.

S é múltiplo de 10, isto é, $10 \mid S$.

Se S' fosse múltiplo de 10, então $10 \mid S'$ e portanto $10 \mid (S' - S)$, com

$$S' - S = 7(a_i' - a_i).$$

Assim,

$$7(a_i' - a_i) = 10 \times s' - 10 \times s,$$

com s' e s números inteiros pois por hipótese $10 \mid S'$ e $10 \mid S$.

Ora,

$$7(a_i' - a_i) = 10(s' - s) \Rightarrow 10 \mid 7(a_i' - a_i) \underbrace{\Longrightarrow}_{\text{10 não divide 7}} 10 \mid (a_i' - a_i),$$

o que é absurdo pois $a_i' - a_i$ é um número inteiro não nulo entre -9 e 9, uma vez que $a_i, a_i' \in \{0, 1, \ldots, 9\}$ e $a_i \neq a_i'$. Portanto S' não é múltiplo de 10. ■

Isto só acontece porque 10 não divide 7, pelo que $mdc(7,10) = 1$.

Também se verifica que, nos casos em que ocorre um erro singular em algarismos de coeficiente $-1, 3$ ou 9, se tem $mdc\,(3,10) = mdc\,(9,10) = mdc\,(-1,10) = 1$.

Proposição 2.4.5.

As transposições não detetadas, de algarismos adjacentes a_{i+1} e a_i, com $i \in \{1, \cdots, 8\}$, de um número de identificação bancária, são aquelas em que $|a_{i+1} - a_i| = 5$.

Demonstração:

Será analisado o caso em que ocorre uma transposição dos algarismos adjacentes a_{3i-2} e a_{3i-1} (a diferença dos coeficientes desses algarismos é 4), com $i \in \{1, 2, 3\}$ cuja diferença entre eles é, em módulo, igual a 5.

Nos casos em que ocorre uma transposição dos algarismos adjacentes em que a diferença dos coeficientes desses algarismos é, 2 ou -6, a demonstração é análoga.

Consideremos um número de identificação bancária correto e outro errado, onde ocorreu uma transposição dos algarismos adjacentes a_{3i-2} e a_{3i-1}:

$$a_1\, a_2 \cdots a_{3i-2}\, a_{3i-1} \cdots a_8\, a_9 \;\rightarrow n.º\ correto \text{ e}$$

$$a_1\, a_2 \cdots \underbrace{a_{3i-1}\, a_{3i-2}}_{\text{Ocorreu uma transposição dos algarismos adjacentes } a_{3i-2} \text{ e } a_{3i-1}} \cdots a_8\, a_9 \rightarrow n.º\ errado$$

Assim, $S = (\cdots + 7a_{3i-2} + 3a_{3i-1} + \cdots - a_9)$ e

$$S' = (\cdots + \underbrace{7a_{3i-1} + 3a_{3i-2}}_{\text{Transposição dos algarismos adjacentes } a_{3i-2} \text{ e } a_{3i-1}} + \cdots - a_9)$$

$$S' - S = 7a_{3i-1} + 3a_{3i-2} - (7a_{3i-2} + 3a_{3i-1}) = 4 \times (a_{3i-1} - a_{3i-2})$$

Quando $|a_{3i-1} - a_{3i-2}| = 5$, tem-se que $S' - S$ é múltiplo de 10, e portanto S' será também múltiplo de 10, pelo que o sistema não deteta esta troca de algarismos.

Esta situação ocorre também nos casos em que os algarismos adjacentes têm, respetivamente, coeficientes 7 e 4. Generealizando, pode-se dizer que o sistema não deteta a transposição de algarismos adjacentes sempre que $|a_{i+1} - a_i| = 5$ com $i = 1$, $i = 4$ ou $i = 7$.

Assim, nestes casos, se S', soma de teste de um número onde houve uma transposição de dois algarismos adjacentes, for múltiplo de 10, vem:

$$10 \mid S' \Leftrightarrow 10 \mid (S' - S) \Leftrightarrow 10 \mid 4|a_{i+1} - a_i| \Leftrightarrow |a_{i+1} - a_i| = 5,$$

pois $a_{i+1} - a_i$ é um número inteiro não nulo entre -9 e 9, uma vez que $a_i, a_{i+1} \in \{0, 1, \dots, 9\}$.

De modo análogo ao sistema UPC, concluímos, neste caso, que quando S' é múltiplo de 10, as trocas de algarismos adjacentes não são detetadas, ou seja, quando $|a_{i+1} - a_i| = 5$.

Podemos depreender retirar que, no sistema de identificação bancária, as trocas de algarismos adjacentes não são detetadas quando $|a_{i+1} - a_i| = 5$. ■

2.4.4. Sistema de identificação dos bilhetes de identidade

No caso dos bilhetes de identidade, o algoritmo de deteção de erros é idêntico ao do sistema ISBN. O número do bilhete de identidade (BI) é constituído por nove algarismos, sendo que o nono algarismo é o algarismo de teste:

$\underbrace{a_1\ a_2\ a_3\ a_4\ a_5\ a_6\ a_7\ a_8}\ \boxed{a_9}$

Identificação do BI — Algarismo de teste ou de controlo

$\boldsymbol{a_9}$ é escolhido de modo que a soma de teste

$$\boldsymbol{S = (9a_1 + 8a_2 + 7a_3 + \cdots + a_9) \equiv 0\ mod\ 11},$$

isto é, de tal modo que S seja divisível por 11. Assim,

$$a_9 = -(9a_1 + 8a_2 + 7a_3 + \cdots 2a_8) = -[(11-2)a_1 + (11-3)a_2 + \cdots + (11-9)a_8]$$

$$\Leftrightarrow a_9 = \left(-\sum_{i=1}^{8}[11 - (i+1)]a_i\right) mod\ 11$$

Se existir um erro singular, de transposição ou outro semelhante na transmissão do número, o resultado já não será divisível por 11. Analisemos alguns exemplos que ilustram esta situação.

Exemplo 2.4.3. Número de identificação do bilhete de identidade

Consideremos o número de identificação 8496712 9, constante na figura 2.4.1:

Figura 2.4.1. Número de identificação do bilhete de identidade

Neste caso, aparecem apenas oito algarismos. Quando isto acontece, e para testar a veracidade do sistema, é necessário acrescentar um número de zeros à esquerda de modo a que o BI perfaça os nove algarismos. Assim, o número de identificação acima deve ler-se 08496712 9, com 9 o algarismo de teste. Verifiquemos se é correto:

$$-(9\times 0+8\times 8+7\times 4+6\times 9+5\times 6+4\times 7+3\times 1+2\times 2)=-211$$

Ora $-211\equiv 9\ mod\ 11$, pelo que podemos concluir que não ocorreu erro na transmissão do número.

De facto,

$$S=9\times 0+8\times 8+7\times 4+6\times 9+5\times 6+4\times 7+3\times 1+2\times 2+9\equiv 0\ mod\ 11.$$

O problema surge quando o algarismo de teste é o 0, pelo que apresentaremos dois exemplos distintos em que isso acontece:

Exemplo 2.4.4. Número de identificação do BI com algarismo de teste 0

Seja o número de identificação 06991096 0, em que 0 é o algarismo de teste, como se pode verificar na figura 2.4.2.

Figura 2.4.2. Número de identificação de um BI com algarismo de teste 0

Esse algarismo é calculado da seguinte forma:

$$a_9=-(9\times 0+8\times 6+7\times 9+6\times 9+5\times 1+4\times 0+3\times 9+2\times 6)=-209.$$

Ora $-209\equiv 0\ mod\ 11$ e portanto podemos concluir que não ocorreu erro na transmissão do número. De facto,

$$S=9\times 0+8\times 6+7\times 9+6\times 9+5\times 1+4\times 0+3\times 9+2\times 6+0\equiv 0\ mod\ 11.$$

Exemplo 2.4.5. Número de BI com um falso 0 como algarismo de teste

Observemos o número de BI 06994704 0 constante na figura 2.4.3, também com o 0 como algarismo de teste:

Figura 2.4.3. Número de BI com falso 0 como algarismo de teste

Tal como no exemplo 2.4.4, o algarismo de teste é calculado da seguinte forma:

$$a_9 = -(9\times 0 + 8\times 6 + 7\times 9 + 6\times 9 + 5\times 4 + 4\times 7 + 3\times 0 + 2\times 4) = -221.$$

Mas $-221 \equiv 10\ mod\ 11$, ou seja $-221 \not\equiv 0\ mod\ 11$. A soma de teste S também não é divisível por 11 pois

$$S = 9\times 0 + 8\times 6 + 7\times 9 + 6\times 9 + 5\times 4 + 4\times 7 + 3\times 0 + 2\times 4 + 0 = 221$$

e $221 \not\equiv 0\ mod\ 11$. No entanto não se pode concluir quanto à ocorrência de algum erro na transmissão do número uma vez que $-221 \not\equiv 0\ mod\ 11$. Então este sistema de identificação tem um *bug*, uma vez que o algarismo de teste não controla a autenticidade do número. Contudo, se o último algarismo fosse substituído pelo valor 10, a soma de teste já seria divisível por 11. De facto,

$$S = 9\times 0 + 8\times 6 + 7\times 9 + 6\times 9 + 5\times 4 + 4\times 7 + 3\times 0 + 2\times 4 + \mathbf{10} = 231,$$

e $231 \equiv 0\ mod\ 11$. Como 10 não é um algarismo, para o sistema do bilhete de identidade ser implementado corretamente, bastaria usar, tal como no sistema ISBN, um caracter não numérico, que substituísse o valor 10, como por exemplo a letra **X**, uma vez que na numeração romana **X** representa o valor 10.

Para uma melhor compreensão desta situação faremos uma análise, de forma generalizada, do funcionamento dos sistemas modulares apresentados anteriormente.

2.5. Análise geral dos sistemas modulares

De um modo geral, todos os sistemas modulares têm características comuns, permitindo assim fazer uma caracterização sistemática, fixando:

- um conjunto A, designado *alfabeto,* com k elementos (de cardinal k);
- uma bijeção $\tau: A \longrightarrow \mathbb{Z}_k$;
- um *n-upio* $(p_1, p_2, \cdots, p_n)$, *vetor de verificação de algarismos* do sistema, de *pesos* inteiros não nulos;

- sequências finitas $a_1 a_2 \cdots a_n$, de elementos do *alfabeto*, de comprimento *n*, designadas por *palavras*, definidas no alfabeto A , com $\tau(a_i) \in \mathbb{Z}_k$, com $i = 1, \cdots, n$

As *palavras* são definidas no alfabeto A, de tal modo que:

$$(p_1, p_2, \cdots, p_n) \cdot \big(\tau(a_1), \tau(a_2), \cdots, \tau(a_n)\big) \equiv 0 \; mod \; k \Leftrightarrow$$

$$p_1 \tau(a_1) + p_2 \tau(a_2) + \cdots + p_n \tau(a_n) \equiv 0 \; mod \; k,$$

generalizando assim os **sistemas de identificação *módulo k*** apresentados anteriormente. A partir desta formalização, é fácil identificar os sistemas abordados:

Sistema ISBN (International Standard Book Number)

- $A = \{0, 1, 2, 3, \cdots, 9, X\}$ $\boldsymbol{k = 11}$
- $\tau: \{0, 1, 2, 3, \cdots, 9, X\} \longrightarrow \mathbb{Z}_{11}$, com $\tau(0) = 0, \tau(1) = 1, \cdots, \tau(9) = 9, \tau(X) = 10$
- $(p_1, p_2, \cdots, p_{10}) = (10, 9, 8, 7, 6, 5, 4, 3, 2, 1)$

Sistema UPC (Universal Product Code)

- $A = \{0, 1, 2, 3, \cdots, 9\}$ $\boldsymbol{k = 10}$
- $\tau: \{0, 1, 2, 3, \cdots, 9\} \longrightarrow \mathbb{Z}_{10}$, $\tau(a) = a, \; \forall a \in A$
- $(p_1, p_2, \cdots, p_{12}) = (3, 1, 3, 1, 3, 1, 3, 1, 3, 1, 3, 1)$

