Universidade Federal de Mato Grosso
Instituto de Ciências Exatas e da Terra
Departamento de Matemática

Fabricio Antonio Barzan

Explorando Elementos da Criptografia RSA no Processo Educacional.

Cuiabá/MT 2025

Fabricio Antonio Barzan

Explorando Elementos da Criptografia RSA no Processo Educacional.

Dissertação apresentada ao curso de Mestrado Profissional em Matemática - Profmat, da Universidade Federal de Mato Grosso, como requisito parcial para obtenção do título de Mestre em Matemática.

Área de concentração: Matemática na Educação Básica. Linha de pesquisa: Formação de Professores de Matemática da Educação Básica.

Prof. Dr. Pedro Manuel Sanchez Aguilar
Orientador

Cuiabá - MT
2025

Dados Internacionais de Catalogação na Fonte.

B296e Barzan, Fabricio Antonio.
Explorando Elementos da Criptografia RSA no Processo Educacional [recurso eletrônico] : Criptografia RSA no Processo Educacional / Fabricio Antonio Barzan. -- Dados eletrônicos (1 arquivo : 113 f., il. color., pdf). -- 2025.

Orientador: Pedro Manuel Sánchez Aguilar.
Dissertação (mestrado profissional) – Universidade Federal de Mato Grosso, Instituto de Ciências Exatas e da Terra, Programa de Pós-Graduação Profissional em Matemática, Cuiabá, 2025.
Modo de acesso: World Wide Web: https://ri.ufmt.br.
Inclui bibliografia.

1. Ensino de matemática. 2. Proposta didática. 3. Criptografia RSA. I. Sánchez Aguilar, Pedro Manuel, *orientador.* II. Título.

Ficha catalográfica elaborada automaticamente de acordo com os dados fornecidos pelo(a) autor(a).

MINISTÉRIO DA EDUCAÇÃO

UNIVERSIDADE FEDERAL DE MATO GROSSO

PRÓ-REITORIA DE ENSINO DE PÓS-GRADUAÇÃO

PROGRAMA DE PÓS-GRADUAÇÃO EM MATEMÁTICA EM REDE NACIONAL - PROFMAT

AV. FERNANDO CORRÊA DA COSTA, 2367 - BOA ESPERANÇA - 78.060-900 - CUIABÁ/MT

FONE: (65) 3615-8576 – E-MAIL: PROFMAT.ICET@UFMT.BR

FOLHA DE APROVAÇÃO

TÍTULO**: EXPLORANDO ELEMENTOS DA CRIPTOGRAFIA RSA NO PROCESSO EDUCACIONAL.**

AUTOR: **FABRICIO ANTONIO BARZAN**

Dissertação defendida e aprovada em 21 de março de 2025.

COMPOSIÇÃO DA BANCA EXAMINADORA

1. **Prof. Dr. Pedro Manuel Sanchez Aguilar** (Presidente Banca/orientador)

Instituição: Universidade Federal de Mato Grosso

2. **Prof. Dr. Hector Flores Callisaya** (Membro Interno)

Instituição: Universidade Federal de Mato Grosso

3. **Prof. Dr. Jorge Mauricio Jaramillo Monsalve** (Membro externo)

Instituição: Instituto Federal de Mato Grosso

Cuiabá, 21/03/2025.

Documento assinado eletronicamente por **JORGE MAURICIO JARAMILLO MONSALVE**, **Usuário Externo**, em 22/03/2025, às 11:13, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020 .

Documento assinado eletronicamente por **HECTOR FLORES CALLISAYA**, **Docente da Universidade Federal de Mato Grosso**, em 22/03/2025, às 11:38, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020 .

Documento assinado eletronicamente por **PEDRO MANUEL SANCHEZ AGUILAR**, **Coordenador(a) do Programa de Pós-Graduação em Matemática em Rede - PROFMAT / ICET - UFMT**, em 22/03/2025, às 11:44, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020 .

A autenticidade deste documento pode ser conferida no site http://sei.ufmt.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **7746734** e o código CRC **BE797965**.

Referência: Processo nº 23108.040389/2024-17 SEI nº 7746734

Agradecimentos

A realização deste trabalho só foi possível graças ao apoio e à dedicação de muitas pessoas e da instituição, às quais registro aqui minha mais sincera gratidão.

Em primeiro lugar, agradeço ao meu orientador, Prof. Dr. Pedro Manuel Sánchez Aguilar, por sua paciência, dedicação e valiosas orientações durante o desenvolvimento deste trabalho. Sua experiência e contribuições foram fundamentais para a conclusão deste trabalho.

Aos professores do Programa PROFMAT – Mestrado Profissional em Matemática em Rede Nacional, da Universidade Federal de Mato Grosso, que contribuíram para o meu crescimento acadêmico e profissional por meio de aulas enriquecedoras e discussões reflexivas. Agradeço também à instituição pelo suporte e pela oportunidade de fazer parte deste programa.

Aos meus colegas e amigos do PROFMAT, pela amizade, troca de ideias e apoio mútuo durante toda esta jornada.

À minha família, em especial aos meus pais, pelo amor, paciência e suporte incondicional ao longo desta trajetória. A vocês, minha eterna gratidão por estarem sempre ao meu lado nos momentos de dificuldade e nas conquistas.

À minha esposa, pelo companheirismo, compreensão e incentivo em todos os momentos. Sua presença foi meu porto seguro ao longo desta jornada. E à minha filha, cuja alegria e amor incondicional iluminaram meus dias e me deram forças para persistir.

Por fim, agradeço a todos que, direta ou indiretamente, contribuíram para a realização deste trabalho. A cada um de vocês, meu mais sincero obrigado.

"Matemática pura é, à sua maneira,
a poesia de ideias lógicas"

Albert Einstein.

Resumo

Neste trabalho, apresenta-se uma proposta alternativa para o ensino de Matemática na Educação Básica, destinada a professores, com o objetivo de despertar o interesse dos alunos e facilitar a compreensão da aplicabilidade da Matemática no cotidiano. A proposta integra conceitos matemáticos e o uso de planilhas eletrônicas, tendo como pano de fundo a evolução da criptografia, com ênfase na criptografia RSA. Além disso, são sugeridas atividades viáveis para a implementação nas escolas estaduais de Mato Grosso, considerando sua estrutura e os recursos disponíveis, ressaltando a importância da aplicação prática para o diagnóstico, para ajustes metodológicos e para o desenvolvimento de novas abordagens.

Palavras-chave: Ensino de matemática; proposta didática; criptografia RSA.

Abstract

This work presents an alternative approach to teaching Mathematics in Basic Education, aimed at teachers, with the objective of sparking students' interest and facilitating the understanding of the practical applications of Mathematics in everyday life. The proposal integrates mathematical concepts with the use of spreadsheets, using the evolution of cryptography as its background, with an emphasis on RSA cryptography. Furthermore, feasible activities for implementation in state schools in Mato Grosso are suggested, taking into account their structure and available resources, and highlighting the importance of practical application for diagnostic purposes, methodological adjustments, and the development of new approaches.

Keywords: Mathematics education; didactic proposal; RSA cryptography.

Lista de Figuras

Lista de Tabelas

Lista de siglas

A seguir, segue-se as siglas utilizadas nesta dissertação.

Profmat	Mestrado Profissional em Matemática em Rede Nacional;
UFMT	Universidade Federal de Mato Grosso;
ICET	Instituto de Ciências Exatas e da Terra;
DMAT	Departamento de Matemática.

Sumário

Introdução

Como professor de matemática na rede básica de ensino do estado de Mato Grosso, deparo-me constantemente com o seguinte dilema: estudantes imersos na cultura digital que, apesar de serem usuários frequentes da tecnologia, possuem um pensamento computacional pouco desenvolvido e, muitas vezes, são incapazes de compreender o funcionamento de um algoritmo simples. Além disso, a maioria enfrenta dificuldades em perceber a relação da matemática com o cotidiano, o que pode ser um dos fatores que compromete seu interesse e envolvimento no aprendizado.

Este trabalho busca apresentar uma alternativa para superar esses desafios, propondo uma abordagem voltada ao ensino de conceitos matemáticos na Educação Básica, oferecendo uma solução viável para os professores. A proposta baseia-se na integração de conceitos matemáticos com o uso de planilhas eletrônicas, tendo como pano de fundo a evolução da criptografia, com ênfase na criptografia RSA. A ideia é fornecer uma alternativa viável que possibilite aos docentes utilizar as ferramentas tecnológicas disponíveis nas escolas para promover uma maior interação entre os alunos e os conteúdos matemáticos. Assim, busca-se uma solução para enfrentar a falta de motivação dos estudantes, um desafio recorrente observado no exercício da docência em matemática.

Na elaboração das propostas, considerei a infraestrutura tecnológica das escolas estaduais de Mato Grosso, que, em sua maioria, oferecem suporte adequado para a implementação dessas atividades. Cada aluno tem acesso a um *Chromebook* com conexão à internet e acesso ao Google Sheets, permitindo que as atividades sejam desenvolvidas tanto no ambiente escolar quanto em casa. Essa infraestrutura possibilita que os professores explorem novas metodologias e utilizem os meios digitais como aliados no ensino da matemática.

A escolha da criptografia RSA como tema central justifica-se por sua relevância no mundo moderno e pela tentativa de conectar os elementos matemáticos que a compõem ao cotidiano dos estudantes. Além disso, a segurança digital pode atuar como um elemento

motivador para que os alunos aprofundem seus estudos por meio de aplicações práticas. Essa proposta também pode servir como ponto de partida para o desenvolvimento de habilidades em linguagens de programação mais avançadas, como Python. Assim, espera-se que essa abordagem contribua para um ensino mais significativo e alinhado com as demandas tecnológicas e educacionais contemporâneas, permitindo que os professores utilizem os recursos tecnológicos disponíveis para aprimorar o ensino da matemática na Educação Básica.

Para embasar essa abordagem, foram utilizados referenciais teóricos que abrangem tanto a fundamentação matemática quanto os aspectos históricos da criptografia. Dentre os principais autores consultados, destacam-se Singh (2022), que apresenta um panorama da evolução da criptografia; Coutinho (2007) e Hefez (2013), que exploram a teoria dos números e sua aplicação em criptografia; além das contribuições de Euler, Gauss e Fermat, fundamentais para o desenvolvimento da aritmética modular e do algoritmo RSA.

O trabalho está estruturado em cinco capítulos. O **Capítulo 1** apresenta uma contextualização histórica da evolução da criptografia, desde os métodos clássicos até os modernos, enfatizando sua importância na proteção da informação. O **Capítulo 2** aborda os fundamentos matemáticos essenciais para a compreensão do RSA, com ênfase em divisibilidade, aritmética modular, o algoritmo de Euclides e a fatoração por Fermat. O **Capítulo 3** apresenta uma explicação detalhada do funcionamento do método RSA, ilustrando seu processo de precodificação, codificação e decodificação.

O **Capítulo 4** desenvolve aplicações didáticas, no contexto do ensino básico, para explorar alguns dos conceitos matemáticos que embasam o método RSA, com atividades voltadas para facilitar a assimilação desses conteúdos. Além disso, essa abordagem faz uso parcial da metodologia da sala de aula invertida, permitindo que os alunos tenham contato prévio com os conteúdos e desenvolvam autonomia na construção do conhecimento. O **Capítulo 5** apresenta as considerações finais sobre os possíveis impactos dessa abordagem no ensino de matemática, trazendo reflexões sobre desafios, tais como a rigidez curricular, que pode dificultar a implementação dessas propostas. Além disso, destaca-se a necessidade de aplicar essa alternativa para a avaliação diagnóstica dos dados, investigando a viabilidade e os possíveis aprimoramentos.

Na **primeira parte do Apêndice**, são apresentadas sugestões de solução para os exercícios propostos ao longo do trabalho. **Nela**, são explicadas as expressões lógicas utilizadas nas planilhas, explorando as possibilidades de integração da programação em planilhas eletrônicas com os conteúdos matemáticos, oferecendo uma alternativa para ampliar o repertório de estratégias pedagógicas. Na **segunda parte do Apêndice**, são apresentadas as habilidades da Base Nacional Comum Curricular (BNCC) que se pretende

contemplar com o desenvolvimento de cada atividade.

Capítulo

1

Um Breve Histórico sobre a Importância da Criptografia

A proposta principal deste capítulo é apresentar o contexto histórico da evolução da criptografia, destacando sua relevância como uma ferramenta pedagógica para enriquecer o ensino de conteúdos matemáticos no ensino básico de Mato Grosso. Esse percurso histórico pode servir como um pano de fundo relevante para reafirmar a importância da criptografia, ao mesmo tempo em que torna o estudo da matemática mais atrativo e estimula o interesse dos estudantes.

Nesse sentido, realizaremos uma breve contextualização histórica sobre a importância da criptografia e sua evolução. Para isso, utilizaremos como referência os trabalhos de Simon Singh [11], Coutinho [1], Fisher [9] e Carneiro [15].

Ao longo da história, pode-se afirmar que, entre os acontecimentos mais significativos para a evolução humana, destacam-se os desenvolvimentos da linguagem oral e, posteriormente, da escrita. A primeira possibilitou o compartilhamento de conhecimento de maneira mais precisa e eficiente, enquanto a segunda garantiu a preservação desse conhecimento ao longo do tempo. Com o avanço da escrita e a crescente valorização das informações registradas, surgiu também a necessidade de protegê-las e controlar o acesso a elas. Como diria o notável cientista inglês Francis Bacon (1561–1626), "conhecimento é poder"(*scientia potentia est*, em latim).

A necessidade de proteger informações por meio da codificação, garantindo o acesso apenas a indivíduos autorizados, foi o principal embrião para o desenvolvimento da criptologia, que, de acordo com Sant'Ana Júnior [19]:

> é o estudo da criptografia e da criptoanálise. A criptografia é o estudo de técnicas matemáticas ou computacionais relacionadas à segurança da informação, que visam esconder e proteger as informações, enquanto a criptoanálise é o estudo de técnicas matemáticas ou computacionais que têm como objetivo acessar as informações ocultadas pela criptografia.

Inicialmente, as informações importantes codificadas não necessitavam de encriptação, visto que apenas um seleto grupo de indivíduos conhecia o significado dos símbolos que registravam essas informações. Ademais, a quantidade de pictogramas e símbolos utilizados nas mensagens representava um obstáculo adicional para sua compreensão. De acordo com Fisher [9, p. 16], "a Suméria manteve durante muitos séculos um acervo vago e ambíguo de cerca de 18 mil pictogramas e símbolos. Houve uma simplificação e padronização e, por volta de 2700-2350 a.C., com as tábuas de Shurupak, o acervo foi reduzido a aproximadamente oitocentos". Devido à grande quantidade de símbolos, a compreensão da mensagem visível exigia certo grau de conhecimento e familiaridade com eles, fazendo com que parecesse naturalmente criptografada, ainda que essa não fosse a intenção.

Figura 1.1: Amenotep, filho de Hapu, ilustre escriba egípcio,
lê um rolo de papiro parcialmente aberto. A estátua é datada do século XIV a.C. Museu Egípcio, Cairo.[9]

Com a evolução da sociedade e a consolidação da escrita, surgiu a preocupação com a segurança das mensagens, especialmente aquelas que continham informações relevantes, das quais muitas vezes dependia o futuro de uma nação. Essa questão tornava-se ainda mais crítica em tempos de guerra, quando reis, presidentes e generais confiavam nessas mensagens para garantir que suas decisões chegassem de forma segura a toda a região sob seu comando.

Nesse contexto, o conteúdo das mensagens era de importância vital, pois, se interceptadas pelo inimigo, poderiam resultar em perdas de vidas e até na queda de um reino. Para proteger suas informações e estratégias, diversos líderes começaram a adotar métodos de comunicação secreta, obtendo uma vantagem significativa no campo de batalha. Esse processo marcou o início do desenvolvimento das práticas de criptografia, que abriram caminho para o surgimento da criptologia moderna, conforme destacado por Singh [11].

> Essa busca pelo segredo levou as nações a criarem departamentos especializados na elaboração de códigos, responsáveis por garantir a segurança das comunicações por meio da invenção e do uso dos melhores métodos criptográficos. Paralelamente, decifradores de códigos inimigos empenhavam-se em quebrar esses códigos para roubar segredos. Esses decifradores eram vistos como alquimistas linguísticos, uma tribo mística que tentava dar significado a uma mistura de símbolos aparentemente sem sentido. A história dos códigos e de suas chaves é, portanto, uma batalha secular entre criadores e decifradores de códigos, uma corrida armamentista intelectual que influenciou profundamente o curso da história humana.

Um dos primeiros relatos sobre o uso de escritas secretas remonta a Heródoto, "o pai da história", conforme relatado pelo filósofo romano Cícero. Heródoto narrou os conflitos entre a Grécia e a Pérsia, ocorridos no século V a.C. Em sua visão, a utilização da escrita secreta foi determinante para salvar a Grécia de ser conquistada por Xerxes, que havia planejado a construção de Persépolis como a nova capital de seu império. Diferentemente dos estados vizinhos, Atenas e Esparta abstiveram-se de enviar qualquer tributo ou presente ao líder dos persas.

Esse ato simbolizou a independência e o compromisso com a liberdade dessas cidades, distinguindo-as das outras cidades-estado gregas que, em muitos casos, preferiam submeter-se ao domínio persa para evitar conflitos. Atenas e Esparta foram vistas como rebeldes que deveriam ser punidas, o que levou diretamente aos confrontos que culminaram nas Guerras Médicas. Xerxes, então, dedicou-se à organização de uma das maiores forças combatentes da história e, em 480 a.C., lançou um ataque surpresa contra as cidades-estado.

Entretanto, Demarato, um grego exilado que vivia na cidade persa de Susa, testemunhou os preparativos para a invasão. Apesar de estar no exílio, ele ainda mantinha lealdade à Grécia e decidiu enviar uma mensagem para alertar os espartanos sobre os planos de Xerxes. Segundo Fisher [9, p. 21], o desafio era transmitir essa mensagem de forma imperceptível, evitando que fosse interceptada pelos guardas. Heródoto descreveu:

> O perigo de ser descoberto era grande; havia apenas um modo pelo qual a mensagem poderia passar: isso foi feito raspando a cera de um par de tabuletas de madeira e escrevendo, na base, o que Xerxes pretendia fazer. Depois, a mensagem foi novamente coberta com cera. Dessa forma, as tabuletas pareciam estar em branco e não levantariam suspeitas entre os guardas ao longo da estrada. Quando a mensagem chegou ao seu destino, ninguém percebeu o segredo, até que, pelo que entendi, a filha de Cleômenes, Gorgo, que era casada com Leônidas, adivinhou e contou aos outros que, se raspassem a cera, encontrariam algo escrito na madeira. Isso foi feito, revelando a mensagem, que foi então transmitida aos outros gregos.

Segundo Singh [11], o local escolhido para a batalha oferecia uma vantagem estratégica aos gregos, cujas embarcações, menores e mais ágeis, demonstravam clara superioridade nas manobras em espaços restritos.

O que se seguiu foi um dia histórico, marcado pela humilhação dos persas. Assim que a frota persa adentrou completamente a baía, os gregos aproveitaram a oportunidade e lançaram um ataque coordenado. No espaço limitado, os navios persas, incapazes de manobrar de forma eficiente, colidiam uns com os outros em uma tentativa desesperada de escapar.

Esse tipo de comunicação secreta, responsável pela vitória grega e caracterizado pela ocultação da mensagem, é conhecido como esteganografia, termo derivado das palavras gregas *steganos*, que significa 'coberto', e *graphein*, que significa 'escrever'.

No entanto, a esteganografia possui uma vulnerabilidade crucial, pois, se a mensagem for interceptada, o mensageiro corre o risco de expor seu conteúdo, comprometendo toda a segurança da comunicação.

Para mitigar essa vulnerabilidade, surgiu, paralelamente à esteganografia, o desenvolvimento da criptografia, uma técnica cujo nome deriva da palavra grega *kryptos*, que significa "oculto". Enquanto a esteganografia buscava ocultar a existência da mensagem, a criptografia tem como principal objetivo esconder seu significado, num processo denominado encriptação. Por meio desse processo, a mensagem é transformada em um texto incompreensível, utilizando um protocolo específico de codificação. A mensagem só pode ser revelada mediante a reversão desse protocolo, que está em posse do receptor.

É importante destacar que a criptografia e a esteganografia são técnicas distintas, embora possam ser combinadas para aumentar a segurança de uma mensagem. Essa combinação permite tanto codificar quanto ocultar a mensagem simultaneamente, elevando seu nível de proteção. Um exemplo clássico dessa integração foi o uso do micropontо durante a Segunda Guerra Mundial.

Conforme Singh [11], agentes alemães desenvolveram uma técnica para reduzir fotograficamente uma página de texto até transformá-la em um ponto com menos de um milímetro de diâmetro. Esse micropontо era então colocado sobre o ponto final de uma carta aparentemente inofensiva, tornando a comunicação extremamente discreta e difícil de detectar.

No entanto, em 1941, agentes de segurança dos Estados Unidos descobriram essa técnica. A partir de então, os americanos passaram a acessar as informações contidas na maioria dos micropontos interceptados. Ainda assim, nas situações em que os agentes alemães adotavam a precaução adicional de codificar a mensagem antes de reduzi-la a um micropontо, o conteúdo permanecia inacessível, mesmo após a interceptação. Isso evidencia uma vantagem importante da criptografia sobre a esteganografia: mesmo que a mensagem seja descoberta, a codificação dificulta significativamente o acesso ao seu conteúdo.

O código gerado por um sistema de criptografia, de forma geral, baseia-se em duas premissas fundamentais: a primeira é a codificação da mensagem e a segunda, sua decodificação. Pode-se então adotar os seguintes significados técnicos, descritos por Coutinho [1, p. 1]:

> Decodificar é o que o usuário legítimo do código faz quando recebe uma mensagem codificada e deseja lê-la. Já decifrar significa ler uma mensagem codificada sem ser o usuário legítimo. Portanto, para decifrar, é preciso 'quebrar' o código.

O primeiro documento militar codificado de que se tem notícia consistia na substituição sistemática de cada letra por outra e aparece nas Guerras da Gália, de Júlio César. Em uma ocasião, ele enviou uma mensagem para Cícero, que estava cercado e prestes a se render. Para tornar a mensagem incompreensível para o inimigo, César a escreveu alterando as letras do alfabeto romano por letras gregas. Segundo Singh [11, p. 26], César descreve a dramática entrega da mensagem:

> O mensageiro recebeu instruções para que, se não pudesse se aproximar, jogasse uma lança com a mensagem amarrada por uma tira de couro dentro das fortificações do campo... Com medo, o gaulês arremessou a lança como fora instruído. Por acaso, a arma encravou-se em uma torre e permaneceu dois dias sem ser vista pelos nossos soldados, até que, no terceiro dia, um soldado a encontrou, retirou-a e entregou a mensagem a Cícero. Ele a leu e depois a recitou em voz alta para a tropa em formação, trazendo grande alegria para todos.

Outra cifra de substituição bastante utilizada foi empregada por Júlio César. Esse método consistia em substituir cada letra da mensagem original por outra que estivesse três posições à frente no alfabeto, seguindo um padrão predeterminado.

Alfabeto original:	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
Alfabeto cifrado:	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C

Tabela 1.1: Cifra de César: representação do alfabeto cifrado com deslocamento de três posições.

A mensagem criptografada apresentada a seguir é baseada na Tabela 1.1. Normalmente, na criptografia, o alfabeto original é escrito em letras minúsculas, enquanto o alfabeto cifrado é representado em letras maiúsculas.

Alfabeto original:	c	r	i	p	t	o	g	r	a	f	i	a
Alfabeto cifrado:	F	U	L	S	W	R	J	U	D	I	L	D

Tabela 1.2: Exemplo de mensagem criptografada usando a Tabela 1.1.

A cifra de substituição polialfabética pode ser considerada um modelo aprimorado da Cifra de César. Embora tenha sido reconhecida apenas no século XVI, suas origens remontam ao italiano Leon Battista Alberti, da cidade de Florença, no século XV. De acordo com Carneiro, Alberti foi uma figura de destaque na Renascença, realizando trabalhos em diversas áreas. No entanto, ele ficou mais conhecido por sua atuação na arquitetura, projetando a primeira Fonte de Trevi e escrevendo o primeiro livro sobre arquitetura.

Por volta de 1460, Alberti propôs a primeira cifra de substituição polialfabética. Contudo, ele não recebeu reconhecimento na época, já que suas ideias não estavam completamente desenvolvidas e o sistema de cifragem polialfabética ainda não havia sido formalizado. Apesar disso, suas concepções serviram de base para outros pesquisadores que posteriormente aprimoraram a técnica.

Foi então que, em 1556, Blaise de Vigenère publicou seu tratado sobre escrita secreta, *Traicté des Chiffres*. Nele, apresentou um processo para cifrar mensagens utilizando 26 alfabetos organizados em uma tabela deslocada ciclicamente, que ficaria conhecida como a Tabela de Vigenère. Para usar esse método, escolhe-se uma palavra qualquer que servirá como chave do sistema, denominada 'palavra-chave'.