Sistema EAN (European Article Number)

- $A = \{0, 1, 2, 3, \cdots, 9\}$ $\boldsymbol{k = 10}$
- $\tau: \{0, 1, 2, 3, \cdots, 9\} \longrightarrow \mathbb{Z}_{10}$, $\tau(a) = a, \; \forall a \in A$
- $(p_1, p_2, \cdots, p_{13}) = (1, 3, 1, 3, 1, 3, 1, 3, 1, 3, 1, 3, 1)$

Sistema dos cheques bancários

- $A = \{0, 1, 2, 3, \cdots, 9\}$ $\boldsymbol{k = 10}$
- $\tau: \{0, 1, 2, 3, \cdots, 9\} \longrightarrow \mathbb{Z}_{10}$, $\tau(a) = a, \; \forall a \in A$
- $(p_1, p_2, \cdots, p_9) = (7, 3, 9, 7, 3, 9, 7, 3, -1)$

Sistema dos bilhetes de identidade (corrigido)

- $A = \{0, 1, 2, 3, \cdots, 9, \boldsymbol{X}\}$ $\boldsymbol{k = 11}$
- $\tau: \{0, 1, 2, 3, \cdots, 9, \boldsymbol{X}\} \longrightarrow \mathbb{Z}_{11}$, com $\tau(0) = 0, \tau(1) = 1, \cdots, \tau(9) = 9, \tau(\boldsymbol{X}) = 10$
- $(p_1, p_2, \cdots, p_9) = (9, 8, 7, 6, 5, 4, 3, 2, 1)$

Fixados tais elementos no processo de sistematização dos sistemas modulares e identificando $\tau(a)$ com a, para não sobrecarregar a notação, podemos dizer que,

em geral, nos sistemas *módulo k*, o algarismo de controlo, a_n, de um determinado número $a_1 a_2 \cdots a_n$ é calculado resolvendo a equação:

$$p_1 a_1 + p_2 a_2 + \cdots + p_n \boldsymbol{a_n} \equiv 0 \; mod \; k, \text{ com } \boldsymbol{a_n} \equiv (p_1 a_1 + p_2 a_2 + \cdots + p_{n-1} a_{n-1}) \; mod \; k,$$

uma vez que, $p_n \in \{-1, 1\}$, nos sistemas abordados anteriormente.

É então possível determinar quais os erros singulares e quias as transposições que são detetáveis, conforme demonstrado na proposição 2.5.1:

Proposição 2.5.1.

Seja $a_1 a_2 \cdots a_n$ um número de um sistema de identificação *módulo k*, com *pesos* $p_1 \; p_2 \; , \cdots , p_n$.

(a) Um erro singular $a_i \rightarrow a_i'$ na $i - ésima$ posição é detetável **se e só se**

$$p_i(a_i' - a_i) \not\equiv 0 \; mod \; k$$

(b) Uma transposição dos algarismos a_i e a_j nas posições i e j é detetável **se e só se**

$$(p_i - p_j)(a_j - a_i) \not\equiv 0 \; mod \; k$$

Demonstração:

(a) Sendo S, a soma de teste do número correto e S' a soma de teste do número incorreto, tem-se que

$$S' - S = p_i(a_i' - a_i)$$

Como $S \equiv 0 \; mod \; k$, o erro só se pode detetar quando $S' \not\equiv 0$, ou seja o erro é detetável se e só se $p_i(a_i' - a_i) \not\equiv 0 \; mod \; k$. ■

(b) Numa transposição dos algarismos a_i e a_j nas posições i e j, sendo S a soma de teste do número correto e S' a soma de teste do número incorreto, tem-se que:

$$\begin{aligned} S' - S &= p_i a_j - p_i a_i + p_j a_i - p_j a_j \\ &= p_i(a_j - a_i) + p_j(a_i - a_j) \\ &= p_i(a_j - a_i) - p_j(a_j - a_i) \\ &= (p_i - p_j)(a_j - a_i). \end{aligned}$$

Como $S \equiv 0 \; (mod \; k)$, a transposição de algarismos só é detetável quando $S' \not\equiv 0$, ou seja se e só se $(p_i - p_j)(a_j - a_i) \not\equiv 0 \; mod \; k$. ■

Corolário 2.5.1.

Um sistema de identificação *módulo k*, com pesos $p_1, p_2, \cdots, p_n$, deteta:

(a) todos os erros singulares na posição i **se e só se** $mdc(p_i, k) = 1$.

(b) todas as transposições de algarismos nas posições i e j **se e só se**

$$mdc(p_i - p_j, k) = 1.$$

Demonstração:

(a) Pela proposição 2.5.1, sabemos que o sistema deteta todos os erros singulares na $i-ésima$ posição **se e só se** $p_i(a_i' - a_i) \not\equiv 0\ (mod\ k)$ para quaisquer $a_i, a_i' \in \{0, 1, \cdots, k-1\}$, com $a_i \neq a_i'$.

Consideremos, por hipótese, que o sistema deteta todos os erros singulares na posição i, logo $p_i(a_i' - a_i) \not\equiv 0\ (mod\ k)$. Se, por absurdo, $mdc(p_i, k) = d > 1$, viria $p_i = dd_1$ e $k = dd_2$, com $d_2 \in \{1, \cdots, k-1\}$. Fazendo, por exemplo $a_i = 0$ e $a_i' = d_2$, obter-se-ia $p_i(a_i' - a_i) = dd_1(d_2 - 0) = dd_1\left(\frac{k}{d}\right) = kd_1 \equiv 0\ (mod\ k)$, o que é uma contradição, pois $p_i(a_i' - a_i) \not\equiv 0\ mod\ k$.

Reciprocamente, se existissem $a_i, a_i' \in \{0, 1, \cdots, k-1\}$, com $a_i \neq a_i'$ tais que $p_i(a_i' - a_i) \equiv 0\ (mod\ k)$, isto é, tais que $p_i(a_i' - a_i)$ fosse múltiplo de k, obter-se-ia, uma vez que por hipótese $mdc(p_i, K) = 1$, que $(a_i' - a_i)$ seria múltiplo de k, o que é uma contradição pois $a_i' - a_i \in \{0, 1, \cdots, k-1\}$. ■

(b) Pela proposição 2.5.1, sabemos que o sistema deteta todas as transposições de algarismos nas posições i e j **se e só se** $(p_i - p_j)(a_j - a_i) \not\equiv 0\ (mod\ k)$, para quaisquer $a_i, a_i' \in \{0, 1, \cdots, k-1\}$, com $a_i \neq a_i'$.

Consideremos, por hipótese, que o sistema deteta todas as transposições de algarismos nas posições i e j, portanto $(p_i - p_j)(a_j - a_i) \not\equiv 0\ (mod\ k)$. Se, por absurdo, $mdc(p_i - p_j, K) = d > 1$, viria $p_i - p_j = dd_1$ e $k = dd_2$, com $d_2 \in \{1, \cdots, k-1\}$. Fazendo, por exemplo $a_i = 0$ e $a_j = d_2$ obter-se-ia

$$(p_i - p_j)(a_j - a_i) = dd_1(d_2 - 0) = dd_1\left(\frac{k}{d}\right) = kd_1 \equiv 0\ (mod\ k),$$

o que é uma contradição, pois $(p_i - p_j)(a_j - a_i) \not\equiv 0\ mod\ k$.

Reciprocamente, se existissem a_i e $a_j \in \{1, \cdots, k-1\}$ tais que $(p_i - p_j)(a_j - a_i)$ fosse múltiplo de k, $(a_j - a_i)$ teria de ser múltiplo de k, o que é uma contradição pois $a_j - a_i \in \{0, 1, \cdots, k-1\}$. ■

O sistema de identificação do cartão de cidadão supera este *bug*, como verificaremos mais à frente.

2.6. Limitações dos sistemas modulares

Da análise feita dos sistemas de identificação modulares abordados, podemos depreender que um sistema *módulo k* é mais eficiente se $k > 10$, com k um número primo. Por exemplo, o sistema ISBN é um sistema *módulo 11* que deteta a 100% todos os erros singulares e todas as transposições. No entanto, a sua eficiência não é total pois não deteta todos os erros fonéticos [1]. Existem outros sistemas *módulo 11* mais eficientes, onde o vetor de verificação dos algarismos é diferente, ou que utilizam dois algarismos de teste. Ainda assim, esses sistemas não detetam alguns dos erros aleatórios mencionados na secção 2.2. Também são utlizados alguns sistemas bastante eficientes *módulo k*, com $k > 11$ e (k um número primo), mas que, tal como o ISBN, utilizam caracteres não numéricos para dígitos de controlo.

Embora a eficiência dos sistemas *módulo k*, com $k > 10$ seja superior, o problema destes sistemas reside no facto de ser necessário utilizar mais do que um algarismo de teste ou, em alternativa, utilizar caracteres não numéricos para dígitos de controlo, o que acarreta alguns problemas técnicos, além de se correr o risco daqueles caracteres poderem ser substituídos por algarismos que alguém, "na sua reconfortante ignorância matemática sobre códigos" [3], mas com poder de decisão, considere mais convenientes, como aconteceu no caso do BI.

2.7. Sistemas de identificação modulares melhorados

Esta desvantagem dos sistemas *módulo k*, com $k > 10$ deu o mote para tentar conceber um sistema idêntico ao sistema *módulo 10*, com algumas alterações, generalizado, que detetasse todos os erros singulares e todas as transposições, que são os erros que ocorrem com maior frequência.

Recordemos que, como foi enunciado no corolário 2.5.1, para um sistema de identificação *módulo k* detetar todos os erros singulares os pesos p_i, $i = 1, \cdots, n$ devem ser escolhidos de forma a p_i e k serem primos entre si, pelo que $0, p_i \times 1, p_i \times 2, \cdots, p_i \times (k-1)$ constitui uma permutação dos elementos de $\mathbb{Z}_k$. Assim, se em vez de se multiplicar o elemento a_i do alfabeto *A*, pelo peso p_i, se aplicar uma determinada função peso, ψ_i, a cada um dos algarismos do número considerado, é possível alterar o sistema substituindo a função $x \mapsto p_i x$ por uma função arbitrária $x \mapsto \psi_i(x)$, de $\mathbb{Z}_k$ em $\mathbb{Z}_k$, apresentando uma versão melhorada dos sistemas de identificação *módulo k*:

Definição 2.7.1. Sistemas de identificação *módulo k* melhorados

Sejam A um conjunto, designado *alfabeto,* de cardinal k, $\tau: A \longrightarrow \mathbb{Z}_k$ uma bijeção e $(\psi_1, \psi_2, \cdots, \psi_n)$ um *n-uplo*, de aplicações de $\mathbb{Z}_k$ em $\mathbb{Z}_k$. Um sistema de identificação *módulo k* é constituído pelas *palavras* $a_1 a_2 \cdots a_n$, definidas no alfabeto A, com $\tau(a_i) \in \mathbb{Z}_k$ e $i = 1, \cdots, n$, de modo a satisfazerem a seguinte condição de teste:

$$(\psi_1, \psi_2, \cdots, \psi_n) \cdot (\tau(a_1), \tau(a_2), \cdots, \tau(a_n)) \equiv 0 \; mod \; k \Leftrightarrow$$

$$\psi_1(\tau(a_1)) + \psi_2(\tau(a_2)) + \cdots + \psi_n(\tau(a_n)) \equiv 0 \; mod \; k.$$

Cometendo o abuso de identificar $\tau(a)$ com a, a condição de teste será:

$\psi_1(a_1) + \psi_2(a_2) + \cdots + \psi_n(a_n) \equiv 0 \; mod \; k$, com o algarismo de teste:

$$a_n = \psi_n^{-1}\left(-\sum_{i=1}^{n-1} \psi_i(a_i)\right) mod \; k$$

O n-uplo $(\psi_1, \psi_2, \cdots, \psi_n)$ passa a ser o vetor de verificação de algarismos do sistema, generalizando os sistemas de identificação modulares.