	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
a	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
b	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	A
c	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
d	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
e	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
f	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
g	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
h	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
i	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
j	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
k	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
l	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
m	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
n	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
o	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
p	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
r	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
s	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
t	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
u	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
v	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
w	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
x	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
z	Z	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

Tabela 1.3: A Tabela de Vigenère.

Para enviar uma mensagem usando esse método, deve-se proceder da seguinte forma: escrever a palavra-chave sobre a mensagem, repetindo-a quantas vezes forem necessárias para que cada letra da mensagem esteja associada a uma letra da palavra-chave. A letra da mensagem cifrada será determinada pela interseção entre a letra da palavra-chave, localizada na primeira linha da tabela, e a letra da mensagem original, encontrada na primeira coluna da tabela. Esse processo é repetido para cada letra da mensagem até que toda a mensagem esteja cifrada.

Neste exemplo, apresentamos a aplicação prática da **Criptografia de Vigenère** para cifrar uma mensagem.

Exemplo 1.1 (Conversão com a Criptografia de Vigenère)**.** *Considere a mensagem original "PROFMAT NA UFMT" e a palavra-chave "FERMAT". Siga os passos a seguir para realizar a cifragem:*

1. Prepare a mensagem a ser cifrada, removendo os espaços. O resultado será: `PROFMATNAUFMT`.
2. Expanda a palavra-chave para que tenha o mesmo comprimento da mensagem. O resultado será: `FERMATFERMAT`.
3. Realize a cifragem aplicando a criptografia de Vigenère. O resultado da mensagem cifrada será: `UVFRMTYRRGFFY`.

A tabela a seguir apresenta, de forma mais intuitiva, a base para a aplicação do método de cifra de Vigenère, ao associar cada letra da mensagem original à letra correspondente da palavra-chave.

MENSAGEM ORIGINAL	P	R	O	F	M	A	T	N	A	U	F	M	T
PALAVRA-CHAVE	F	E	R	M	A	T	F	E	R	M	A	T	F
MENSAGEM CIFRADA	U	V	F	R	M	T	Y	R	R	G	F	F	Y

Tabela 1.4: Tabela de associação entre a mensagem original e a palavra-chave.

A chave utilizada na cifra de Vigenère é chamada de **chave simétrica**, pois a mesma chave usada para codificar a mensagem também é utilizada para decodificá-la. Conforme explica Singh [11], na criptografia simétrica, o processo de decifração é simplesmente o inverso do processo de cifração. No entanto, a necessidade de compartilhar a chave entre as partes envolvidas na comunicação aumenta o risco de interceptação e compromete a segurança do sistema.

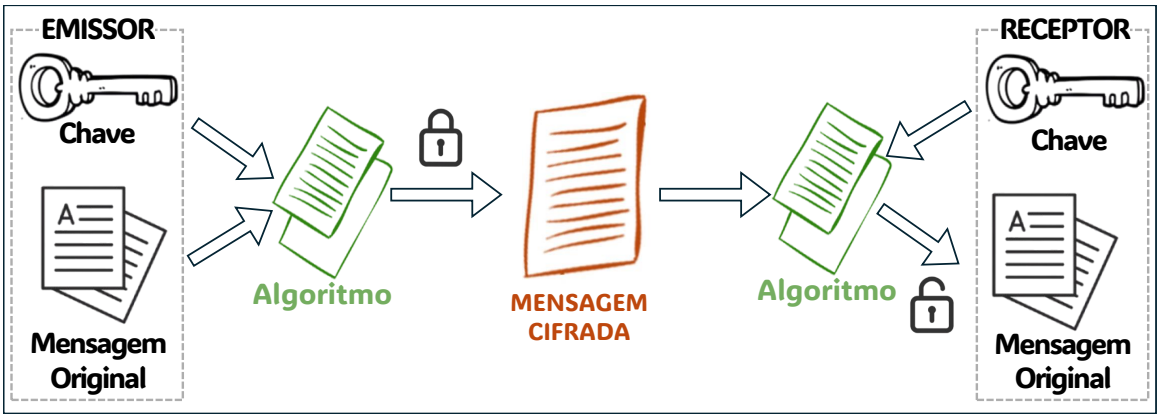

Figura 1.2: Processo de ciframento e deciframento de mensagens com chave simétrica.

Os primeiros sistemas de criptografia, como os códigos de César e Vigenère, exemplificam bem as limitações dos métodos primitivos de substituição. Esses códigos eram considerados simples e, por isso, fáceis de decifrar. Uma das razões para isso é que a frequência média com que cada letra aparece em uma língua é relativamente constante, o que facilita a aplicação de técnicas de análise de frequência para quebrar o código. Hefez [7] aponta as seguintes frequências para as letras utilizadas na língua portuguesa:

Letra	**%**	**Letra**	**%**	**Letra**	**%**	**Letra**	**%**
A	14.63	H	1.28	O	10.73	V	1.67
B	1.4	I	6.18	P	2.52	W	0.01
C	3.88	J	0.40	Q	1.20	X	0.21
D	4.99	K	0.02	R	6.53	Y	0.01
E	12.57	L	2.78	S	7,81	Z	0.47
F	1.02	M	4.74	T	4,34		
G	1.30	N	5.5	U	4,63		

Tabela 1.5: Frequência média de cada letra na língua portuguesa

- as vogais são mais frequentes que as consoantes;
- a vogal mais frequente é o A;
- se um monossílabo tem uma única letra, então esta letra é uma vogal;
- consoantes como S e M são mais frequentes que as outras.

Dessa maneira, ao contar a frequência de cada símbolo no texto, torna-se possível identificar a quais letras correspondem os símbolos mais frequentes. Esse algoritmo rudimentar é, em geral, suficiente para decifrar toda a mensagem. No entanto, sua eficácia

diminui quando a mensagem é muito curta, pois, nesse caso, a distribuição das letras pode diferir significativamente do padrão esperado para a língua, comprometendo a precisão da análise.

Um exemplo disso é a seguinte frase: **FIFA é futebol**. A letra mais frequente é o **F**, que aparece três vezes, representando 25% do total de letras, um valor significativamente maior que o indicado na Tabela (1.5). Por outro lado, a letra E aparece duas vezes, representando aproximadamente 16,667%, superando os 12,57% observados na Tabela 1.5. Além disso, as demais letras aparecem apenas uma vez, correspondendo a 8,333% cada, o que diverge dos percentuais apresentados na Tabela 1.5, tornando ineficaz o método de contagem por frequência.

Letra	**Quantidade**	**Frequência (%)**
A	1	8.33%
B	1	8.33%
E	2	16.67%
F	3	25.00%
I	1	8.33%
L	1	8.33%
O	1	8.33%
T	1	8.33%
U	1	8.33%

Tabela 1.6: Frequência de letras na frase "FIFA é futebol".

Além de ser útil para decifrar mensagens modernas, o método de contagem de frequência de caracteres tem aplicações históricas relevantes, como na decodificação de inscrições antigas. O exemplo mais conhecido é a decifração dos hieróglifos egípcios por Jean-François Champollion em 1822.

Champollion viveu em uma época conturbada na França, marcada por turbulências políticas que ameaçavam interromper sua pesquisa de diversas maneiras. Ainda assim, havia espaço para debates, entre os quais se destacava a questão sobre a natureza dos hieróglifos: ideográfica, silábica ou alfabética.

Na escrita ideográfica, cada símbolo ou "ideograma"representa uma ideia ou conceito específico, em vez de sons, como ocorre no chinês. Nas escritas silábicas, cada símbolo representa uma sílaba, isto é, uma combinação de sons que formam palavras. Um exemplo histórico de escrita silábica é o Linear B, utilizado pelos gregos antigos. Por fim, na escrita alfabética, utiliza-se um conjunto de símbolos, cada um representando um único som (fonema), que se combinam para formar palavras. Esse sistema estabelece uma relação

mais direta entre símbolo e som, como ocorre nas línguas que utilizam o alfabeto latino.

Figura 1.3: Jean-François Champollion.

Por isso, Champollion iniciou seu processo de decifração contando os caracteres nas inscrições da Pedra de Roseta. Ele sabia que, ao traduzir um texto escrito em grego (uma língua de escrita alfabética) para uma escrita ideográfica, o número de símbolos na tradução corresponderia aproximadamente ao número de palavras do texto grego.

Assim, conforme relato de Singh [11], Champollion descobriu que havia 486 palavras no texto grego e 1.419 caracteres no texto em hieróglifos. Portanto, a escrita dos antigos egípcios não podia ser exclusivamente ideográfica. Hoje, sabemos que os hieróglifos formavam um sistema misto, contendo caracteres ideográficos, silábicos e os chamados determinativos. Estes últimos servem para distinguir homônimos, indicando a que classe pertencem. Por exemplo, se escrevêssemos em português usando hieróglifos, poderíamos diferenciar a fruta "manga"da "manga"da camisa por meio de um determinativo de fruta associado à primeira.

O uso do método de contagem de frequência para decifrar uma mensagem tornou-se ainda mais eficiente com a popularização dos computadores, que dinamizaram o processo de decifração. Os computadores podem, por exemplo, calcular rapidamente a frequência dos caracteres com alta precisão, comparando-a com bancos de dados que contêm a distribuição conhecida de letras em diversas línguas. Dessa forma, a mensagem pode ser decifrada de maneira muito mais ágil. Isso inviabiliza, essencialmente, todos os códigos que envolvem substituição simples de letras.

Não por coincidência, a criação dos computadores está intimamente ligada à criptografia. Após a Primeira Guerra Mundial, o engenheiro alemão Arthur Scherbius, formado pelas universidades de Munique e Hanôver, desenvolveu uma notável máquina criptográfica chamada Enigma. Essa invenção foi considerada o sistema de cifragem mais complexo da história. Os militares alemães começaram a utilizá-la em 1926, passando a

contar com o sistema de criptografia mais seguro do mundo naquela época.

Figura 1.4: Máquina Enigma, versão da Marinha, exposta em Bletchley Park. [14]

No início da Segunda Guerra Mundial, de acordo com Singh [11], os alemães acreditavam que a máquina Enigma seria decisiva para garantir a vitória nazista. No entanto, ela acabou desempenhando um papel crucial na queda de Hitler. Determinados a decifrar as mensagens secretas nazistas, uma equipe de matemáticos trabalhava anonimamente. Entre eles, destacava-se um homem reservado chamado Alan Turing, cuja contribuição foi fundamental para o sucesso dessa missão.

Turing atuava como professor de matemática na Universidade de Cambridge, onde desenvolvia suas pesquisas. Em novembro de 1936, submeteu um artigo ao periódico *Proceedings of the London Mathematical Society*, que foi publicado no ano seguinte. Esse trabalho, intitulado "*On Computable Numbers, with an Application to the Entscheidungsproblem*"(Sobre Números Computáveis, com uma Aplicação ao Problema da Decisão), é considerado sua contribuição mais significativa. Segundo Turing [13, p.1]:

> "The "computable" numbers may be described briefly as the real numbers whose expressions as a decimal are calculable by finite means. According to my definition, a number is computable if its decimal can be written down by a machine."[1]

Nesse artigo, Turing introduziu o conceito de máquinas de Turing, um modelo abstrato que se tornou a base teórica da ciência da computação moderna, revolucionando o

[1]Tradução: Os números "computáveis"podem ser descritos, de forma breve, como números reais cujas expressões decimais podem ser calculadas por meios finitos. De acordo com minha definição, um número é computável se sua expansão decimal puder ser gerada por uma máquina.

entendimento dos limites do cálculo e da computação. Nele, ele descreve uma máquina imaginária capaz de executar, de forma automática, os processos geralmente realizados por um matemático.

A princípio, a proposta era que haveria uma máquina para cada processo: uma para somar, outra para dividir, outra para elevar ao quadrado e assim por diante. Essas máquinas ficaram conhecidas como máquinas de Turing. Posteriormente, a ideia evoluiu para que uma única máquina pudesse realizar todas as operações. Assim, surgiu a máquina universal de Turing, que serviu de base para os primeiros computadores.

Em 1939, Turing interrompeu sua carreira acadêmica para atuar como criptoanalista, integrando a Escola de Cifras e Códigos do Governo Britânico, localizada na mansão de Bletchley Park. O principal objetivo dessa instituição era decifrar as mensagens codificadas pela máquina alemã Enigma, uma tarefa essencial para os esforços de guerra dos Aliados.

Embora os ingleses tivessem obtido um exemplar da máquina Enigma, graças à traição de um oficial alemão, isso não era suficiente para quebrar seus códigos. A complexidade da Enigma residia na sua configuração diária, que determinava como as mensagens eram cifradas. O verdadeiro desafio não estava em possuir a máquina, mas em compreender como ajustá-la corretamente, já que as chaves de codificação permaneciam altamente protegidas pelo exército alemão.

Nesse contexto, a máquina universal de Turing, aliada ao esforço de outros pesquisadores em Bletchley Park, desempenhou um papel crucial na decifração do complexo código da Enigma. Esse feito histórico representou a superação da máquina de Scherbius e possivelmente alterou o curso da Segunda Guerra Mundial, encurtando o conflito em cerca de dois a quatro anos e salvando milhões de vidas.

Figura 1.5: Retrato fotográfico de Alan Turing.

Suas ideias visionárias, aplicadas tanto em tempos de guerra quanto em tempos de paz, não apenas moldaram profundamente a forma como concebemos e utilizamos os computadores atualmente, mas também estabeleceram as bases para áreas cruciais, como a criptografia e a inteligência artificial, consolidando-o como um dos maiores cientistas do século XX e garantindo que suas contribuições continuem a influenciar os avanços tecnológicos.

Com o avanço dos meios de telecomunicação, especialmente com a difusão de smartphones e computadores conectados à internet, novos desafios para a criptografia têm surgido. Como a interceptação de mensagens enviadas por redes de comunicação se tornou relativamente fácil, é imprescindível garantir sua codificação de forma robusta sempre que contiverem informações sensíveis. Isso inclui transações bancárias, comerciais e até mesmo compras realizadas com cartões de crédito.

Assim, tornou-se necessário desenvolver novos códigos que fossem difíceis de decifrar, mesmo com o auxílio de computadores. Esses códigos foram projetados, principalmente, para aplicações comerciais, em vez de serem destinados exclusivamente à comunicação entre espiões. Por essa razão, os sistemas modernos de criptografia utilizam, em sua maioria, **chaves assimétricas**, um conceito que será explorado com mais detalhes adiante.

Na vanguarda do desenvolvimento dessa ideia, destacou-se o criptógrafo Whitfield Diffie, formado em matemática pelo Instituto de Tecnologia de Massachusetts (MIT). Ao longo de sua carreira, Diffie trabalhou em diversas empresas especializadas em segurança de computadores, consolidando sua reputação como um dos maiores especialistas na área. Singh [11] comenta ainda que, compartilhando o mesmo interesse, Martin Hellman, professor da Universidade de Stanford, na Califórnia, uniu forças com Diffie para explorar novas possibilidades na criptografia.

Com o tempo, outro pesquisador que também se interessou pelo tema foi Ralph Merkle, bacharel em matemática pela Universidade da Califórnia. Merkle passou a desenvolver suas pesquisas no mesmo campo, contribuindo significativamente para o avanço dos estudos. O trabalho conjunto desses três cientistas focava na investigação de funções que não fossem de "mão dupla", ou seja, funções que fossem fáceis de executar, mas extremamente difíceis de reverter sem informações específicas — um conceito fundamental para a criptografia moderna.

Nesse contexto, as funções de "mão única"se mostraram especialmente promissoras, pois, embora sejam simples de calcular, apresentam alta complexidade para serem revertidas. Com base nessa abordagem, o grupo propôs, ainda que de forma teórica, o conceito revolucionário de **cifra assimétrica**, que, diferentemente da cifra simétrica,

utiliza uma chave para codificar, denominada **chave pública**, e outra chave distinta para decifrar, chamada **chave privada**.

Apesar do desenvolvimento de um novo conceito, o grupo enfrentava uma frustração crescente, pois, a cada mês que se passava, tornava-se mais provável que as funções de mão única talvez não existissem. Isso significava que a ideia poderia funcionar na teoria, mas não na prática.

No entanto, a corrida para encontrar uma cifra assimétrica foi vencida por outro trio de cientistas na costa leste dos Estados Unidos: Ron Rivest, Adi Shamir e Leonard Adleman. Os dois primeiros eram cientistas da computação, enquanto o último era matemático, todos atuando como pesquisadores no Laboratório de Ciências da Computação do Instituto de Tecnologia de Massachusetts (MIT).

A principal função de Adleman era identificar falhas nas ideias de Rivest e Shamir, garantindo que eles não desperdiçassem tempo explorando caminhos equivocados. Sempre que novas ideias eram apresentadas, cabia a Adleman, um matemático reconhecido por seu rigor, apontar quaisquer inconsistências ou problemas no raciocínio. Esse trabalho era essencial para assegurar que a dupla concentrasse seus esforços nas abordagens mais promissoras, evitando perder tempo com pistas falsas. Dessa forma, Adleman analisava e refutava, uma a uma, as teses propostas.

Em abril de 1977, no entanto, Rivest apresentou uma nova proposta para a função de mão única, submetendo-a à análise rigorosa de Adleman. Mais uma vez, ele seguiu seu processo formal para tentar encontrar falhas, mas, desta vez, não conseguiu refutá-la. Após um ano de colaboração intensa entre Adleman, Rivest e Shamir, eles alcançaram a descoberta mais importante da criptografia moderna: surgia o sistema de criptografia assimétrico **RSA**, cujo nome homenageia os pesquisadores responsáveis pelo seu desenvolvimento — Rivest, Shamir e Adleman. Essa descoberta marcou um divisor de águas na história da segurança da informação.

A função unidirecional, ou de mão única, que possibilita a cifra assimétrica do sistema RSA baseia-se na utilização da **aritmética modular**. Essa função pode ser usada para cifrar uma mensagem, essencialmente representada como um número. Quando esse número é inserido na função, o resultado obtido é um texto cifrado, também expresso como um número.

Um dos parâmetros mais importantes dessa função, desenvolvida por Rivest, será representado aqui como n, um número **semiprimo** (isto é, o produto de *dois números primos*). Esse parâmetro é flexível, permitindo que cada pessoa selecione um valor distinto para n, escolhendo pares únicos de primos e, dessa forma, personalizando sua própria função unidirecional.

Por exemplo, suponhamos que os valores escolhidos sejam $p = 17.159$ e $q = 10.247$. Multiplicando esses dois números, obtém-se $n = 17.159 \times 10.247 = 175.828.273$. Esse valor de n, resultante da composição dos primos p e q, torna-se a base da **chave pública** de cifragem. Essa chave pode ser divulgada livremente: pode-se imprimi-la, publicá-la na internet ou registrá-la em diretórios de chaves públicas, juntamente com os valores de n de outras pessoas.

Para facilitar a compreensão, suponhamos que alguém deseje enviar uma mensagem cifrada para Alice. Nesse caso, o remetente utiliza a **chave pública** de Alice, de conhecimento geral, para cifrar a mensagem e transmiti-la. Apenas Alice, que possui a **chave privada** correspondente, será capaz de decifrá-la. Mais adiante, na Observação 2.25, verificaremos como essas chaves se relacionam matematicamente e como ambas são compostas por um par de números, com n como elemento comum.

A mensagem cifrada é segura, e apenas Alice pode decifrá-la com sua **chave privada**. Contudo, a decifração exige acesso a informações específicas: os números primos p e q, cuja multiplicação resulta em n. Mesmo que $n = 175.828.273$ seja conhecido, os valores de p e q permanecem ocultos, garantindo a segurança do sistema.

Um questionamento comum seria: se a chave pública consiste no produto de dois números primos, outras pessoas não poderiam deduzir p e q, comprometendo a segurança do sistema? Em teoria, isso seria possível, já que n é um semiprimo constituído a partir de p e q.

Contudo, na prática, se n for suficientemente grande, calcular p e q a partir de n torna-se um problema computacionalmente inviável. Essa dificuldade prática é o elemento central que garante a segurança do **RSA** e representa um dos aspectos mais elegantes e sofisticados da criptografia assimétrica. Na verdade, verificaremos na Seção 2.4 que não basta que os primos sejam números grandes; eles também não podem ser próximos entre si, pois isso facilitaria a fatoração de n.

Nesse sentido, no próximo capítulo, exploraremos alguns conceitos matemáticos que podem ser aplicados à concepção e validação do sistema RSA. Para isso, grande parte do que precisamos compreender será encontrada nos métodos da teoria dos números, desenvolvidos tanto pelos gregos antigos quanto por matemáticos como Fermat, Euler e Gauss. Desse modo, o próximo capítulo será dedicado à abordagem desses fundamentos.

Capítulo

2

Fundamentos da Teoria dos Números na Criptografia RSA

A segurança do sistema **RSA** está fundamentada em conceitos sólidos da teoria dos números, desenvolvidos ao longo dos séculos por matemáticos notáveis, como Fermat, Euler e Gauss. Este capítulo tem como objetivo explorar esses fundamentos, incluindo tópicos essenciais, como a aritmética modular, os números primos e a fatoração. Além disso, revisaremos conceitos básicos, como a divisão de inteiros e o cálculo do máximo divisor comum, que são indispensáveis para compreender a base matemática que sustenta a robustez da criptografia moderna.

É importante destacar que, mesmo que não seja possível desenvolver todo esse conteúdo diretamente com os alunos, é fundamental que os professores compreendam a relação entre a matemática e os sistemas de criptografia. Esse conhecimento permite ao professor contextualizar os conceitos matemáticos de maneira significativa, trazendo exemplos práticos e históricos que podem despertar o interesse dos estudantes.

Os conceitos apresentados neste capítulo são fortemente inspirados em ideias desenvolvidas por renomados autores, como Hefez [7], Coutinho [1] e Domingues [6]. Suas contribuições não apenas enriquecem o entendimento matemático necessário para o estudo da criptografia, mas também orientam as demonstrações e observações que serão discutidas ao longo deste texto.

2.1 Princípio da Boa Ordenação e Divisibilidade

O Princípio da Boa Ordenação estabelece que todo conjunto não vazio de números naturais possui um menor elemento. Em outras palavras, dado um conjunto de números naturais não vazio, sempre há um número natural que é o menor entre eles. Esse princípio

é fundamental para a demonstração do Algoritmo da Divisão e do Algoritmo Euclidiano, os quais serão abordados na Seção 2.2.

Princípio da Boa Ordenação (PBO)

Se $S \subset \mathbb{N}$ é um subconjunto não vazio, então existe um elemento $s_0 \in S$ tal que $s_0 \leq s$, para todo $s \in S$.

Por exemplo, para o conjunto S, cujos elementos são $a_1 = 8$, $a_2 = 5$, $a_3 = 7$, $a_4 = 3$, $a_5 = 6$ e $a_6 = 2$, temos que $a_6 = 2$ é o menor elemento deste conjunto.

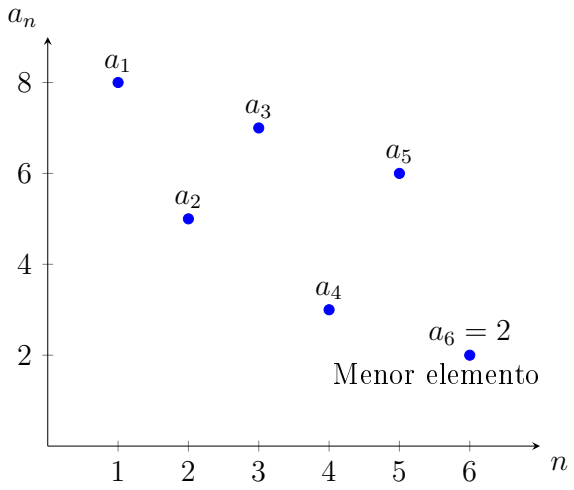

Figura 2.1: Representação gráfica do PBO

Por outro lado, para os conjuntos

$$S_1 = \{2k : \ k = 1, 2, \cdots\} \quad \text{e} \quad S_2 = \{2k + 1 : \ k = 1, 2, \cdots\},$$

temos que 2 e 3 são os menores elementos de S_1 e S_2, respectivamente, pois

$$2 \leq 2k \quad \text{e} \quad 3 \leq 2k + 1, \quad \text{para todo } k = 1, 2, \cdots$$

A seguir, serão explorados conceitos essenciais sobre divisibilidade, que constituem a base para diversos resultados importantes.

Definição 2.1. *Sejam a, $b \in \mathbb{Z}$, com $a \neq 0$. Dizemos que a divide b, ou que b é um múltiplo de a, se existir $c \in \mathbb{Z}$ tal que $b = ac$. Nesse caso, utilizamos a notação $a \mid b$.*

Por exemplo, para todo $a \in \mathbb{Z}$ com $a \neq 0$, temos $a \mid 0$, pois existe $0 \in \mathbb{Z}$ tal que $0 = a \cdot 0$. Além disso, $a \mid a$, pois há um $1 \in \mathbb{Z}$ que satisfaz $a = a \cdot 1$. Observe ainda que

$0 \mid a$ se e somente se $a = 0$. De fato, suponha que $0 \mid a$. Então, existe $c \in \mathbb{Z}$ tal que $a = c \cdot 0$. Como $c \cdot 0 = 0$, concluímos que $a = 0$. A recíproca decorre do fato de que $0 \mid 0$ é um caso particular de $a \mid a$.