2.7.1. Sistema de identificação do cartão de cidadão

O cartão de cidadão (CC) é o cartão de identificação de um indivíduo que substitui não só o bilhete de identidade (BI), como também o cartão de contribuinte, o cartão de identificação da segurança social, o cartão de utente dos serviços de saúde e o cartão de eleitor. Além de outras características, o CC de cada indivíduo é constituído por um número que inclui o antigo número do BI, que passou a chamar-se número de identificação civil, o dígito de controlo desse número de identificação civil, dois caracteres não numéricos e ainda um dígito de controlo de todo o número de documento, como ilustra a figura 2.7.1:

Figura 2.7.1. Número de documento do cartão de cidadão

Os caracteres não numéricos representam o número da emissão do cartão para um determinado cidadão, isto é, o primeiro cartão emitido apresenta as letras ZZ. Se for emitido um novo cartão, já apresentará as letras ZY e assim

sucessivamente. A cada letra é atribuído um valor numérico conforme consta na tabela 2.7.1.

Tabela 2.7.1. Valor numérico correspondente a cada letra

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	20	31	32	33	34	35

O dígito de controlo do número de documento é um algarismo entre 0 e 9, que permite detetar algum erro que possa ter ocorrido na escrita de todo o número de documento, colmatando o *bug* verificado no sistema do BI, em que o algarismo de teste 0 também representava o número 10. Relativamente a este sistema, o sistema de identificação do cartão de cidadão é um sistema de identificação modular melhorado, em que o seu número de documento é traduzido por doze caracteres alfanuméricos, sendo o último, o dígito de controlo:

$a_1\ a_2\ a_3\ a_4\ a_5\ a_6\ a_7\ a_8\ a_9\ a_{10}\ a_{11}\ \boxed{a_{12}}$ → Dígito de teste ou de controlo

n.º de documento

com a_{12} escolhido de modo que a soma de teste

$$S = (\psi(a_1) + \psi(a_2) + \cdots + \psi(a_{11}) + \psi(a_{12})) \equiv 0\ mod\ 10,$$

Isto é, a_{12} é escolhido de modo a que S seja divisível por 10, com $a_{10}, a_{11} \in \{10, \cdots, 35\}$, conforme a letra a que correspondam na tabela 2.7.1. e com

$$\psi(a_i) = \begin{cases} 2a_i & \text{se } 2a_i < 10 \text{ e } i \text{ ímpar} \\ 2a_i - 9 & \text{se } 2a_i \geq 10, \text{para } i \text{ ímpar} \\ a_i & \text{para } i \text{ par} \end{cases}$$

O dígito de controlo será então:

$$a_{12} = -(\psi(a_1) + a_2 + \psi(a_3) + a_4 + \psi(a_5) + a_6 + \psi(a_7) + a_8 + \psi(a_9) + a_{10} + \psi(a_{11}))\ mod\ 10$$

Se existir um erro singular, de transposição ou outro semelhante na transmissão do número, o resultado já não será divisível por 10. O exemplo 2.7.1 ilustra a veracidade desta afirmação.

Exemplo 2.7.1. Verificação do algarismo de controlo do número do CC

Consideremos o número de documento constante na figura 2.7.1:

08496712 9ZZ6,

com 6 o algarismo de teste. Verifiquemos se é correto:

$$-(0 + 8 + 8 + 9 + 3 + 7 + 2 + 2 + 9 + 35 + 61) = -144$$

Ora $-144 \equiv 6\, mod\, 10$, pelo que concluímos que não ocorreu erro na transmissão do número. De facto,

$$S = 6 + 61 + 35 + 9 + 2 + 2 + 7 + 3 + 9 + 8 + 8 + 0 \equiv 0\, mod\, 10.$$

O problema surgia no BI quando o dígito de controlo 0 representava o número 10. No caso do número de documento do cartão de cidadão, esse problema é superado como veremos no exemplo 2.7.2.

Exemplo 2.7.2. Determinação do dígito de controlo do número do CC

Consideremos o número de identificação do bilhete de identidade apresentado na figura 2.4.3, cujo algarismo de teste é um falso zero: $06994704\ 0$.

Como já analisámos anteriormente, o dígito de controlo deveria representar o número 10 e não o número 0. Quando o referido bilhete de identidade perder a validade, o portador do mesmo irá obter um cartão de cidadão em que o seu número de documento será da forma $06994704\ 0ZZC$, com Z a tomar o valor constante na tabela 2.7.1. e C será o dígito de controlo escolhido de modo a que a soma de teste seja um múltiplo de 10, colmatando a falha verificada no BI. Assim:

$$C = -(0 + 6 + 9 + 9 + 8 + 7 + 0 + 4 + 0 + 35 + 61)\, mod\, 10 = -139\, mod\, 10.$$

Como $-139 \equiv 1\, mod\, 10$, um possível número de documento para o cartão de identificação daquele cidadão será $06991096\ 0ZZ\mathbf{1}$, com 1 o dígito de controlo do número de documento. De facto,

$$S = 0 + 6 + 9 + 9 + 8 + 7 + 0 + 4 + 0 + 35 + 61 + \mathbf{1} = 140$$

e $140 \equiv 0\, mod\, 10$, pelo que o algarismo de teste do número do documento está bem calculado.

Ainda assim, os sistemas modulares melhorados não são 100% eficientes na deteção de todos os erros. Nenhum sistema de identificação *módulo k*, verifica, simultaneamente, as seguintes condições:

- detetar todos os erros singulares e todas as transposições de algarismos adjacentes;
- usar somente os algarismos 0, 1, 2, ..., 9 para dígitos de controlo, sem a necessidade de símbolos extra, e que trabalhem somente com um algarismo de teste.

A solução passa por generalizar a definição 2.7.1, substituindo a estrutura do grupo $\mathbb{Z}_k$ por outro grupo arbitrário G, conforme referimos na secção 2.8.

2.8. Sistemas baseados na teoria de grupos

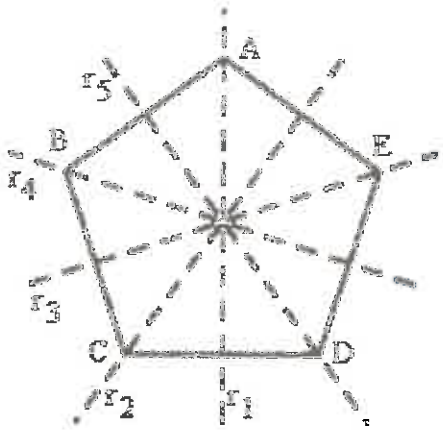
Este tipo de sistemas de identificação é baseado na teoria de grupos e são definidos sobre um grupo com uma estrutura suficientemente rica, que permite detetar a 100% todos os erros singulares e todas as transposições adjacentes, mas sem as limitações dos sistemas modulares. Um desses sistemas é o que se obtém utilizando um grupo não-abeliano de ordem 10. Foi o matemático holandês J. Verhoeff que, em 1969, encontrou esse sistema eficiente ([1] p. 64), baseado no grupo diedral D_5, das simetrias de um polígono regular com cinco lados. Esta estrutura envolve conceitos mais complexos que os utilizados na aritmética modular, mas é 100% eficiente na deteção de todos os erros singulares e transposições adjacentes, sem a necessidade de utilizar mais do que um algarismo de teste ou introduzir um símbolo não numérico para dígito de controlo.

O sistema de controlo baseado nas ideias de Verhoeff e na teoria de grupos está ilustrado no exemplo 2.8.1:

Exemplo 2.8.1. Sistema de Verhoeff

Consideremos o pentágono regular representado na figura 2.8.1.

Figura 2.8.1. Simetrias de um pentágono regular



O conjunto das simetrias de um pentágono regular, designado por grupo diedral D_5, é constituído por 5 rotações α_i, para $i \in \{1,2,3,4,5\}$, com α_i a rotação de ângulo $\frac{i-1}{5} \times 2\pi$, e 5 reflexões ρ_i, para $i \in \{1,2,3,4,5\}$, com ρ_i a reflexão sobre o eixo r_i, para $i \in \{1,2,3,4,5\}$.

Cada uma dessas simetrias é numerada conforme a tabela 2.8.1:

Tabela 2.8.1. Simetrias do grupo diedral D_5

0	Identidade: Rotação de ângulo **0**
1	Rotação de ângulo $\frac{1}{5} + 2\pi$
2	Rotação de ângulo $\frac{2}{5} + 2\pi$
3	Rotação de ângulo $\frac{3}{5} + 2\pi$
4	Rotação de ângulo $\frac{4}{5} + 2\pi$
5	Reflexão em relação à reta r_1
6	Reflexão em relação à reta r_2
7	Reflexão em relação à reta r_3
8	Reflexão em relação à reta r_4
9	Reflexão em relação à reta r_5

A operação de D_5 é então definida pela tabela 2.8.2:

Tabela 2.8.2. Operação do grupo diedral D_5

$\star$	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
0	0	1	2	3	4	5	6	7	8	9
1	1	2	3	4	0	6	7	8	9	5
2	2	3	4	0	1	7	8	9	5	6
3	3	4	0	1	2	8	9	5	6	7
4	4	0	1	2	3	9	5	6	7	8
5	5	9	8	7	6	0	4	3	2	1
6	6	5	9	8	7	1	0	4	3	2
7	7	6	5	9	8	2	1	0	4	3
8	8	7	6	5	9	3	2	1	0	4
9	9	8	7	6	5	4	3	2	1	0

Este sistema deteta todos os erros singulares e deteta todas as transposições de dois algarismos adjacentes, pelo que é um sistema de controlo, muito mais eficiente que os modulares. Estranhamente, não é praticamente utilizado. Um dos exemplos da aplicação do sistema de Verhoeff era o sistema implementado nas notas de marco alemãs. No entanto, com a adoção da moeda única, esse sistema deixou de ser usado, em detrimento de um sistema muito menos eficiente. Esse sistema, implementado nas notas do euro, é também baseado na aritmética modular (Anexo 2).

O sistema de Verhoeff é um ótimo sistema na deteção de erros, mas não faz a sua correção automática. Existem outros sistemas eficientes que detetam os erros e fazem a sua correção automática. Contudo, não são muito utilizados pois na maior parte das situações a correção não é permitida, como é o caso dos bancos.

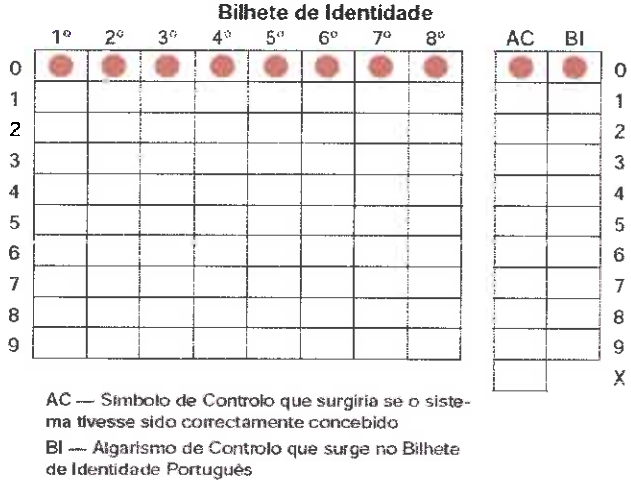
2.9. Aplicações dos sistemas modulares e de Verhoeff

Podemos verificar experimentalmente, em www.atractor.pt/mat/alg_controlo/, quais os erros que alguns dos sistemas modulares ou o sistema de Verhoeff detetam, além de podermos confirmar se o dígito de controlo, por exemplo, de um bilhete de identidade, está ou não correto. Podemos também confirmar se o número de documento de um cartão de cidadão está ou não corretamente escrito através de um algoritmo concebido no programa Octave.