Exemplo 2.1. *Sejam a, b e $c \in \mathbb{Z}$, com $a \neq 0$. Mostre que, se $a \mid b$, então $a \mid bc$.*

Solução.
Usando a definição de divisibilidade, se $a \mid b$, então existe $q \in \mathbb{Z}$ tal que $b = aq$. Multiplicando ambos os lados dessa igualdade por c, temos:

$$bc = (aq)c = a(qc).$$

Como $qc \in \mathbb{Z}$, segue, pela definição de divisibilidade, que $a \mid bc$. □

Observação 2.2. *Hefez [7] ressalta que a notação $a \mid b$ não corresponde a nenhuma operação em $\mathbb{Z}$, tampouco representa uma fração. Trata-se de uma declaração que afirma a existência de um $c \in \mathbb{Z}$ tal que $b = ac$. Caso esse número c não exista, dizemos que a não divide b e utilizamos a notação $a \nmid b$.*

A seguir, verificaremos algumas propriedades relacionadas à divisibilidade.

Teorema 2.3. *Sejam a, b, $c \in \mathbb{Z}$. Então:*

I. *Se $a \mid b$ e $b \mid c$, então $a \mid c$.*

II. *Se $a \mid b$ e $a \mid c$, então $a \mid (b + c)$ e $a \mid (b - c)$.*

III. *Se $b \neq 0$ e $a \mid b$, então $|a| \leq |b|$.*

Demonstração.
A demonstração dos items I e II podem ser encontrados em Hefez [7]. Aqui, apresentaremos apenas a demonstração do item III. De fato, se $a \mid b$, então existe $q \in \mathbb{Z}$ tal que $b = aq$. Como $b \neq 0$,

$$0 \neq |b| = |a|\ |q|.$$

Daí, $|q| \neq 0$ e, portanto, $|q| \geq 1$, pois $q \in \mathbb{Z}$. Consequentemente,

$$|a| \leq |a|\ |q| = |b|.$$

□

Por exemplo, como $2 \mid 4$ e $2 \mid 12$, segue do Teorema 2.3, item II, que $2 \mid 14$ e $2 \mid 8$, o que pode ser facilmente verificado.

2.2 O Algoritmo da Divisão e o Algoritmo Euclidiano

A seguir, estudaremos dois algoritmos de extrema importância: o **Algoritmo da Divisão** e o **Algoritmo Euclidiano**. Coutinho [1] destaca que:

> "Ambos eram conhecidos na Grécia Antiga e são descritos nos *Elementos* de Euclides, escrito por volta de 300 a.C. No primeiro, o que nos interessa é calcular o quociente e o resto da divisão de um número inteiro por outro. Já o algoritmo euclidiano é usado para determinar o cálculo do máximo divisor comum."

Antes de prosseguirmos, é necessário compreender o conceito de algoritmo. O dicionário Aurélio define a palavra algoritmo como: o processo de cálculo, ou de resolução de um grupo de problemas semelhantes, em que se estipulam, com generalidade e sem restrições, regras formais para a obtenção de um resultado ou da solução de um problema. De forma geral, pode-se pensar em um algoritmo como um passo a passo para resolver um determinado tipo de problema. Nesse sentido, uma definição mais abrangente e precisa da ideia de algoritmo é apresentada por Ziviani [12]:

> Os algoritmos fazem parte do dia a dia das pessoas. As instruções para o uso de medicamentos, as indicações de como montar um aparelho qualquer, uma receita de culinária, são alguns exemplos de algoritmos. Um algoritmo pode ser visto como uma sequência de ações executáveis para a obtenção de uma solução para um determinado tipo de problema. Segundo Dijkstra (1971), um algoritmo corresponde a uma descrição de um padrão de comportamento, expresso em termos de um conjunto finito de ações. Ao executarmos a operação $a + b$, percebemos um mesmo padrão de comportamento, mesmo que a operação seja realizada para valores diferentes de a e b.

2.2.1 Algoritmo da Divisão

Segundo Hefez [7], mesmo quando um número natural a não divide exatamente o número natural b, Euclides, em sua obra *Elementos*, parte do princípio de que é sempre possível realizar a divisão de b por a. Esse processo consiste em encontrar um quociente q e um resto r, em que r é um número natural menor que a, de modo que a relação $b = aq+r$ seja satisfeita. Embora Euclides não tenha enunciado explicitamente essa propriedade, ela está implícita e constitui a base do algoritmo da divisão, um conceito fundamental na teoria dos números.

Teorema 2.4 (**Algoritmo da Divisão**). *Sejam b e $a \in \mathbb{Z}$ tais que $b \geq 0$ e $a > 0$. Então, existem inteiros únicos q e r tais que: $b = aq + r$, com $0 \leq r < a$.*

Demonstração.
Inicialmente, provaremos que tais q e r existem. Considere o seguinte conjunto $\mathbb{S}$ definido como:

$$\mathbb{S} = \{b - ak : k \in \mathbb{Z} \text{ e } b - ak \geq 0\}. \tag{2.1}$$

A definição do conjunto $\mathbb{S}$, embora sutil, será de grande ajuda, pois ele representa todos os possíveis resíduos $b - ak$ que surgem ao subtrair múltiplos de a (representados por ak) de b.

Sendo $\mathbb{S} \neq \emptyset$, pois $b - a(0) = b \in \mathbb{S}$, segue do PBO que $\mathbb{S}$ possui um menor elemento que denominaremos por r. Assim, de (2.1), existe $q \in \mathbb{Z}$ tal que

$$r = b - aq \qquad \text{e} \qquad r \geq 0.$$

Para mostrar que $r < a$ argumentaremos por contradição. Suponha, então, que $r \geq a$. Segue daí:

$$b - a(q + 1) = b - aq - a = r - a \geq 0.$$

Assim, teríamos que $r - a \in \mathbb{S}$, com $r - a < r$. Mas isso contradiz o fato de que r é o menor elemento de $\mathbb{S}$. Logo, a suposição $r \geq a$ é falsa, e, portanto, $r < a$.

Assim, podemos afirmar que existem inteiros q e r tais que:

$$b = aq + r, \quad \text{com } 0 \leq r < a.$$

Para concluir a nossa demonstração, verificaremos agora que esse par de inteiros, q e r, é único. Suponha, por absurdo, que existam dois pares distintos (q, r) e (q', r') tais que:

$$b = aq + r \quad \text{e} \quad b = aq' + r',$$

com $0 \leq r < a$ e $0 \leq r' < a$. Subtraindo essas duas expressões, obtemos:

$$a(q - q') = r' - r. \tag{2.2}$$

Como r e $r' \in \mathbb{Z}$, podemos supor, sem perda de generalidade, que $r' \geq r$. Isso implica que $r' - r \geq 0$. Além disso, como $r' < a$ e $r \geq 0$, temos que $0 \leq r' - r < a$. Portanto, de (2.2),

$$0 \leq a(q - q') < a.$$

Dividindo todos os termos por $a > 0$, obtemos:

$$0 \leq q - q' < 1.$$

Como $q - q' \in \mathbb{Z}$, segue que $q - q' = 0$, ou seja, $q = q'$. Consequentemente, de (2.2), $r = r'$. Portanto, os pares (q, r) e (q', r') são iguais, o que prova que os inteiros q e r são únicos. □

Para uma melhor compreensão do teorema, apresentaremos um exemplo ilustrativo.

Exemplo 2.2. *Construa o conjunto* $\mathbb{S}$ *considerando* $b = 27$ *e* $a = 4$*, e determine os valores de* q *e* r*.*

Solução.
O conjunto $\mathbb{S}$ pode ser escrito da seguinte forma:

$$\mathbb{S} = \{27 - 4k \in \mathbb{Z} : 27 - 4k \geq 0\}.$$

Substituindo os valores de k por números inteiros, identificamos os elementos que constituem $\mathbb{S}$:

$$\mathbb{S} = \{\cdots, 27 - 4(-2),\ 27 - 4(-1),\ 27 - 4(0),\ 27 - 4(1),\ 27 - 4(2),\ 27 - 4(3), \cdots\}.$$

Simplificando, temos:

$$\mathbb{S} = \{\cdots, 35,\ 31,\ 27,\ 23,\ 19,\ 15,\ 11,\ 7,\ 3,\ -1,\ \cdots\}.$$

Esse conjunto representa os possíveis restos da divisão de 27 por 4. Entre esses valores, o único que satisfaze a condição $0 \leq r < 4$ é 3, o que ocorre quando $k = 6$. Portanto, os valores de r e q são, respectivamente 3 e 6. □

No exemplo acima, o conjunto $\mathbb{S}$ contém valores maiores que 27 ou menores que 0, resultantes da substituição de k por números inteiros positivos ou negativos. Esses valores, contudo, são todos congruentes módulo 4, pois, ao serem divididos por 4, deixam o mesmo resto, 3. A ideia de congruência modular será abordada de forma mais aprofundada nos próximos tópicos.

2.2.2 Algoritmo Euclidiano

O **Algoritmo Euclidiano** é utilizado para determinar o **Máximo Divisor Comum (MDC)** entre dois números naturais. Ele se baseia em uma sequência de divisões sucessivas, fundamentado na ideia de que o MDC de dois números permanece inalterado se o maior número for substituído pelo resto da divisão entre eles, veja (2.3).

Antes de formalizar a demonstração do Algoritmo Euclidiano, é necessário definir formalmente o conceito do MDC.

Definição 2.5. *Sejam a e $b \in \mathbb{Z}$. Dizemos que $d \in \mathbb{Z}$, com $d > 0$, é o MDC de a e b se satisfizer as seguintes propriedades:*

I. *$d \mid a$ e $d \mid b$;*

II. *d é divisível por todo divisor comum de a e b, ou seja, se $c \in \mathbb{Z}$ é tal que $c \mid a$ e $c \mid b$, então $c \mid d$.*

A Definição 2.5, de acordo com Herfez [7], é essencialmente a mesma utilizada por Euclides em sua obra ***Os Elementos*** e constitui um dos pilares de sua aritmética.

Observação 2.6. *Sejam c e d como no item II da Definição 2.5.*

1. *O MDC de a e b é o maior entre todos os divisores comuns desses números. De fato, como $c \mid d$, segue do Teorema 2.3, item III, que*

$$c \leq |c| \leq |d| = d.$$

2. *O MDC é único. De fato, se d' é outro MDC de a e b, então $d' \mid d$ e $d \mid d'$. Usando novamente o Teorema 2.3, item III,*

$$d' \leq d \quad e \quad d \leq d'.$$

   Portanto, $d' = d$.

3. *O MDC de a e $b \in \mathbb{Z}$ será denotado por (a, b).*

Exemplo 2.3. *Para a e $b \in \mathbb{Z}$, temos que:*

i) $(a, b) = (b, a)$;

ii) *Se $a \mid b$, então $(a, b) = |a|$;*

iii) $(a, 0) = |a|$;

iv) $(a, 1) = 1$.

Solução.
Discutiremos o item ii), que servirá de base para os demais. Seja $d = (a, b)$. Assim, $d \mid a$ e, pelo Teorema 2.3, item III, concluímos que $d = |d| \leq |a|$. Além disso, como $a \mid a$ e $a \mid b$, segue da Definição 2.5, item II, que $a \mid d$. Portanto, usando novamente o Teorema 2.3, item III, temos $|a| \leq |d| = d$. Consequentemente, $d = |a|$. A prova dos demais itens é análoga. □

Normalmente, nos anos iniciais do ensino fundamental, a determinação do máximo divisor comum (MDC) de dois números inteiros a e b é feita da seguinte maneira: calcula-se todos os divisores de a e, em seguida, todos os divisores de b. Dessa forma, obtêm-se dois conjuntos: o conjunto dos divisores de a e o conjunto dos divisores de b. A interseção desses dois conjuntos contém os divisores comuns de a e b, e o maior elemento dessa interseção corresponde ao MDC, denotado por (a, b).

No entanto, esse método torna-se totalmente inviável para números inteiros grandes, pois determinar todos os divisores de um número muito grande é uma tarefa extremamente trabalhosa, mesmo com o auxílio de computadores eficientes.

Felizmente, existe um método mais eficiente para calcular (a, b): o algoritmo descrito por Euclides nas Proposições 1 e 2 do **Livro 7 dos Elementos**, como observado por Coutinho [1]. Embora seja conhecido como *algoritmo euclidiano*, acredita-se que sua origem seja anterior ao próprio Euclides.

Algoritmo Euclidiano

Sejam a e $b \in \mathbb{Z}$, com $a \geq b > 0$. Para determinar (a, b), dividimos a por b. Se o resto dessa divisão for $r_1 \neq 0$, dividimos b por r_1. Novamente, se o resto dessa divisão for $r_2 \neq 0$, dividimos r_1 por r_2 e verificamos se o resto é zero. Quando isso ocorre, o último resto diferente de zero da sequência de divisões será o (a, b).

Exemplo 2.4. *Usando o algoritmo Euclidiano, calcule* $(1234, 54)$.

Solução.
O resto da divisão de 1234 por 54 é 46, e ao dividir 54 por 46, obtemos o resto 8. Continuando, dividimos 46 por 8 e obtemos 6, depois dividindo 8 por 6 obtemos um resto 2. Por fim, dividindo 6 por 2, o resto é 0. Portanto, $(1234, 54) = 2$. □

Essa sequência de operações pode ser organizada de forma otimizada, como representado na figura abaixo.

1234	54	46	8	6	2
46	8	6	2	0	

Tabela 2.1: Execução manual do Algoritmo de Euclides.

Para utilizar o algoritmo euclidiano de forma eficaz, é necessário garantir que a sequência de divisões produza, em algum momento, um resto zero. Caso contrário, o algoritmo poderia se repetir indefinidamente. Para assegurar essa propriedade, Euclides empregou o seguinte resultado, conforme apresentado por Hefez [7].

Lema 2.7 (Lema de Euclides)**.** *Sejam a, b e $n \in \mathbb{Z}$. Se $(a, b - na)$ existe, então (a, b) também existe e*

$$(a, b) = (a, b - na).$$

Em particular, se $b = aq + r$, com $0 \leq r < a$, então temos:

$$(a, b) = (a, b - aq) = (a, r), \tag{2.3}$$

Assim, o MDC de dois números permanece inalterado quando o maior deles é substituído pelo resto da divisão entre ambos.

Apresentamos agora a prova construtiva da existência do MDC, conforme descrita por Euclides em *Os Elementos* (Livro VII, Proposição 2). Como destaca Hefez [7], o método, conhecido como algoritmo de Euclides, é um marco de eficiência do ponto de vista computacional e permaneceu praticamente inalterado por mais de dois milênios.

Teorema 2.8. *Dados a e $b \in \mathbb{N}$, então existe um $d \in \mathbb{Z}$, com $d > 0$, tal que $(a, b) = d$.*

Demonstração.
Dados $a, b \in \mathbb{N}$, sem perda de generalidade, podemos supor que $a \leq b$. Do Exemplo 2.3, se $a = 1$, $a = b$ ou $a \mid b$, então $(a, b) = a$.

Suponhamos, então, que $1 < a < b$ e que $a \nmid b$. Assim, pelo Teorema 2.4, podemos escrever:

$$b = aq_1 + r_1, \quad \text{com } 0 < r_1 < a.$$

Temos duas possibilidades:

I. Se $r_1 \mid a$, a partir de (2.3) e do exemplo 2.3, temos que $(a, b) = (a, r_1) = r_1$, e, portanto, o algoritmo termina.

II. Se $r_1 \nmid a$, usando novamente o Teorema 2.4,

$$a = r_1 q_2 + r_2, \quad \text{com } 0 < r_2 < r_1.$$

Assim, novamente temos duas possibilidades:

i. Se $r_2 \mid r_1$, então, usando (2.3) e o Exemplo 2.3, obtemos:

$$(a, b) = (a, r_1) = (r_1, r_2) = r_2.$$

Logo, o algoritmo termina.

ii. Se $r_2 \nmid r_1$, do Teorema 2.4, temos:

$$r_1 = r_2 q_3 + r_3, \quad \text{com } 0 < r_3 < r_2.$$

Continuando esse procedimento, obtemos uma sequencia decrescente de restos $a > r_1 > r_2 > r_3 > \cdots > 0$. Portanto, do PBO, existe r_n tal que

$$a > r_1 > r_2 > r_3 > \cdots > r_{n-1} > r_n > 0.$$

Além disso, $r_n \mid r_{n-1}$, pois, caso contrário, do Teorema 2.4, existiria r_{n+1} tal que

$$r_{n-1} = r_n q_{n+1} + r_{n+1}, \quad \text{com } 0 < r_{n+1} < r_n.$$

o que contradiz o fato do r_n ser o menor elemento da sequencia de restos. Logo, para algum n, teremos :

$$\begin{array}{ccc}
b = aq_1 + r_1 & \text{com} & 0 < r_1 < a \\
a = r_1 q_2 + r_2 & \text{com} & 0 < r_2 < r_1 \\
r_1 = r_2 q_3 + r_3 & \text{com} & 0 < r_3 < r_2 \\
\vdots & & \vdots \\
r_{n-2} = r_{n-1} q_n + r_n & \text{com} & 0 < r_n < r_{n-1} \\
r_{n-1} = r_n q_{n+1} & \text{com} & r_{n+1} = 0
\end{array}$$

Isso implica que

$$(a, b) = (a, r_1) = (r_1, r_2) = \cdots = (r_{n-1}, r_n) = r_n.$$

□

Conforme Coutinho [1], O Algoritmo Euclidiano pode ser adaptado para expressar (a, b) como uma combinação linear de a e b. Esse procedimento segue os passos do algoritmo formalizado por Donald E. Knuth, descrito a seguir:

Algoritmo Euclidiano Estendido

Inicialmente, defina os valores $r_{-1} = b$, $r_0 = a$, $x_{-1} = 1$, $y_{-1} = 0$, $x_0 = 0$ e $y_0 = 1$. A partir de $j = 1, 2, \cdots$, para cada iteração, calcule:

- r_j e q_j, que representam o resto e o quociente, respectivamente, obtidos na divisão de r_{j-2} por r_{j-1};
- $x_j = x_{j-2} - q_j x_{j-1}$ e $y_j = y_{j-2} - q_j y_{j-1}$.

O processo se encerra quando, para algum n, ocorre $r_{n+1} = 0$. Nesse caso, $(a, b) = r_n$ e

$$(a, b) = x_n b + y_n a.$$

Esse algoritmo possui diversas aplicações, destacando-se na resolução de problemas relacionados à aritmética modular e à teoria dos números inteiros.

Exemplo 2.5. *Calcule* $(1234, 54)$ *e expresse esse valor como uma combinação linear de* 1234 *e* 54.

Solução.

Aplicando o Algoritmo de Euclides Estendido com $x_{-1} = 1234$ e $x_0 = 54$, obtemos:

j	r_j	q_j	x_j	y_j
-1	1234	—	1	0
0	54	—	0	1
1	46	22	$1 - 22 \times 0 = 1$	$0 - 22 \times 1 = -22$
2	8	1	$0 - 1 \times 1 = -1$	$1 - 1 \times (-22) = 23$
3	6	5	$1 - 5 \times (-1) = 6$	$-22 - 5 \times 23 = -137$
4	2	1	$-1 - 1 \times 6 = -7$	$23 - 1 \times (-137) = 160$
5	0	3	$*$	$*$

Tabela 2.2: Execução do Algoritmo Euclidiano Estendido para 1234 e 54

□

Um detalhe importante a ser mencionado é que, diferentemente do algoritmo da divisão, os valores α e β obtidos pelo Algoritmo Euclidiano Estendido, que satisfazem $(a, b) = \alpha b + \beta a$ não são únicos. De fato, para $k \in \mathbb{Z}$,

$$(\alpha + kb)a + (\beta - ka)b = (a, b).$$

Mas esse fato nos leva a outra pergunta: Se os valores de α e β não são únicos, por que o desenvolvimento do algoritmo de Knuth para determinar esses valores é tão importante? Na verdade, muitos dos resultados que constituem o método de criptografia RSA, como comenta Coutinho [1], não seriam possíveis se não tivéssemos uma maneira eficiente de calcular α e β. Um exemplo disso é a unicidade da fatoração de um número inteiro em produto de números primos.

2.3 Números Primos

Hefez [7] descreve o estudo dos números primos como um dos conceitos mais importantes de toda a matemática, destacando o papel fundamental desses números no desenvolvimento da disciplina. Além disso, os números primos estão associados a diversos problemas famosos, cujas soluções têm desafiado gerações de matemáticos.

Definição 2.9. *Um número natural maior do que 1 que possui apenas 1 e ele próprio como divisores positivos é chamado de número primo. Por sua vez, um número natural maior do que 1 que não é primo é chamado de número composto.*

Por exemplo, 13, 17 e 19 são números primos, enquanto que 18, 20 e 22 são compostos.

Observação 2.10. *Considerando a definição acima e tomando dois números primos quaisquer, p e q, e um inteiro a qualquer, têm-se os seguintes fatos:*

i. *Se $p \mid q$, então $p = q$. De fato, como $p \mid q$ e q é primo, temos que $p = 1$ ou $p = q$. Como p é primo, tem-se $p > 1$, logo $p = q$.*

ii. *Se $p \nmid a$, então $(p, a) = 1$. De fato, fazendo $(p, a) = d$, tem-se que $d \mid p$. Portanto, $d = p$ ou $d = 1$. Mas $d \neq p$, pois $p \nmid a$, e, consequentemente, $d = 1$.*

iii. *Se $a \geq 2$ e c é o menor divisor de a maior que 1, então c é um número primo. De fato, seja $k \in \mathbb{Z}$ tal que $k \mid c$. Pelo Teorema 2.3, item III, temos que $k \leq c$. Além disso, existe $d \in \mathbb{Z}$ tal que $c = kd$. Por outro lado, como $c \mid a$, existe $f \in \mathbb{Z}$ tal que $a = fc$. Assim,*

$$a = fkd = (fd)k.$$

Logo, $k \mid a$. Como c é o menor dos divisores de a maior que 1, segue que $c \leq k$. Consequentemente $c = k$, Isso significa que o único divisor de c, maior que 1, é ele próprio. Portanto, c é um número primo.

Conforme ressalta Hefez [7], o próximo lema apresenta um resultado fundamental de Euclides (em *Os Elementos*, Proposição 30, Livro VII):

Lema 2.11. *Sejam $a, b \in \mathbb{Z}$ e seja p um número primo. Se $p \mid ab$, então $p \mid a$ ou $p \mid b$.*

Demonstração.
Mostraremos inicialmente que, se $p \mid ab$ e $p \nmid a$, então $p \mid b$. De fato, observe que existe $\theta \in \mathbb{Z}$ tal que $ab = p\theta$. Além disso, pela Observação 2.10, item ii, temos que $(p, a) = 1$. Portanto, do Algoritmo Euclidiano Estendido, existem $\alpha, \beta \in \mathbb{Z}$ tais que $\alpha a + \beta p = 1$. Daí que,

$$\begin{aligned} b &= \alpha(ab) + \beta(pb) = \alpha(p\theta) + \beta(pb) \\ &= p(\alpha\theta + \beta b). \end{aligned}$$

Logo, $p \mid b$, conforme desejado. De forma análoga, podemos mostrar que, se $p \mid ab$ e $p \nmid b$, então $p \mid a$. □

Como consequência deste lema temos o seguinte resultado:

Proposição 2.12. *Para números primos distintos p e q, e $a \in \mathbb{Z}$ tais que $p \mid a$ e $q \mid a$, então $pq \mid a$.*

Solução.
Por hipótese, existem $m, n \in \mathbb{Z}$ tais que $a = mp$ e $a = nq$. Então, $mp = nq$, logo $q \mid mp$. Como $q \nmid p$, pois p e q são primos, segue do Lema 2.11 que $q \mid m$. Portanto, existe $k \in \mathbb{Z}$ tal que $m = kq$. Consequentemente,

$$a = mp = kqp \quad \Longrightarrow \quad pq \mid a.$$

□

Usando os conceitos matemáticos desenvolvidos até aqui, analisaremos agora uma das ideias mais importantes relacionadas aos números primos: o Teorema da Fatoração Única. Este teorema é tão significativo que é frequentemente chamado de Teorema Fundamental da Aritmética.

Coutinho [1] comenta que Carl Friedrich Gauss foi o primeiro matemático a enunciar sistematicamente esse teorema no §16 de sua obra clássica *Disquisitiones Arithmeticae*. Contudo, é relevante observar que esse resultado já era conhecido e aplicado desde a Grécia Antiga. A principal contribuição de Gauss foi formalizar o teorema dentro de um sistema lógico rigoroso, integrando-o de maneira estruturada e moderna à teoria dos números.

Teorema 2.13 (Teorema Fundamental da Aritmética)**.** *Dado $n \in \mathbb{Z}$, com $n \geq 2$, ele pode*

ser representado, de modo único, na forma:

$$n = p_1^{e_1} \cdot p_2^{e_2} \cdot \cdots \cdot p_k^{e_k},$$

onde $p_1 < p_2 < \cdots < p_k$ *são números primos e* $e_1, e_2, \ldots, e_k$ *são números inteiros positivos.*

Demonstração.
Iniciaremos demonstrando a existência da fatoração. Seja p_1 o menor divisor de n maior que 1. De acordo com a Observação 2.10, item *iii*, p_1 é um número primo. Além disso, existe $f_1 \in \mathbb{Z}$ tal que $n = p_1 f_1$, com $1 \leq f_1 < n$.