Exemplo 2.9.1. Correção experimental do sistema de identificação do BI

Em www.atractor.pt/mat/alg_controlo/, além de ser possível testar se um número de BI está ou não bem representado, também podemos verificar que, atribuindo ao dígito de controlo o símbolo X para substituir o valor 10, o número de identificação já estaria correto. Essa verificação pode ser feita através de uma tabela dinâmica como a ilustrada na figura 2.9.1:

Figura 2.9.1. Verificação da autenticidade do dígito de controlo do BI



Relativamente ao cartão do cidadão, uma vez que é um documento recente, não é ainda vulgar encontrar algoritmos que testem o número do documento. Por este facto construímos um algoritmo, desenvolvido nos projetos educacionais I e II, que verifica se o número do cartão de cidadão está corretamente escrito.

Algoritmo 2.9.1. Algoritmo de teste do número de documento do CC

```
num1=input('Escreva os algarismos do seu CC, antes das letras:      ');
num2=input('Escreva o algarismo do seu CC, depois das letras:      ');
letra2=input('Insira o valor da segunda letra seguindo a chave: Z=35, Y=34, X=33,');
aux1=num1/10^8; aux2=floor(num1/10^8);
a1=floor(10*(aux1-aux2));
```

```
aux1=num1/10^6; aux2=floor(num1/10^6);
a2=floor(10*(aux1-aux2));
aux1=num1/10^4; aux2=floor(num1/10^4);
a3=floor(10*(aux1-aux2));
aux1=num1/10^2; aux2=floor(num1/10^2);
a4=floor(10*(aux1-aux2));
a5=35;
a6=num2;

s1=a1+a2+a3+a4+a5+a6;

b1=floor(10*(num1/10^9));
aux1=num1/10^7; aux2=floor(num1/10^7);
b2=floor(10*(aux1-aux2));
aux1=num1/10^5; aux2=floor(num1/10^5);
b3=floor(10*(aux1-aux2));
aux1=num1/10^3; aux2=floor(num1/10^3);
b4=floor(10*(aux1-aux2));
aux1=num1/10; aux2=floor(num1/10);
b5=floor(10*(aux1-aux2));
b6=letra2;

b=[b1 b2 b3 b4 b5 b6];
b=2*b;
for i=1:6
  if b(i)>=10
   b(i)=b(i)-9;
 end
end

s2=sum(b);
s=s1+s2;

controlo=s/10-floor(s/10);
if controlo==0
disp('INTRODUZIU CORRETAMENTE O SEU ID ' )
else disp('INTRODUZIU O SEU ID INCORRETAMENTE')
end
```

3 Projeto educacional II

No projeto educacional II, são abordadas outras aplicações da aritmética modular, como a aritmética do relógio, os calendários e a criptografia. São também ilustrados alguns exemplos da vida quotidiana.

É ainda feita uma breve caracterização da escola e da turma onde o tema foi implementado e uma descrição sucinta da apresentação do tema na sala de aula, bem como uma análise à recetividade ao tema por parte dos alunos.

3.1. Exemplos e aplicações

Nesta secção apresentaremos alguns exemplos elementares e outras aplicações da aritmética modular cuja escolha foi motivada para apresentação em sala de aula.

3.1.1. Aritmética do relógio

A aritmética do relógio é um exemplo de congruência *módulo 12.* De facto, o tempo é observado entre os múltiplos de 12, como ilustra o exemplo 3.1.1:

Exemplo 3.1.1. Números equivalentes a 2 em $\mathbb{Z}_{12}$

Se num determinado momento, um relógio analógico comum, marcar 12:00 horas, passadas 2 horas marcará 2:00 horas pois 2 é o resto da divisão inteira de 14 por 12, isto é $14 \equiv 2\ mod\ 12$. Na aritmética usual, $12 + 2 = 14$, mas na aritmética do relógio $12 + 2 = 2$. Por outro lado, 34 horas antes o relógio terá marcado igualmente 2:00 horas, pois 2 é o resto da divisão inteira de -22 por 12, ou seja $-22 \equiv 2\ mod\ 12$. Concluímos que na aritmética do relógio, os números 14 e -34 são equivalentes ao número 2.

O conjunto de todos os números equivalentes ao número 2 será $\{2 + 12k, k \in \mathbb{Z}\} = \{\cdots, -34, -22, -10, 2, 14, 26, 38, \cdots\}$, constituído por todos os números cuja divisão por 12 tem resto 2.

3.1.2. Calendários

Um exemplo de congruência *módulo 7* está relacionado com os calendários. Cada dia da semana é repetido após 7 dias e essa periodicidade faz com que o número correspondente a cada dia aumente de sete em sete. Assim, cada dia da semana é representado a partir dos múltiplos de 7. No exemplo 3.1.2. é apresentada uma aplicação prática da aritmética modular relativamente aos calendários.

Exemplo 3.1.2. Dia da semana do 25 de Abril de 2016

Cada dia da semana é representado a partir dos múltiplos de 7 e o primeiro dia do mês de Abril de 2013 calhou a uma 2.ª feira, conforme se pode verificar na figura 3.1.1:

Figura 3.1.1. Calendário de Abril de 2013

DOM	SEG	TER	QUA	QUI	SEX	SAB
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30				

Dom	Seg	···	Sab
M_7	$M_7 + 1$	···	$M_7 + 6$
n.ºs que divididos por 7 dão resto **0**	n.ºs que divididos por 7 dão resto **1**	···	n.ºs que divididos por 7 dão resto **6**

Para descobrir em que dia da semana será o 25 e Abril de 2016, podemos, inicialmente, fazer corresponder cada dia da semana a um número de 0 a 6, conforme a tabela 3.1.1:

Tabela 3.1.1. Correspondência do dia da semana

Domingo	2.ª feira	3.ª feira	4.ª feira	5.ª feira	6.ª feira	Sábado
0	1	2	3	4	5	6

De seguida devemos contabilizar quantos dias faltam para o 25 de Abril de 2016, partindo do dia 1 de Abril de 2013, não esquecendo os anos bissextos:

	Ano				
Mês	2013	2014	2015	2016 (bissexto)	
Janeiro		31	31	31	
Fevereiro		28	28	29	
Março		31	31	31	
Abril	30	30	30	25	
Maio	31	31	31		
Junho	30	30	30		
Julho	31	31	31		
Agosto	31	31	31		
Setembro	30	30	30		
Outubro	31	31	31		
Novembro	30	30	30		
Dezembro	31	31	31		
Total	275	365	365	116	**1121**

Fazendo a divisão inteira de 1121 por 7, obtém-se resto **1,** ou seja, $1121 \equiv 1 \ mod \ 7$, e portanto, passados 1121 dias, o 25 de Abril de 2016 irá ser no dia da semana correspondente ao número 1, na tabela 3.1.1, ou seja 2.ª feira.

Exemplo 3.1.3. Em que dia nasceu?

Como cada semana tem 7 dias, a tabela dos dias da semana é construída com base na congruência *módulo* 7, conforme a tabela 3.1.1, apresentada no exemplo 3.1.2.

A associação de cada mês a um número também é feita com base na congruência *módulo* 7, do seguinte modo: janeiro é o mês de referência, uma vez que é o primeiro mês do ano, pelo que associamos a esse mês o número 1. Como janeiro tem 31 dias e $31 \equiv 3 \, mod \, 7$, associa-se o número 4 ao mês de fevereiro pois 1+3=4. Janeiro "empurra" o início do mês de fevereiro três posições.

Fevereiro tem 28 dias e $28 \equiv 0 \, mod \, 7$, pelo que fevereiro não "empurra" o início do mês de março. Assim, a março associa-se igualmente o número 4, pois começa exatamente no mesmo dia da semana que o mês de Fevereiro (se o ano não for bissexto).

Março tem 31 dias e $31 \equiv 3 \, mod \, 7$, pelo que março "empurra" três posições o início do mês de Abril e portanto a abril associa-se o número 0 (pois 4+3=7$\equiv 0 \, mod \, 7$).

Abril tem 30 dias e $30 \equiv 2 \, mod \, 7$, pelo que abril "empurra" o início do mês de maio 2 posições e portanto a maio associa-se o número 2 $(0 + 2 = 2)$.

Maio tem 31 dias e $31 \equiv 3 \, mod \, 7$, pelo que maio "empurra" 3 posições o início do mês de junho e portanto a junho associa-se o número 5 (2+3=5).

Junho tem 30 dias e $30 \equiv 2 \, mod \, 7$, pelo que junho "empurra" 2 posições o início do mês de julho e portanto a julho associa-se o número 0 (5+2=7$\equiv 0 \, mod \, 7$). Seguindo este raciocínio, podemos fazer corresponder cada mês de um ano, a um algarismo compreendido entre 0 e 6, conforme a tabela 3.1.2.

Tabela 3.1.2. Correspondência do mês do ano

Tabela dos meses											
Janeiro	1	Fevereiro	4	Março	4	Abril	0	Maio	2	Junho	5
Julho	0	Agosto	3	Setembro	6	Outubro	1	Novembro	4	Dezembro	6

Atendendo a que 1900 foi um ano bissexto e porque o 1.º dia de Janeiro de 1900 foi uma 2.ª feira, através do procedimento seguinte poderemos determinar o dia de aniversário.

Comecemos por calcular quantos anos passaram desde o ano 1900 até ao ano de nascimento do indivíduo, pois é necessário determinar a translação, na sequência dos sete dias da semana (2.ª, 3.ª, 4.ª, 5.ª, 6.ª, sábado, domingo), relativamente ao primeiro dia de janeiro de 1900. Como $365 \equiv 1 \, mod \, 7$, cada ano que passa aumenta uma posição para a direita no ciclo dos sete dias da semana, relativamente ao primeiro dia de janeiro. Se o ano for bissexto, aumenta duas posições para a direita, em relação ao primeiro dia de janeiro, uma vez que $366 \equiv 2 \, mod \, 7$. Por cada ano bissexto contabilizado desde 01/01/1900 até ao ano de nascimento da pessoa, aumenta-se mais uma posição à translação determinada anteriormente. Para calcular quantos anos bissextos existiram depois de 1900 faz-

se a divisão inteira da diferença entre 1900 e o ano de nascimento por 4 e considera-se o quociente dessa divisão.

Estes cálculos serviram apenas para localizar o dia da semana do primeiro dia de janeiro do ano em que a pessoa nasceu. A partir daí é necessário encontrar o deslocamento feito desde o 1.º de janeiro desse ano até ao dia e mês do aniversário do indivíduo. Recorrendo a uma tabela semelhante à tabela 3.1.2, localiza-se o mês do aniversário. Localizado o primeiro dia desse mês, considerando x o dia de aniversário, o mesmo encontra-se $x-1$ posições após o primeiro dia do referido mês.

3.1.3. Criptografia

A criptografia é outro exemplo onde se verifica a utilização da aritmética modular. A criptografia (do grego *kryptós*, "escondido", e *gráphein*, "escrita") é uma área da criptologia (ciência dos códigos), que se dedica ao estudo das mensagens secretas e dos processos de encriptação [2]. Nesses processos a informação original é transformada numa outra impercetível, de modo a que apenas possa ser conhecida pelo seu destinatário, detentor da chave secreta. A criptografia clássica, primeira aplicação da criptografia, remonta ao tempo do imperador Júlio César, sendo por isso, denominada por cifra de César. Segundo consta, o imperador trocava mensagens com os seus generais, avançando três letras no alfabeto para codificar a mensagem e recuando três letras para descodificar, como ilustra o exemplo 3.1.3:

Exemplo 3.1.3. Cifra de César

Codificar (cifrar): avançar três caracteres do alfabeto para a direita

Alfabeto antes da "cifragem"																					
A	B	C	D	E	F	G	H	I	J	L	M	N	O	P	Q	R	S	T	U	V	X	Z

Descodificar (decifrar): recuar três caracteres do alfabeto para a esquerda

Alfabeto depois da "cifragem"																					
D	E	F	G	H	I	J	L	M	N	O	P	Q	R	S	T	U	V	X	Z	A	B	C

Mensagem original: "ATACAR" Mensagem codificada. "DXDFDU"

Ainda que o uso deste esquema por César tenha sido o primeiro a ser registado, consta que outras cifras de substituições tenham sido utilizadas anteriormente.

A relação deste sistema de encriptação com a aritmética modular é muito simples e é ilustrada no exemplo 3.1.4.