Se $f_1 = 1$, então $n = p_1$, e a fatoração desejada está concluída. Caso contrário, temos $1 < f_1 < n$. Seja p_2 o menor divisor de f_1 maior que 1. Mais uma vez, pela Observação 2.10, item *iii*, p_2 é primo. Além disso, existe $f_2 \in \mathbb{Z}$ tal que $n = p_1 p_2 f_2$, com $1 \leq f_2 < f_1$.

Se $f_2 = 1$, segue que $n = p_1 p_2$, e a fatoração está completa. Caso contrário, temos $1 < f_2 < f_1 < n$.

Prosseguindo com esse processo, obtemos: $1 \leq \cdots < f_{i+1} < f_i < \cdots < f_2 < f_1 < n$. Portanto, do PBO, essa sequência possuiu um menor elemento f_j, o qual é igual a 1. Pois caso contrario, existiria $f_{j+1} \in \mathbb{Z}$ tal que $n = p_1 p_2 \cdots p_j p_{j+1} f_{j+1}$, com $1 \leq f_{j+1} < f_j$, o que contradiz o fato do f_j ser o menor elemento da sequencia.

Assim, para algum j, temos:

$$n = p_1 p_2 p_3 \cdots p_j.$$

Observe que, ao longo do processo, os p_i podem se repetir. Assim, ao agrupar os fatores primos repetidos, se necessário, e organizar os p_i em ordem crescente, obtemos a representação de n conforme descrita no teorema.

Agora, provaremos a unicidade da fatoração utilizando indução em relação a n.

- **Base da Indução:** Para $n = 2$, não há o que demonstrar, visto que 2 é um número primo e sua fatoração é única.

- **Hipótese de Indução:** Suponha que o teorema seja válido para todo número inteiro $n \leq k$, ou seja, é válido para $n = 2, 3, \ldots, k$.

- **Tese de Indução:** Se $n = k + 1$ é um número primo, não há o que se fazer. Suponha, então, que n seja um número composto e admita duas decomposições distintas em fatores primos:

$$p_1 p_2 p_3 \cdots p_j = n = q_1 q_2 q_3 \cdots q_s.$$

Dessa igualdade, temos que $p_1 \mid q_1q_2q_3\cdots q_s$. Assim, $p_1 \mid q_r$ para algum $r \in \{1, 2, \cdots, s\}$. Pela Observação 2.10, item i, temos que $p_1 = q_r$. Sem perda de generalidade, podemos assumir que $r = 1$, ou seja, $p_1 = q_1$. Portanto,

$$p_2 \cdot p_3 \cdots p_j = q_2 \cdot q_3 \cdots q_s$$

Como $n/p_1 = p_2 \cdots p_j$ é menor que n, segue da hipótese de indução que essa fatoração é única. Logo, $j = s$ e, portanto, cada $p_i = q_k$.

□

A fatoração de um número composto em produtos de números primos permite demonstrar um fato extremamente importante para a concepção da função de mão única no método de criptografia RSA: a existência de infinitos números primos. Embora esse fato possa parecer intuitivo à primeira vista, ele pode ser demonstrado de forma rigorosa.

A demonstração que apresentaremos aqui é a que consta nos *Elementos* de Euclides, como Proposição 20 do Livro IX. Conforme observa Hefez [7], ela é considerada uma das joias da matemática, pois marca a primeira aplicação registrada do método de redução ao absurdo em um argumento matemático.

Teorema 2.14. *Existem infinitos números primos.*

Demonstração.

Como mencionado anteriormente, faremos a demonstração por redução ao absurdo. Suponha, então, que exista uma quantidade finita de números primos, dados por $p_1, p_2, p_3, \ldots, p_r$. Considere agora $m \in \mathbb{N}$ definido como

$$m = p_1p_2p_3\cdots p_r + 1.$$

Do Teorema 2.13, algum dos p_i deve dividir m, pois, por hipótese, estes são os únicos números primos que existem. Assim, como $p_i \mid m$ e $p_i \mid p_1p_2p_3\cdots p_r$, segue que

$$p_i \mid (m - p_1p_2p_3\cdots p_r) = 1.$$

Isso é um absurdo, pois nenhum número primo divide 1. □

Cabe aqui comentar, de forma bastante sintética, alguns métodos para encontrar a fatoração de números. Nesse sentido, a maneira mais comum de realizar essa tarefa é utilizar um algoritmo introduzido nos anos iniciais do ensino fundamental: dado um

número inteiro $n \geq 2$, tenta-se dividi-lo por cada um dos inteiros de 2 até $n-1$. Se algum desses números for um divisor de n, então ele será um fator de n e, como vimos no item (iii) da Observação 2.10, o menor desses fatores será um número primo.

Muitas vezes, para otimizar esse processo, recomenda-se escolher apenas números primos para a divisão. Além disso, outras melhorias podem ser implementadas para aperfeiçoar o procedimento de fatoração. Nesse sentido, Coutinho [1] faz a seguinte observação.

O algoritmo de fatoração de n começa em 2 e se estende até $n-1$. Entretanto, não é necessário verificar fatores maiores que $\sqrt{n}$, pois um número inteiro não pode ter um fator superior a si próprio.

Em essência, esse algoritmo busca o menor fator f_0 do número $n > 1$. Assim, se n for composto e $f_0 > 1$ for o seu menor fator, então $f_0 \leq \sqrt{n}$. Caso n seja primo, temos $f_0 = n$. Para demonstrar, seja n composto e $f_0 > 1$ seu menor fator. Então, existe um inteiro a tal que $n = f_0 \cdot a$. Como f_0 é o menor fator, resulta em $f_0 \leq a$. Logo, $f_0 \leq \frac{n}{f_0}$, e portanto, $f_0^2 \leq n$, o que equivale a $f_0 \leq \sqrt{n}$.

Para exemplificar, considere n um número primo com 100 ou mais algarismos, ou seja, $n \geq 10^{100}$. Logo, $\sqrt{n} \geq 10^{50}$. Isso implica que seriam necessárias ao menos 10^{50} divisões para confirmar a primalidade de n usando esse algoritmo de fatoração.

Coutinho [1] comenta a dificuldade de fatorar números tão grandes, enfatizando que não é viável recorrer a métodos convencionais. Ao converter esse cálculo em tempo, supondo que um computador executa 10^{10} divisões por segundo, ele levaria $\frac{10^{50}}{10^{10}} = 10^{40}$ segundos para determinar a primalidade de n. Isso corresponde a aproximadamente 10^{31} anos, demonstrando a inviabilidade prática desse algoritmo para números com 100 ou mais algarismos.

A demonstração completa do método pode ser encontrada em Coutinho [1]. Por outro lado, o exemplo abordado aqui está completo.

Nesse contexto, o algoritmo mostra-se útil para fatorar números menores, da ordem de 10^6, em que o fator f_0 pode ser encontrado rapidamente. No entanto, para valores maiores, sua eficiência é limitada, como vimos. Por outro lado, o algoritmo apresentado por Fermat é mais eficiente quando o número possui um fator primo p próximo de $\sqrt{n}$. A seguir, descreveremos esse método, cuja demonstração completa pode ser encontrada em Coutinho [1], e apresentaremos um exemplo detalhando o passo a passo de sua execução.

2.4 Fatoração por Fermat

A ideia do algoritmo de Fermat consiste em encontrar dois inteiros positivos x e y tais que $n = x^2 - y^2$. Note que n deve ser ímpar, caso contrário, seria divisível por 2. Uma vez encontrados esses números, n pode ser escrito como:

$$n = x^2 - y^2 = (x + y)(x - y). \tag{2.4}$$

Dessa forma, $(x+y)$ e $(x-y)$ serão fatores de n. Para utilizar o algoritmo, é necessário calcular $\sqrt{n}$. No entanto, o que nos interessa é a parte inteira desse número real. Seguindo a notação apresentada por Coutinho [1], denotaremos a parte inteira como $[r]$. Por exemplo, $[\sqrt{17}] = 4$ e $[\pi] = 3$. Note que, se $r \in \mathbb{Z}$, então $[r] = r$.

Tem-se ainda o caso em que $n = r^2$, ou seja, n é um quadrado perfeito, implicando que r é um fator de n. Na equação 2.4, teremos $x = r$ e $y = 0$. Logo, se $y > 0$, então $x = \sqrt{n + y^2} > \sqrt{n}$.

A partir dessa informação, Coutinho [1] sugere o seguinte passo a passo para desenvolver o **algoritmo de Fatoração por Fermat** para determinação de x e y:

Algoritmo de Fatoração por Fermat

a) **Entrada**: Um número inteiro positivo ímpar n.

b) **Inicialização**:

- Calcule $x_0 = [\sqrt{n}\,]$ (parte inteira de $\sqrt{n}$).
- Se $n = x_0^2$, então x_0 é um **fator** de n e o **algoritmo termina**.

c) **Laço principal** (para $k = 1, 2, \ldots$):

- Atualize $x_k = x_0 + k$.
- Calcule $y_k = \sqrt{x_k^2 - n}$.
- **Teste 1**: Se $x_k = \frac{n+1}{2}$, então n **é primo** e o **algoritmo termina.**
- **Teste 2**: Se y_k for um número inteiro, então n **é composto** e seus fatores são $x_k + y_k$ e $x_k - y_k$. O **algoritmo termina.**

d) **Repetição**: Se nenhum dos testes for satisfeito, incremente k e retorne ao passo (c).

Para compreender melhor o método, aplicaremos o algoritmo descrito anteriormente à fatoração de um número. Seguindo os passos estabelecidos, realizaremos os cálculos

necessários para determinar os fatores primos ou confirmar que o número é primo, ilustrando de forma prática o funcionamento do processo de fatoração por Fermat.

Exemplo 2.6. *Utilizando o algoritmo de formatação por Fermat, verifique se o número 17363 é primo; caso contrário, determine os valores de seus fatores.*

Solução.
Consideremos $n = 17363$. Um dos critérios de parada ocorre quando $x = \frac{17363+1}{2} = 8682$, e o outro quando encontramos um valor inteiro para y.

Inicialmente, calculamos $x = [\sqrt{17363}] = 131$. Como $x^2 - n$ não é um quadrado perfeito, incrementamos x em uma unidade e verificamos novamente. O primeiro valor de y é $\sqrt{132^2 - 17363} = 7.81$. Como o critério de parada ainda não foi atendido, seguimos incrementando x e verificando as condições do algoritmo. Os cálculos estão apresentados na tabela abaixo:

$\boldsymbol{x}$	$\boldsymbol{y = \sqrt{x^2 - n}}$
132	7.81
133	18.05
134	24.35
135	29.36
136	33.66
137	37.49
138	41.00

Tabela 2.3: Cálculos intermediários do Algoritmo por Fermat.

Assim, na sétima repetição do algoritmo, encontramos os valores de x e y que correspondem aos fatores de n: $x + y = 179$ e $x - y = 97$. □

À primeira vista, o Algoritmo de Fermat parece um método eficiente para determinar os fatores primos de parâmetro n grande de uma chave pública. Por exemplo, o número 1022117 é o produto de dois primos de quatro algarismos que, ao aplicar esse método, são facilmente determinados já na segunda iteração do algoritmo. Isso significa que seria fácil encontrar a chave privada do sistema RSA? Na realidade, não.

A verdade é que o método funciona bem quando os fatores primos são próximos entre si, pois o número de operações necessárias para a fatoração é reduzido, uma vez que esses números estão próximos de $\sqrt{1022117} \approx 1010,99$. No caso acima, os números primos são 1009 e 1013. No entanto, quando os primos que compõem n estão muito distantes um do outro, a quantidade de operações cresce significativamente, tornando o método de

fatoração inviável. Se fosse possível fatorar rapidamente o semiprimo n, a segurança do sistema RSA já teria sido comprometida, tornando esse método criptográfico obsoleto.

Essa análise revela um aspecto sutil da segurança do RSA, destacado por Coutinho [1]. À primeira vista, pode parecer suficiente escolher primos grandes para dificultar a fatoração de n. No entanto, isso não é verdade. Se os primos escolhidos forem grandes, mas próximos entre si, n poderá ser facilmente fatorado pelo Algoritmo de Fermat.

Existem outros procedimentos que combinam métodos matemáticos com técnicas computacionais, permitindo verificar se um número é composto sem a necessidade de fatorá-lo. Um desses métodos é baseado no Pequeno Teorema de Fermat 2.19. Em vez de realizar a fatoração direta, calculam-se apenas certas potências modulares, o que torna o processo bastante eficiente para identificar números compostos. Entretanto, em algumas situações, o teste pode apresentar resultados inconclusivos, não sendo suficiente para garantir se o número é primo ou composto.