Exemplo 3.1.4. Cifra de César e aritmética modular

A cada letra faz-se corresponder o número inteiro (de 0 a 22), que representa a sua posição no alfabeto.

A	B	C	D	E	F	G	H	I	J	L	M	N	O	P	Q	R	S	T	U	V	X	Z
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22

A cifra de César pode ser então definida por uma função f, que aplica cada número inteiro n, com $0 \leq n \leq 22$, no número inteiro $f(n) = (n+3)\ mod\ 23$.

A mensagem original utilizada no exemplo 3.1.3. foi "ATACAR".

Codificação: $A \leftrightarrow 0$

$f(0) = (0+3)\ mod\ 23 = 3\ mod\ 23 = 3 \leftrightarrow D$

$T \leftrightarrow 18$

$f(18) = (18+3)\ mod\ 23 = 21\ mod\ 23 = 21 \leftrightarrow X$

$C \leftrightarrow 2$

$f(2) = (2+3)\ mod\ 23 = 5\ mod\ 23 = 5 \leftrightarrow F$

$R \leftrightarrow 16$

$f(16) = (16+3)\ mod\ 23 = 19\ mod\ 23 = 19 \leftrightarrow U$

Para decifrar a mensagem codificada, basta considerar a função inversa f^{-1}, que transforma um número inteiro n, com $0 \leq n \leq 22$, noutro número inteiro $f^{-1}(n) = (n-3)\ mod\ 23$.

Note-se que a cifra de César pode ser generalizada considerando que se avança b posições no alfabeto, em vez de três, com $b \in \mathbb{N}$, ou seja $f(n) = (n+b)\ mod\ 23$.

Neste caso não é difícil encontrar a função que descodifica a mensagem original, uma vez que f é uma função bijetiva. A sua inversa será a função definida por

$$f^{-1}(n) = (n-b)\ mod\ 23.$$

Ainda assim esta cifra é pouco segura hoje em dia, pois é relativamente fácil descobrir a chave secreta, podendo ser melhorada definindo f da seguinte forma:

$$f(n) = (an+b)\ mod\ 23$$

onde a deverá ser escolhido de forma a f ser uma bijeção, e, nesse caso,

$$f^{-1}(n) = a^{-1}(n-b)\ mod\ 23$$

A determinação de a^{-1} pode ser feita recorrendo ao algoritmo de Euclides, como descrevemos no exemplo 3.1.5.

Exemplo 3.1.5. Aplicação do algoritmo de Euclides no cálculo do inverso

Seja a função $f(n) = (3n + 7)\ mod\ 23$, então $f^{-1}(n) = 3^{-1}(n - 7)\ mod\ 23$.

Pretendemos saber qual o valor de 3^{-1}, ou seja, qual o inverso de 3 em $\mathbb{Z}_{23}$.

Pelo algoritmo de Euclides,

$$23 = 7 \times 3 + 2 \qquad 1 = 3 - 1 \times 2$$
$$3 = 1 \times 2 + 1 \qquad 1 = 3 - 1 \times (23 - 7 \times 3)$$
$$mdc(23,7) = 1 \qquad 1 = 3 - 1 \times 23 + 7 \times 3$$
$$1 = (-1) \times 23 + 8 \times 3$$
$$1 \equiv 8 \times 3\ mod\ 23$$

Portanto, $3 \times 8 \equiv 1\ mod\ 23$ e consequentemente o inverso de 3 em $\mathbb{Z}_{23}$é 8.

No caso do alfabeto de 23 letras, a existência de a^{-1} em $\mathbb{Z}_{23}$ é garantida, uma vez que 23 é um número primo. Com o alargamento do alfabeto para 26 letras, levanta-se a questão relativamente à determinação do inverso em $\mathbb{Z}_{26}$, uma vez que 26 não é um número primo. O exemplo 3.1.6. ilustra esse prolema.

Exemplo 3.1.6. Alfabeto de 26 letras

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

Para a função f definida por $f(n) = (2n + 2)\ mod\ 26$ tem-se que $f(17) = f(4) = 10$

$$R \leftrightarrow 17$$
$$f(17) = (2 \times 17 + 2)\ mod\ 26 = 36\ mod\ 26 = 10 \leftrightarrow K$$
$$E \leftrightarrow 4$$
$$f(4) = (2 \times 4 + 2)\ mod\ 26 = 10\ mod\ 26 = 10 \leftrightarrow K$$

Como a função não é bijetiva, não existe f^{-1}. Este facto está relacionado com a não existência, em $\mathbb{Z}_{26}$, do inverso de 2 pois se existisse f^{-1}, seria definida por

$$f^{-1}(n) = 2^{-1}(n - 2)\ mod\ 26.$$

Para melhor entendimento sobre a relação entre números primos e existência de inversos, passamos a apresentar algumas considerações sobre a existência de função inversa e de elementos invertíveis em $\mathbb{Z}_n$.

Definição 3.1.1. Números coprimos e função de Euler

Dois números naturais m e n dizem–se *coprimos* ou *primos entre si* se $mdc(m,n) = 1$.

O número de coprimos de n, inferiores a n, designa-se por $\phi(n)$, designando-se por ϕ a função de Euler.

Vejamos agora em que condições um número inteiro $a \neq 0$ é invertível.

Proposição 3.1.1.

Seja a um elemento não nulo de $\mathbb{Z}_n$. Então a é invertível se e só se a e n são coprimos.

Demonstração: Suponhamos que a é invertível em $\mathbb{Z}_n$, o que significa que existe $b \in \mathbb{Z}_n$ tal que $ab \equiv 1 \, mod \, n \Leftrightarrow \exists q \in \mathbb{Z} : ab = nq + 1$. Assim, $1 = ab - nq$.

Seja d um inteiro divisor comum de a e n, então $d|a$ e $d|b$ e consequentemente $d|(ab - nq)$, ou seja $d|1$. Logo $d = 1$ ou $d = -1$ pelo que $mdc(a,n) = 1$.

Provemos agora o recíproco. Por hipótese $mdc(a,n) = 1$. Pelo algoritmo de Euclides usado em ordem inversa (ver exemplo 3.1.5), $\exists s,t \in \mathbb{Z} : 1 = sa + tn$. Assim,

$$sa = 1 - tn \Leftrightarrow sa = n \times (-t) + 1 \Rightarrow sa \equiv 1 \, mod \, n,$$

pois $n \times (-t)$ é múltiplo de n. Portanto $(s + kn)a \equiv 1 \, mod \, n$, pois kn é múltiplo de n, com $k \in \mathbb{Z}$. O inverso de a será um número de $\mathbb{Z}_n$, com a forma $s + kn$, com $k \in \mathbb{Z}$, isto é: $a^{-1} = s + kn, \; k \in \mathbb{Z}$. ■

Para codificar e descodificar uma mensagem através da cifra de César melhorada, utilizando o alfabeto de 26 letras, é necessário saber que números são invertíveis em $\mathbb{Z}_{26} = \{0, 1, \cdots, 25\}$.

Exemplo 3.1.7. Elementos invertíveis de $\mathbb{Z}_{26}$

Pela proposição 3.1.1., os elementos invertíveis de $\mathbb{Z}_{26}$ são todos os coprimos de 26, isto é, são apenas os números $1, 3, 5, 7, 9, 11, 15, 17, 19, 21, 23, 25$, donde $\phi(26) = 12$.

Como $\phi(26) < 25$, podem-se levantar problemas na descodificação. Colmata-se esta dificuldade, por exemplo, acrescentando ao alfabeto, símbolos não numéricos, até aquele perfazer 29 elementos. Como 29 é um número primo, $\phi(29) = 28$ e já não há problemas de codificação ou descodificação com qualquer cifra melhorada de César, da forma

$$f(n) = (an + b) \, mod \, 29, \; \text{com } a \neq 0.$$

Exemplo 3.1.8. Alfabeto de 29 elementos

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	@	*	∪
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28

Descodificar a mensagem "TCIHKX", codificada com a função

$$f(n) = (2n + 2) \, mod \, 29.$$

A função inversa de f será $f^{-1}(n) = 2^{-1}(n-2)\ mod\ 29$ com 2^{-1} o inverso de 2 em $\mathbb{Z}_{29}$.

Pelo algoritmo de Euclides:

$$29 = 14 \times 2 + 1$$
$$mdc(29,2) = 1$$

$$1 = 29 - 14 \times 2$$
$$1 \equiv 2 \times (-14)\ mod\ 29$$
$$1 \equiv 2 \times 15\ mod\ 29$$

O inverso de 2 em $\mathbb{Z}_{29}$ é 15, e portanto $f^{-1}(n) = 15(n-2)\ mod\ 29$.

Descodificando a mensagem "TCIHKX":

$T \leftrightarrow 19$

$f^{-1}(19) = 15 \times (19-2)\ mod\ 29 = 255\ mod\ 29 = 23 \leftrightarrow X$

$C \leftrightarrow 2$

$f^{-1}(2) = 15 \times (2-2)\ mod\ 29 = 0\ mod\ 29 = 0 \leftrightarrow A$

$I \leftrightarrow 8$

$f^{-1}(8) = 15 \times (8-2)\ mod\ 29 = 90\ mod\ 29 = 3 \leftrightarrow D$

$H \leftrightarrow 7$

$f^{-1}(7) = 15 \times (7-2)\ mod\ 29 = 75\ mod\ 29 = 17 \leftrightarrow R$

$K \leftrightarrow 10$

$f^{-1}(10) = 15 \times (10-2)\ mod\ 29 = 120\ mod\ 29 = 4 \leftrightarrow E$

$X \leftrightarrow 23$

$f^{-1}(23) = 15 \times (23-2)\ mod\ 29 = 25\ mod\ 29 = 25 \leftrightarrow Z$

3.2. Implementação na escola

A implementação do tema desenvolvido foi feita numa turma de cozinha e pastelaria "on the job" (CP OTJ), na escola de hotelaria e turismo de Coimbra (EHTC), pelo que iniciamos este capítulo com uma breve caracterização de ambas.

3.2.1. Caracterização da EHTC e da turma CP OTJ

A EHTC, inaugurada em 1989 pelo ministro do comércio e turismo, Joaquim Martins Ferreira do Amaral, está localizada na antiga Quinta da Boavista, casa solarenga da família Barata Alpoim. Esta casa foi adquirida pela Câmara Municipal de Coimbra na década de 80 e cedida ao então Instituto Nacional de Formação Turística, para os efeitos que ainda hoje cumpre: um estabelecimento de ensino e formação profissional em turismo, hotelaria e restauração.

Os cursos de dupla certificação ministrados pela EHTC destinam-se a jovens com o 9º ano de escolaridade, conferindo-lhes equivalência ao 12º ano e uma certificação profissional de nível IV. Estes cursos têm uma duração de três anos, distribuídos por seis semestres. Destacam-se os cursos de técnicas de cozinha e pastelaria (TCP) e técnicas do serviço de restauração e bebidas (TSRB). A oferta formativa de nível IV inclui ainda os cursos de dupla certificação "on the job" (OTJ), que se destinam a jovens com o 11.º ano de escolaridade, conferindo-lhes a equivalência ao 12.º ano de escolaridade e a certificação profissional de nível IV. Os cursos OTJ são cursos de educação e formação para jovens (CEF).

As várias componentes de formação dos cursos OTJ abrangem as áreas científica, sociocultural, tecnológica e prática, sendo que a formação prática é feita em regime de alternância entre a escola e uma unidade hoteleira. Por este motivo, um curso OTJ potencia uma rápida integração no mercado de trabalho. A duração destes cursos é de um ano e destacam-se os cursos de cozinha e pastelaria e técnicas do serviço de restauração e bebidas.

A EHTC tem ainda uma oferta formativa de nível V, pós secundário, com os cursos de gestão e produção de cozinha (GPC), gestão hoteleira – restauração e bebidas (GHRB) e gestão hoteleira – alojamento (GHA).

O plano curricular dos cursos de cozinha e pastelaria "on the job" (CP OTJ), conta, entre as várias componentes de formação, com 16 disciplinas no período escolar inicial e 15 disciplinas no período em alternância com a unidade hoteleira.

A orientação curricular dos cursos OTJ tem por base o programa dos cursos profissionais de nível secundário, tendo em conta a especificidade dos mesmos, que podem receber alunos das diferentes áreas do ensino secundário. Assim, a disciplina de matemática está estruturada de forma a fazer uma revisão aprofundada de conceitos fundamentais do ensino básico, que servem de base às áreas temáticas abordadas.

A turma de CP OTJ era constituída por 20 alunos, dos quais 9 raparigas e 12 rapazes, com uma média de idades de cerca de 20 anos, sendo que alguns elementos do grupo já tinham tido contacto com o mercado de trabalho. Apresentou-se como uma turma heterogénea, devido aos diferentes níveis de conhecimento no que diz respeito à disciplina de Matemática.

3.2.2. Atividade: aplicações da aritmética modular

A atividade proposta foi desenvolvida numa aula de dois tempos de 50 minutos, na sala de informática, e foi feita uma breve apresentação da aritmética modular (Anexo 3), como uma ferramenta importante da teoria de números. Foram apresentados alguns exemplos como a aritmética do relógio, os calendários e a

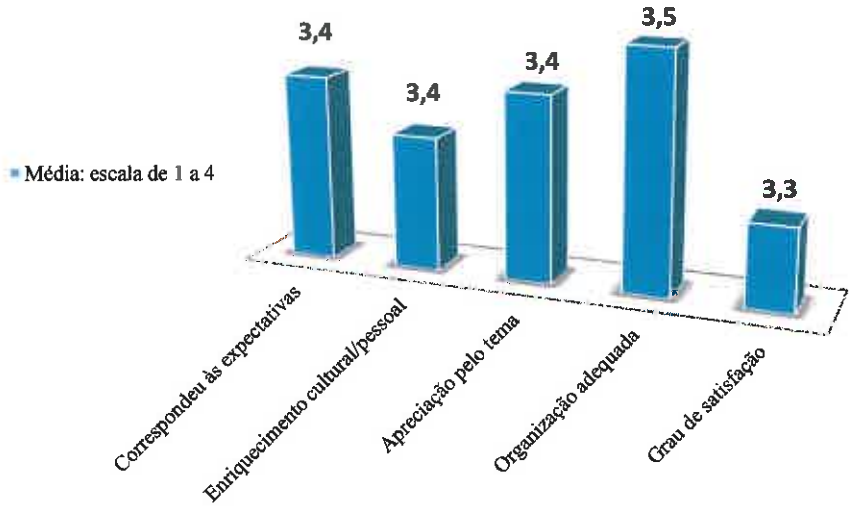
criptografia, tendo sido, por isso, introduzida a noção de congruência. Foram também apresentadas algumas das suas aplicações, como o funcionamento dos sistemas de identificação modulares, nomeadamente, o sistema de identificação de livros, International Standard Book Number (ISBN), o sistema de identificação de artigos, European Article Number (EAN), o sistema de identificação dos bilhetes de identidade (BI) e o sistema de identificação dos cartões de cidadão (CC).

De modo a despertar o interesse dos alunos, a apresentação foi feita recorrendo a esquemas animados e utilizando exemplos da vida quotidiana, práticos e palpáveis, como o relógio analógico, calendários, livros, artigos e cartões. No sentido de consolidarem e aplicarem os conceitos apresentados, os alunos realizaram, ao longo da aula, uma ficha de trabalho (Anexo 4), com aplicações práticas da aritmética modular. Puderam ainda verificar experimentalmente, em www.atractor.pt/mat/alg_controlo/, o resultado de alguns exercícios propostos na ficha de trabalho. O programa Octave, um *freeware* do programa Matlab, foi instalado nos computadores a que os alunos tiveram acesso, de modo a poderem verificar, através do algoritmo concebido para o efeito, se os respetivos números do CC estavam corretamente escritos.

3.2.3. Avaliação da atividade por parte dos alunos

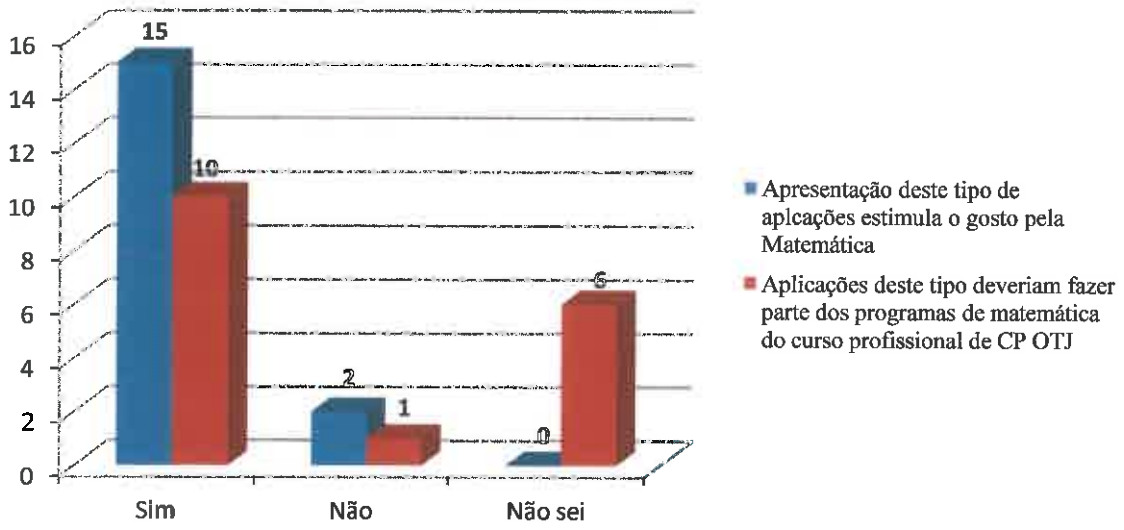
A atividade implementada foi avaliada através de um questionário (Anexo 5), que inquiriu os alunos acerca do interesse geral pelo tema, cujos resultados se apresentam na figura 3.2.1.

Figura 3.2.1. Opinião dos alunos sobre o interesse geral pela atividade



No mesmo questionário, os alunos foram chamados a dar a sua opinião sobre a influência deste tipo de aplicações matemáticas, no estímulo da aprendizagem pela disciplina. Também deram a sua opinião acerca da inclusão, ou não, deste tipo de aplicações nos programas de matemática do ensino profissional. Essa opinião está retratada na figura 3.2.2:

Figura 3.2.2. Opinião dos alunos sobre a motivação pelo tema



A maioria dos alunos revelou interesse no tema abordado, tendo participado ativamente com dúvidas e sugestões.

4 Conclusão

"A linguagem realmente não é um veículo de transmissão perfeito."
Mark Zuckerberg

A análise do funcionamento dos sistemas de identificação detetores de erros fez-nos compreender que, nos sistemas *módulo k*, a deteção de erros será mais eficiente se k for um número primo, com $k > 10$. Existem, no entanto, alguns constrangimentos no caso de $k > 10$, pelo facto de ser necessário recorrer a dois ou mais algarismos de teste, ou a caracteres não numéricos para dígitos de controlo. Esses problemas são ultrapassados com uma generalização e/ou modificação dos sistemas *módulo 10*, que permite aumentar significativamente a sua eficiência. Um exemplo de um sistema *módulo 10* melhorado é o sistema implementado no cartão de cidadão português, que vem colmatar a falha existente no sistema de identificação do bilhete de identidade.

A perceção do funcionamento do sistema do cartão de cidadão foi determinante para a conceção do algoritmo apresentado.

Concluímos que, mesmo melhorados, a eficácia dos sistemas *módulo k* não é total, pelo que a aritmética modular não nos fornece uma solução ótima para a deteção dos erros mais frequentes. Essa solução é-nos fornecida pela teoria de grupos.

A aritmética modular ajudou-nos, no entanto, a apresentar aos alunos, os sistemas de identificação modulares mais conhecidos e por eles utilizados no dia-a-dia. Também foi importante para ensinar a codificar e a descodificar mensagens secretas através da criptografia. Os exemplos e aplicações da aritmética modular, no quotidiano, foram fundamentais para cativar a atenção dos discentes.

Consideramos que, sendo a matemática a disciplina que, por excelência, ensina a pensar, no ensino profissional deve apresentar-se não só como uma atividade intelectual, mas também como uma ferramenta útil na vida quotidiana e profissional.

Segundo o programa para o ensino profissional, o essencial da aprendizagem da matemática concentra-se mais na procura da resolução de problemas e de aplicações matemáticas, pelo que consideramos que abordar conteúdos do interesse dos alunos, relacionando-os com as aplicações práticas do quotidiano, é fundamental para estimular a motivação e a criatividade dos mesmos.

O tema abordado permitiu desenvolver o raciocínio de forma lúdica e pensar de modo abstrato. Assim, temas como este poderiam ser abordados nos programas de matemática do ensino profissional. Pelas características destes cursos, o recurso às aplicações práticas e quotidianas da matemática é um fator essencial para despertar o interesse dos alunos e, consequentemente, obter sucesso no aproveitamento da disciplina.

Referências bibliográficas

[1] J. Picado, *A álgebra dos sistemas de identificação*, Boletim da SPM, nº 44, Abril de 2001, pp.39–73.

[2] J. Picado, *Estruturas Discretas* - Textos de apoio, Cap. 3., DMUC, 2010, pp.85–94.

[3] J. Buescu, *O Mistério do Bilhete de Identidade e Outras Histórias*, 9ª ed, Gradiva, Lisboa, 2004.

[4] R. Lidl, e H. Niederreiter, *Introduction to finite fields and their applications*, Cambridge University Press, Cambridge, 2002.

Referências webgráficas

1 *Sistemas de Identificação com algarismos de Controlo*, www.atractor.pt/mat/alg_controlo/ , consultado ao longo de todo o trabalho

2 *Sistemas de identificação com algarismos de controlo*, http://www.atractor.pt/publicacoes/artigo-apm.pdf, consultado em janeiro de 2013

3 *Codificação para controlo de erros. História da codificação*, http://paginas.fe.up.pt/~sam/homepage/codes.htm, consultado em janeiro de 2013

4 *A Matemática do cartão de cidadão I*, *http://www.tribunadasilhas.pt/index.php/opiniao/item/5450-a-matemática-do-cartão-de-cidadão-i*, consultado em janeiro de 2013

5 *A Matemática do cartão de cidadão II*, http://www.tribunadasilhas.pt/index.php/opiniao/item/5608-a-matemática-do-cartão-de-cidadão-ii, consultado em fevereiro de 2013

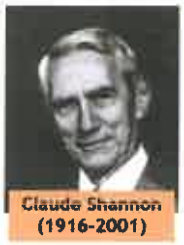
6 *Guia prático de utilização do cartão de cidadão*, *http://www.cartaodecidadao.pt/images/stories/cc_nota_informativa.pdf, consultado em fevereiro de 2013*

Anexos

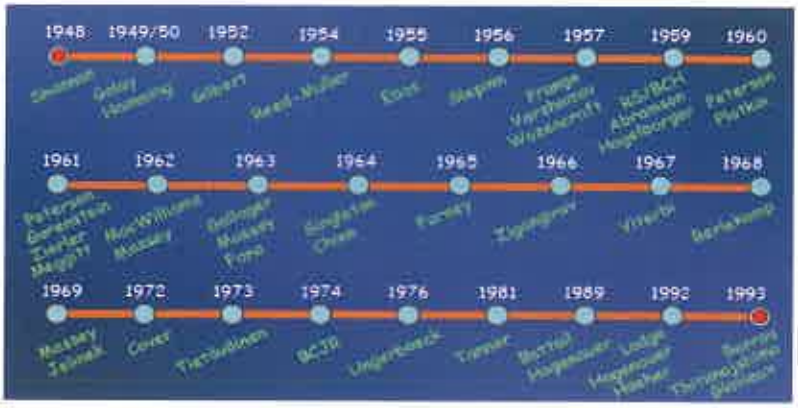
Anexo 1 Referência histórica sobre codificação para controlo de erros

O seguinte cronograma refere as principais personagens da história da codificação para controlo de erros:

Tudo começou com o "Claude" Shannon



Claude Shannon
(1916-2001)



Claude Berrou
(1951-)

e recomeçou com o "Claude" Berrou, com os seus 'turbo-códigos'.

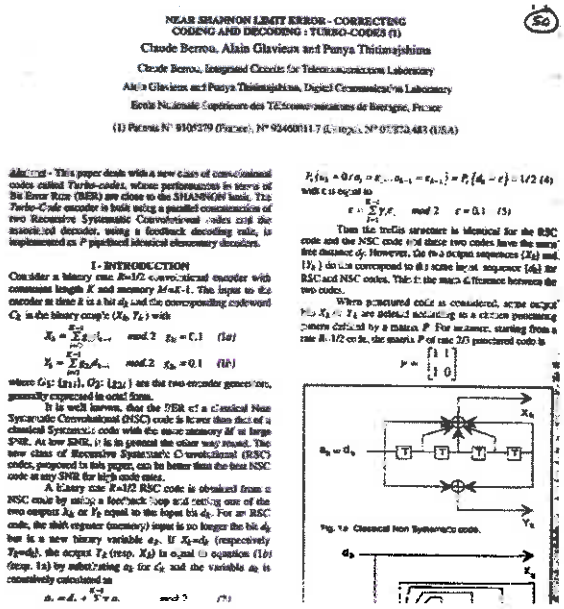
As primeiras páginas dos principais artigos dos "Claudes" estão representadas nas seguintes imagens:

A Mathematical Theory of Communication

By C. E. SHANNON

Communication in the Presence of Noise*

CLAUDE E. SHANNON, MEMBER, IRE



NEAR SHANNON LIMIT ERROR - CORRECTING CODING AND DECODING : TURBO-CODES (1)

Claude Berrou, Alain Glavieux and Punya Thitimajshima

Claude Berrou, Integrated Circuits for Telecommunications Laboratory

Alain Glavieux and Punya Thitimajshima, Digital Communications Laboratory

Ecole Nationale Supérieure des Télécommunications de Bretagne, France

(1) Patents N° 9105279 (France), N° 92460011.7 (Europe), N° 07/870,483 (USA)

Anexo 2 Sistema de identificação das notas do euro

O número de série que se encontra numa qualquer nota de euro (verdadeira) é constituído por uma letra seguida de onze algarismos, como ilustra a figura 2.4.1:

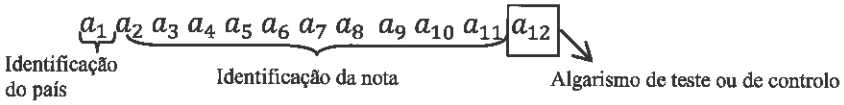
Figura 2.1. Nota de 10 euros e respetivo número de série

A letra representa o país do qual é proveniente a nota e à qual é atribuído um valor numérico constante na tabela 2.1.

Tabela 2.1. Valor atribuído a cada letra correspondente ao respetivo país

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

Os dez algarismos que se seguem à letra identificam a nota e o último algarismo é o digito de controlo. Assim, o número de série de uma nota de euro pode ser representado da seguinte forma:



a_{12} é escolhido de modo a que a soma de teste seja divisível por 9, ou seja:

$$S = a_1 + a_2 + \cdots + a_{10} + a_{11} + a_{12} \equiv 0 \; mod \; 9,$$

com $a_1 \in \{1, \cdots, 9\}$, o valor representado pela letra correspondente ao país. Então,

$$a_{12} \equiv -(a_1 + a_2 + a_3 + a_4 + a_5 + a_6 + a_7 + a_8 + a_9 + a_{10} + a_{11}) \bmod 9.$$

Repare-se que este sistema não deteta nenhuma transposição de algarismos, uma vez que os pesos são todos iguais a 1, nem deteta todos os erros singulares, conforme ilustra o exemplo 2.1.

Exemplo 2.1.

No exemplo da figura 2.1, o valor de T é 3, conforme a tabela 2.1 e S é tal que

$$S = 3 + 2 + 3 + 1 + 8 + 6 + 9 + 0 + 9 + 6 + 9 + 7 = 63 \equiv 0\ mod\ 9,$$

pelo que o dígito de controlo está bem calculado e podemos concluir que a nota é verdadeira.

No entanto, note-se que se o algarismo zero for trocado por um nove ou vice-versa o número de controlo mantém-se inalterado uma vez que $9 \equiv 0\ mod\ 9$, verificando que este sistema é pouco eficiente.

Anexo 3 Apresentação do tema em sala de aula

Noções básicas de aritmética modular

1. Exemplos

Alguns exemplos que utilizam a noção de congruência:

Exemplo 1.1 Aritmética do relógio

"Aritmética do relógio": congruência módulo 12.

0:00 → 12:00 após 12 horas, **0+12=12**
12 ≡ 0 *módulo 12*

0:00 → 13:00 após 13 horas, **1+12=13**
13 ≡ 1 *módulo 12*

0:00 → 02:00 após 26 horas, 2+2x12=26
26 ≡ 2 *módulo 12*

Olhamos para o tempo entre os múltiplos de 12.

$0 \equiv 12 \equiv 24 \equiv \ldots \ mod\ 12$
$1 \equiv 13 \equiv 25 \equiv \ldots \ mod\ 12$
$2 \equiv 14 \equiv 26 \equiv \ldots \ mod\ 12$

Exemplo 1.2 Calendários

ABRIL

DOM	SEG	TER	QUA	QUI	SEX	SAB
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30				

O mês de Abril começou a uma 2.ª feira...

Em que dia da semana irá calhar o feriado do dia 25 de Abril de 2016?

Dias	Segunda	Terça	Quarta	Quinta	Sexta	Sábado	Domingo
n.º	1	2	3	4	5	6	0

ARITMÉTICA MODULAR E SUAS APLICAÇÕES

TURISMO DE PORTUGAL escola de hotelaria e turismo de coimbra

Quantos dias podemos contar até o dia 25 de Abril de 2016?

Mês	Ano				
	2013	2014	2015	2016 (bissexto)	
Janeiro		31	31	31	
Fevereiro		28	28	29	
Março		31	31	31	
Abril	30	30	30	25	
Maio	31	31	31		
Junho	30	30	30		
Julho	31	31	31		
Agosto	31	31	31		
Setembro	30	30	30		
Outubro	31	31	31		
Novembro	30	30	30		
Dezembro	31	31	31		
Total	275	365	365	116	**1121**

Contabilizámos **1121** dias.

ARITMÉTICA MODULAR E SUAS APLICAÇÕES

TURISMO DE PORTUGAL escola de hotelaria e turismo de coimbra

Domingo	**2.ª feira**	...	**Sábado**
M_7	M_7+1	...	M_7+6
n.ºs que divididos por 7 dão resto **0**	n.ºs que divididos por 7 dão resto **1**	...	n.ºs que divididos por 7 dão resto **6**

DOM	SEG	TER	QUA	QUI	SEX	SAB
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30				

Divisão inteira de 1121 por 7 e observar o seu **resto**:

$$1121 = 160 \times 7 + \mathbf{1}$$

$$\mathbf{1121} \equiv \mathbf{1} \ mod\ 7$$

$$\begin{array}{r|l} 1121 & 7 \\ \hline 42 & 160 \\ 01 & \\ \mathbf{1} & \end{array}$$

25 / Abril / 2016:

↓

2.ª feira

Dias	n.º
Domingo	**0**
Segunda	**1**
Terça	**2**
Quarta	**3**
Quinta	**4**
Sexta	**5**
Sábado	**6**

MINISTÉRIO DA ECONOMIA E DA INOVAÇÃO

Exemplo 1.3 Criptografia e congruência

CIFRA DE CÉSAR

Alfabeto antes da cifragem																					
A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	X	Z

Cifrar: avançar 3 caracteres do alfabeto para a direita

Alfabeto depois da cifragem																					
D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	X	Z	A	B	C

Decifrar: recuar 3 caracteres para a esquerda

Mensagem antes de Cifrar: "ATACAR"

Mensagem após a Cifragem: "DXDFDU"

Letra → n.º que representa a sua posição no alfabeto

A	B	C	D	E	F	G	H	I	J	L	M	N	O	P	Q	R	S	T	U	V	X	Z
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22

Cifrar (codificar):

cod = n.º letra + 10 (Se *cod* ≥ 23, volta-se ao início do alfabeto)

Decifrar (descodificar):

des = n.º letra - 10 (Se *des* < 10 faz-se → *des* + 23)

Exemplo:

$X = 21 \xrightarrow{cifrar} 21 + 10 = 31 \equiv 8 \pmod{23} \rightarrow I$ (congruência ***mod 23***)

$I = 8 \xrightarrow{8<10} 8 + 23 = 31 \xrightarrow{decifrar} 31 - 10 = 21 \rightarrow X$

Se a mensagem recebida for **CGP EPNL**, como decifrá-la?

Decifrar (descodificar):

CGP

EPNL

des = n.º letra - 10

(Se *des* < 10 faz-se → *des* = *des* + 23)

(Se *des* ≥23, volta-se ao início do alfabeto)

A	B	C	D	E	F	G	H	I	J	L	M	N	O	P	Q	R	S	T	U	V	X	Z
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22

C → 2 ≡ 25 (*mod 23*) (2+23=25) → 25−10 = 15 → → Q

G → 6 ≡ 29 (*mod 23*) (6+23=29) → 29−10 = 19 → → U

P → 14 → → → → → → → → → 14−10 = 4 → → E

E → 4 ≡ 27 (*mod 23*) (4+23=27) → 27 − 10 = 17 → S

P → 14 → → → →→ → → → → → 14 − 10 = 4 → → E

N → 12 → → → → → → → → → 12 − 10 = 2 → → C

L → 10 → → → → → → → → → 10 − 10 = 0 → → A

1.2 Noção de congruência módulo k

• Dois números inteiros ***a*** e ***b*** são **congruentes**, *módulo k se* tiverem o mesmo resto quando divididos pelo mesmo número inteiro k (k>0), Esta relação de congruência representa-**se** normalmente por:

$$a \equiv b \bmod k$$

• Outro modo equivalente de dizer que *a* e *b* são congruentes, *módulo k* é verificar que a diferença ***a − b*** *ou* ***b − a*** é divisível por *k* (ou que *k* ***é*** divisor dessa diferença)

Exemplo 1.2.1

Se 53 ≡ 3 mod 5, então 53 − 3 é divisível por 5.

De facto

$$\begin{array}{r|l} 53 & 5 \\ 03 & 1 \end{array} \qquad 53 - 3 = 50 \qquad \begin{array}{r|l} 50 & 5 \\ 00 & 10 \\ 0 & \end{array}$$

2. Aplicações da aritmética modular

MATEMÁTICA **MATEMÁTIAC**

962531479 **962531497**

2.1 Sistemas de identificação modulares

Em muitas situações, utilizam-se números (códigos numéricos) para identificar algo:
Livros (ISBN), artigos (EAN), pessoas (BI ou CC), etc.

Como detetar um erro na transmissão de um número?

Criação de **algarismos de teste ou de controlo** do número.

Esses algarismos são calculados com base na noção de congruência.

Exemplo 2.1. ISBN (International Standard Book Number)

ISBN-10: $a_1\ a_2\ a_3\ a_4\ a_5\ a_6\ a_7\ a_8\ a_9\ a_{10}$ → Algarismo de teste

Identificação do livro

Dicionário Prático de Matemática

ISBN-10: 9 7 2 - 7 1 0 - 2 2 5 - 5

Livros editados em Portugal — Editora — Item — Algarismo de teste

Sequência: 10 9 8 7 6 5 4 3 2

10 9 8 7 6 5 4 3 2
9 7 2 7 1 0 2 2 5

10 x 9 + 9 x 7 + 8 x 2 + 7 x 7 + 6 x 1 + 5 x 0 + 4 x 2 + 3 x 2 + 2 x 5 = 248

C → dígito de controlo: **248 + C ≡ 0 *mod 11*** (248 + C divisível por 11)

C = 11 − 6 = 5 → Algarismo de teste

248 | 11
028 | 22
06

13 ARITMÉTICA MODULAR E SUAS APLICAÇÕES

TURISMO DE PORTUGAL escola de hotelaria e turismo de coimbra

ISBN-13: $a_1\ a_2\ a_3\ a_4\ a_5\ a_6\ a_7\ a_8\ a_9\ a_{10}\ a_{11}\ a_{12}\ a_{13}$

Identificação do produto e do produtor — Algarismo de teste

1 de Janeiro de 2007

Matemática *Módulo A*

ISBN-13: 978-972-044-407-3

Artigo — Livros editados em Portugal — Editora — Item — Algarismo de teste

Sequência: 1 3 1 3 1 3 1 3 1 3 1 3

1 3 1 3 1 3 1 3 1 3 1 3
9 7 8 9 7 2 0 4 4 4 0 7

1 x 9 + 3 x 7 + 1 x 8 + 3 x 9 + 1 x 7 + 3 x 2 + 1 x 0 + 3 x 4 + 1 x 4 + 3 x 4 + 1 x 0 + 3 x 7 = 127

C → : **127 + C ≡ 0 *mod 10*** (127 + C divisível por 10)

C = 10 − 7 = 3 → Algarismo de teste

127 | 10
027 | 12
07

MINISTÉRIO DA ECONOMIA E DA INOVAÇÃO

14 ARITMÉTICA MODULAR E SUAS APLICAÇÕES

TURISMO DE PORTUGAL escola de hotelaria e turismo de coimbra

Exemplo 2.2. EAN (European Article Number)

EAN 13: $a_1\ a_2\ a_3\ a_4\ a_5\ a_6\ a_7\ a_8\ a_9\ a_{10}\ a_{11}\ a_{12}\ a_{13}$ → Algarismo de teste

Identificação do produto e do produtor

Creme para mãos

EAN 13: 847-000-261-073-6 → Algarismo de teste

Artigo — Produtor — Produto — Item

Sequência: 1 3 1 3 1 3 1 3 1 3 1 3

1 3 1 3 1 3 1 3 1 3 1 3
8 4 7 0 0 0 2 6 1 0 7 3

1 x 8 + 3 x 4 + 1 x 7 + 3 x 0 + 1 x 0 + 3 x 0 + 1 x 2 + 3 x 6 + 1 x 1 + 3 x 0 + 1 x 7 + 3 x 3 = 64

C → : **64 + C ≡ 0 *mod 10*** (64 + C divisível por 10)

C = 10 − 4 = 6 → Algarismo de teste

64 | 10
04 | 6

MINISTÉRIO DA ECONOMIA E DA INOVAÇÃO

Exemplo 2.3. BI (Bilhetes de identidade)

BI: $a_1\ a_2\ a_3\ a_4\ a_5\ a_6\ a_7\ a_8\ \boxed{a_9}$ → Algarismo de teste

(ID) Identificação do BI

BI: 0 8 4 9 6 7 1 2 - 9

Identificação da pessoa

Algarismo de controlo

Sequência: 9 8 7 6 5 4 3 2

9 8 7 6 5 4 3 2
0 8 4 9 6 7 1 2

9 x 0 + 8 x 8 + 7 x 4 + 6 x 9 + 5 x 6 + 4 x 7 + 3 x 1 + 2 x 2 = 211

C → dígito de controlo: **211 + C ≡ 0 *mod 11*** (211 + C divisível por 11)

C = 11 − 2 = 9 → Algarismo de controlo

211 | 11
101 | 19
02

E se o resto fosse 1?

BI: 0 6 9 9 4 7 0 4 - 0

Identificação da pessoa

Algarismo de teste

9 8 7 6 5 4 3 2
0 6 9 9 4 7 0 4

9 x 0 + 8 x 6 + 7 x 9 + 6 x 9 + 5 x 4 + 4 x 7 + 3 x 0 + 2 x 4 = 221

C = 11 − 1 = 10 ≠ 0 !

221 | 11
001 | 20
01

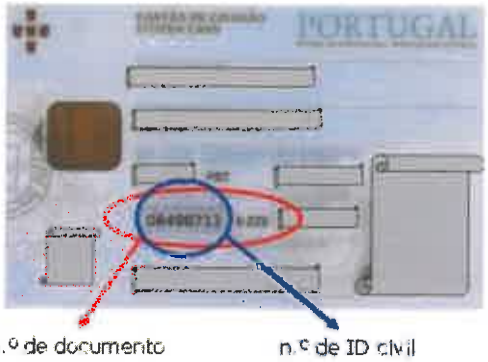
10 não é um algarismo!

O sistema do BI não está implementado corretamente.

www.atractor.pt/mat/alg_controlo/index.htm

Exemplo 2.4 CC (Cartão do cidadão)

Além do ID (antigo n.º BI), o número de documento do cartão do cidadão inclui dois caracteres não numéricos e um algarismo de teste do número de documento.



CC: 0 8 4 9 6 7 1 2 9 Z Z 6

Testem o número de documento do vosso cartão de cidadão no programa Octave.

18 ARITMÉTICA MODULAR E SUAS APLICAÇÕES
TURISMO DE PORTUGAL escola de hotelaria e turismo de coimbra

FIM

MINISTÉRIO DA ECONOMIA E DA INOVAÇÃO

Anexo 4 Ficha de trabalho proposta

TURISMO DE PORTUGAL escola de hotelaria e turismo de coimbra	Ano letivo 2012/2013 - 2.º Semestre
Cozinha e Pastelaria On the Job	Ficha de trabalho de Matemática - Aritmética Modular

Nome: _____

Noção de congruência:
Dois números são **congruentes** se divididos **pelo mesmo** número inteiro positivo, tiverem o mesmo resto.

Exemplo:

Como em ambas as divisões (por 7), **o resto é 5, diz-se que 26 e 12 são números congruentes** *módulo 7* ($12 \equiv 26 \ mod\ 7$)

1) Verifique se os seguintes números são congruentes *mod 7* :

a) 29 e 43

b) 38 e 26

c) 53 e 32

2) Em que dia da semana nasceu?
Para saber em que dia da semana nasceu, é necessário ter em conta as informações das tabelas abaixo:

TABELA 1

Tabela dos meses			
Janeiro	1	Julho	0
Fevereiro	4	Agosto	3
Março	4	Setembro	6
Abril	0	Outubro	1
Maio	2	Novembro	4
Junho	5	Dezembro	6

TABELA 2

Domingo	0
2.ª feira	1
3.ª feira	2
4.ª feira	3
5.ª feira	4
6.ª feira	5
Sábado	6

2.1) Indique o ano do seu nascimento: _____. Subtraia 1900 a esse valor. Isto é: _____ – 1900 = _____

2.2) Faça a divisão inteira do resultado por 4 (isto é, determine o quociente inteiro dessa divisão): _____

2.3) Associe o mês do seu nascimento ao algarismo correspondente, conforme a Tabela 1: _____

2.4) Indique a "quantos" nasceu: _____. Subtraia uma unidade a esse valor. Isto é: _____ – 1 = _____

2.5) Adicione os quatro números encontrados nas alíneas anteriores: _____ + _____ + _____ + _____ = _____

2.6) Divida o resultado por 7 e considere o resto da divisão: _____

2.7) Associe esse resto ao dia da semana, conforme a tabela 2 acima. Nasceu num(a) _____

3) Observe a seguinte mensagem codificada:

WESDY LOW

Decifre a mensagem sabendo que foi "codificada" segundo a chave: **adicionar 10 unidades** ao número correspondente à letra constante na seguinte tabela:

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

A mensagem é: _ _ _ _ _ _ _ _ _ _

4) O *Dicionário Prático Matemática* foi editado antes de 2007. No entanto, para facilitar a sua comercialização e ser identificado por leitores óticos, o seu ISBN foi convertido em código de barras, conforme mostra a figura:

Através da figura pode-se constatar que o último algarismo (de controlo) é o 9.
Antes de ser convertido num ISBN 13 (EAN), o *Dicionário Prático Matemática* tinha o seguinte ISBN 10:

Determine o seu algarismo de controlo, tendo em conta que deve utilizar a sequência **10 9 8 7 6 5 4 3 2** e a congruência *mod 11*.

R: _____

5) Imagine que está **na caixa de um** supermercado para pagar **as** suas compras **e um** produto **de** superfície rugosa não consegue ser identificado pelo leitor ótico. A funcionária digita os algarismos do código do produto, mas não consegue obter o preço do mesmo. O produto tem o seguinte código de barras:

Como pode verificar, **falta o** algarismo de controlo. Determine-o, recordando que deve **utilizar a** sequência **1 3 1 3 1 3 1 3 1 3 1** e a congruência *mod 10*.

R: _____

NOTA: confirme o algarismo encontrado consultando **www.atractor.pt/mat/alg_controlo/index.htm**

6) Confirme, através da sequência **9 8 7 6 5 4 3 2** e da congruência *mod 11*, o valor do algarismo de controlo do seu BI.

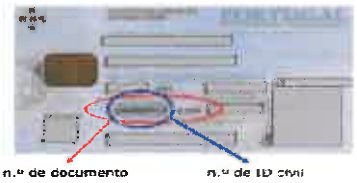
R: _____

NOTAS:

1. se o seu ID (n.º de identificação) tiver menos de 8 algarismos, deve acrescentar zeros à esquerda do n.º de modo a perfazer essa quantidade de algarismos.

2. Confirme o algarismo encontrado consultando **www.atractor.pt/mat/alg_controlo/index.htm**

7) O número de documento do Cartão do Cidadão inclui o número de identificação civil ID (antigo n.º do BI), dois carateres alfanuméricos e um algarismo de controlo do número do documento.
Exemplo:



Observe o n.º de documento do seu CC. Considere $Z = 35$ e se for o caso, $Y = 34$.

7.1) Adicione os valores dos dígitos das posições ímpares, a contar da direita para a esquerda. Designe esse resultado por S_1;

R: $S_1 =$ _____

7.2) Multiplique por 2, o valor de cada um dos dígitos das posições pares, a contar da direita para a esquerda.

7.3) Retire 9 unidades aos valores obtidos em 7.2) com mais de um dígito.

7.4) Adicione os valores obtidos em 7.3). Designe esse resultado por S_2.

R: $S_2 =$ _____

7.5) Adicione o resultado obtido em 7.1) ao resultado obtido em 7.4) e designe esse valor por S, isto é: $S = S_1 + S_2$.

R: $S =$ _____

7.6) Calcule $\frac{S}{10}$.

R: $\frac{S}{10} =$ _____

7.7) Atendendo ao resultado obtido em 7.6), estará o seu número de CC corretamente escrito? Justifique.

R: _____

_____

Bom trabalho
Cláudia Sanhá Santos

Anexo 5 Inquérito aos alunos

TURISMO DE PORTUGAL escola de hotelaria e turismo de coimbra

INQUÉRITO DE AVALIAÇÃO DA ATIVIDADE SOBRE ARITMÉTICA MODULAR E SUAS APLICAÇÕES

Com o objetivo de refletir e avaliar a atividade sobre aritmética modular e suas aplicações, realizada em 29 / 04 / 2013, solicito a vossa participação através do preenchimento da tabela e das questões que a seguir se apresentam:

Atividade	**1**	**2**	**3**	**4**
Correspondeu às suas expectativas				
Contribuiu para aprofundar o seu enriquecimento cultural / pessoal				
O tema tratado foi do seu agrado				
A organização da atividade foi adequada				
De forma geral, indique o seu grau de satisfação				

Acha que aplicações da matemática em situações da vida corrente como a apresentada na atividade desenvolvida deveriam fazer parte do programa da disciplina de Matemática no curso profissional de Cozinha / Pastelaria On The Job?

Sim		Não		Não sei	

Acha que a apresentação deste tipo de aplicações poderia estimular mais o gosto pela matemática?

Sim		Não		Não sei	

Outras observações que considere pertinentes:

Agradecida pela colaboração
A professora,
Cláudia Seabra Santos

Cláudia Seabra Santos

Julho de 2013