Para ilustrar o funcionamento desse método, apresentaremos a seguir um exemplo descrito por Coutinho [3], denominado **Teste de composição**. Esse procedimento é bastante simples e oferece uma boa noção de como operam testes desse tipo.

Conforme mencionado acima, este teste baseia-se no Teorema de Fermat 2.19, que afirma que, sendo p um número primo e a um inteiro não divisível por p, então $a^{p-1} \equiv 1 \pmod{p}$. Considere, então, o seguinte número n com 101 algarismos:

$$\begin{aligned} n = \quad & 11111111111111111111111111111111111111111111111111 \\ & 11111111111111111111111111111111111111111111111111. \end{aligned}$$

Ao atribuir $a = 2$ e calcular $2^{n-1} \equiv r \pmod{n}$ com o auxílio de um computador, obtém-se o seguinte resto:

$$\begin{aligned} r = \quad & 52929141876542730585715988851992025959407287581866 \\ & 39710565760508670021985609520505802825349865669415. \end{aligned}$$

Sendo assim, o Teorema de Fermat garante que n é composto. Coutinho [3] afirma ainda que, com o auxílio de seu computador, conseguiu realizar os cálculos das potências modulares, obtendo o valor de r. No entanto, o algoritmo de fatoração utilizado não conseguiu determinar nenhum fator para este número n, apesar de o Teorema de Fermat assegurar que ele é composto.

De forma geral, se desejamos saber se n é composto, escolhemos um inteiro b qualquer, que não seja divisível por n, e calculamos $b^{n-1} \equiv r \pmod{n}$. Se $r \neq 1$, então o Teorema

de Fermat nos garante que n não pode ser primo. Dessa forma, temos uma maneira indireta de provar que um dado número é composto. Em outras palavras, mesmo **sem determinar nenhum fator de** n, podemos afirmar com certeza que n é composto.

O caso em que $b^{n-1} \equiv 1 \pmod{n}$, mesmo quando $b \not\equiv \pm 1 \pmod{n}$, é inconsistente, pois pode ocorrer que um número ímpar composto satisfaça esse requisito. Um exemplo simples é $n = 25$ e $b = 7$. Neste caso, calculamos $7^{24} \equiv r \pmod{25}$. Note que $7^3 \equiv 18 \pmod{25}$. A partir disso, segue que:

$$7^{24} \equiv (((7^3)^2)^4) \equiv (18^2)^4 \equiv (-1)^4 \equiv 1 \pmod{25}.$$

Conforme comenta Coutinho [3] oque acontece é que 25 comporta-se como se fosse um número primo relativamente à congruência do Teorema de Fermat, quando tomamos a base da potência como sendo 7. Tais números são conhecidos como pseudoprimos; isto é, falsos primos (pseudo é um prefixo grego que significa falso)

Assim, podemos formular o Teste de Composição utilizando de forma indireta o Teorema de Fermat com o seguinte algoritmo:

Teste de Composição.

a) **Entrada**: Um número $n > 1$, inteiro ímpar, e b, um número inteiro não divisível por n.

b) **Inicialização**:

- Calcule $b^{n-1} \equiv r \pmod{n}$;

c) **Saída**:

- Se $r \neq 1$, então n é composto.
- Se $r = 1$, então o teste é inconclusivo.

Evidentemente, " inconclusivo " significa que não podemos ter certeza se n é primo ou composto.

Essas considerações sobre os números primos e compostos nos conduzem diretamente ao próximo tópico: a Aritmética Modular, um ramo fundamental da matemática que trata de operações envolvendo restos. Esses dois conceitos estão profundamente interligados e desempenham um papel central na Teoria dos Números, sendo especialmente relevantes para aplicações práticas, como a criptografia RSA.

2.5 Aritmética Modular

A ideia de operar com restos, conforme mencionado por Domingues [6], foi formalmente introduzida por Carl Friedrich Gauss em sua obra seminal Disquisitiones Arithmeticae, publicada em 1801. Esse trabalho marcou um divisor de águas na teoria dos números, ao consolidar os fundamentos da aritmética modular e estabelecer importantes conexões entre números primos e sistemas de congruências. Essas contribuições não apenas impulsionaram o desenvolvimento da matemática pura, mas também abriram caminho para aplicações práticas, como a criptografia moderna.

Nosso objetivo aqui é compreender os princípios básicos dessa "nova"aritmética, com foco nas operações de soma, multiplicação e divisão dentro desse contexto. Para isso, começaremos formalizando sua definição, explorando suas propriedades e destacando exemplos práticos que ilustram seu funcionamento.

Definição 2.15. *Seja $m \in \mathbb{N}$. Dizemos que a e $b \in \mathbb{Z}$ são congruentes módulo m se os restos de suas divisões euclidianas por m são iguais. A relação de congruência é representada pela seguinte notação:*

$$a \equiv b \pmod{m}.$$

Por outro lado, se os restos de suas divisões euclidianas por m forem diferentes, dizemos que a e b não são congruentes, ou são incongruentes, módulo m. Nesse caso, escrevemos:

$$a \not\equiv b \pmod{m}.$$

Segue diretamente da Definição 2.15 que, para determinar se dois números têm o mesmo resto quando divididos por um número fixo $m \in \mathbb{N}$, não é preciso realizar a divisão euclidiana de ambos e comparar os restos diretamente. Em vez disso, pode-se utilizar o seguinte critério:

Proposição 2.16. *Sejam a, b, $m \in \mathbb{Z}$, com $m > 0$. Dizemos que $a \equiv b \pmod{m}$ se, e somente se, $m \mid (a - b)$, ou seja, se existe um inteiro q tal que $a - b = m \cdot q$.*

Por exemplo, $21 \equiv 13 \pmod{2}$, pois $2 \mid (21 - 13) = 8$, enquanto $19 \not\equiv 17 \pmod{3}$, pois $3 \nmid (19 - 17) = 2$.

Outra consequência da Definição 2.15 é que a congruência módulo m estabelece uma relação de equivalência em $\mathbb{Z}$.

Proposição 2.17. *Dado $m \in \mathbb{N}$ fixo e $a, b, c \in \mathbb{Z}$, valem as seguintes propriedades:*

i. ***Reflexividade:*** $a \equiv a \pmod{m}$;

ii. ***Simetria:*** *Se* $a \equiv b \pmod{m}$, *então* $b \equiv a \pmod{m}$;

iii. ***Transitividade:*** *Se* $a \equiv b \pmod{m}$ *e* $b \equiv c \pmod{m}$, *então* $a \equiv c \pmod{m}$.

Pela definição de classe de equivalência, a classe de equivalência de $a \in \mathbb{Z}$, denotada por $\bar{a}$, é dada por:

$$\bar{a} = \{b \in \mathbb{Z} : b \equiv a \pmod{m}\}.$$

O conjunto de todas as classes de equivalência da relação de congruência módulo m em $\mathbb{Z}$ é denotado por $\mathbb{Z}_m$, sendo chamado de conjunto dos inteiros módulo m. Como $a \in \mathbb{Z}$, podemos dividi-lo por m. Isso significa que existem inteiros q e r tais que:

$$a = mq + r, \qquad \text{com} \quad 0 \leq r < m.$$

Portanto, $a - r = mq$. Assim, da Proposição 2.16, temos que $a \equiv r \pmod{m}$. Isso mostra que qualquer inteiro a é congruente módulo m a um inteiro $r \in [0, m-1]$. Assim, o conjunto $\mathbb{Z}_m$ é formado unicamente pelas classes de equivalência $\bar{0}, \bar{1}, \bar{2}, \cdots, \overline{m-1}$, as quais são disjuntas entre si, ou seja,

$$\boxed{\mathbb{Z}_m = \{\bar{0}, \bar{1}, \bar{2}, \cdots, \overline{m-1}\}.}$$

A congruência é uma ferramenta extremamente útil e poderosa, principalmente por sua natureza como relação de equivalência, que é compatível com as operações de adição e multiplicação no conjunto dos números inteiros. Assim, nosso próximo passo será conceituar a soma e o produto de classes de congruência módulo m.

Proposição 2.18. *Sejam* $a, b, c, d, m \in \mathbb{Z}$, *com* $m > 1$.

i. *Se* $a \equiv b \pmod{m}$ *e* $c \equiv d \pmod{m}$, *então* $a + c \equiv b + d \pmod{m}$.

ii. *Se* $a \equiv b \pmod{m}$ *e* $c \equiv d \pmod{m}$, *então* $ac \equiv bd \pmod{m}$.

iii. *Se* $a \equiv b \pmod{m}$, *então tem-se* $a^n \equiv b^n \pmod{m}$, $\forall n \in \mathbb{N}$.

iv. *Se* $(c, m) = 1$. *Então, temos:*

$$ac \equiv bc \pmod{m} \quad \Leftrightarrow \quad a \equiv b \pmod{m}.$$

Por exemplo, como $4 \equiv 2 \pmod{2}$ e $7 \equiv 3 \pmod{2}$, segue da Proposição 2.18, itens *ii* e *iii*, respectivamente, que $28 \equiv 6 \pmod{2}$ e $4^n \equiv 2^n \pmod{2}$.

O item *iv* da Proposição 2.18 é especialmente importante nas operações com congruências e será fundamental para comprovar uma das ferramentas mais úteis da teoria dos números: o Pequeno Teorema de Fermat.

Teorema 2.19. *Sejam p um número primo e $a \in \mathbb{Z}$. Se $p \nmid a$, então*

$$a^{p-1} \equiv 1 \pmod{p}.$$

Demonstração.
Suponha inicialmente, que $a \in \mathbb{N}$ e considere os $p-1$ primeiros múltiplos positivos de a, isto é, os inteiros:

$$a, 2a, 3a, \cdots (p-1)a.$$

Observe que nenhum desses números é congruente a 0 módulo p. Além disso, dois quaisquer deles são incongruentes módulo p, pois, caso contrário, teríamos:

$$ra \equiv sa \pmod{p}, \qquad \text{com} \quad 1 \leq r < s \leq p-1.$$

Então, pela Proposição 2.18, item *iv*, o fator a poderia ser cancelado, pois $(a, p) = 1$, resultando em:

$$r \equiv s \pmod{p},$$

o que é impossível, já que $0 < s - r < p$.

Assim sendo, cada um dos inteiros $a, 2a, 3a, \cdots (p-1)a$ é congruente módulo p a um único dos inteiros $1, 2, 3, \cdots, p-1$, considerados em alguma ordem e por conseguinte, da Proposição 2.18, item *ii*,

$$a\ 2a\ 3a \cdots (p-1)a \equiv 1\ 2\ 3 \cdots (p-1) \pmod{p}.$$

Ou seja:

$$a^{p-1}(p-1)! \equiv (p-1)! \pmod{p}.$$

Como p é primo e p não divide $(p-1)!$, usando novamente a Proposição 2.18, item *iv*, podemos cancelar o fator comum $(p-1)!$. Assim, obtemos a congruência de Fermat:

$$a^{p-1} \equiv 1 \pmod{p}, \ \forall\ a \in \mathbb{N}.$$

Agora, se a for um inteiro negativo, então $-a \in \mathbb{N}$ e, como $p-1$ é par para todo $p > 2$, segue que

$$(-a)^{p-1} = a^{p-1} \equiv 1 \pmod{p}.$$

Os casos em que $a = 0$ ou $p = 2$ ficam como exercício para o leitor. □

Corolário 2.20. *Se p é um número primo e $a \in \mathbb{Z}$, então:*

$$a^p \equiv a \pmod{p}.$$

Para compreender o quão poderosas são algumas das ferramentas que estudamos até aqui, vamos discutir alguns exemplos de sua utilização.

Exemplo 2.7. *Verifique qual é o valor do resto da divisão de 5^{38} por 11.*

Solução.
Pelo Teorema 2.19, temos que $5^{10} \equiv 1 \pmod{11}$. Assim, da Proposição 2.18, item *iii*, obtemos $(5^{10})^3 \equiv 1^3 = 1 \pmod{11}$, o que implica:

$$5^{38} = (5^{10})^3 5^8 \equiv 5^8 \pmod{11}.$$

Agora, como $25 \equiv 3 \pmod{11}$, usando novamente a Proposição 2.18, item *iii*, temos:

$$5^8 = (5^2)^4 = 25^4 \equiv 3^4 = 81 \equiv 4 \pmod{11}.$$

Portanto, $5^{38} \equiv 4 \pmod{11}$. Assim, o resto da divisão de 5^{38} por 11 é 4. □

Acredito que o aspecto mais intrigante deste resultado seja o fato de que, mesmo sem conhecer diretamente o valor de 5^{38} (um número extremamente grande, cuja ordem de grandeza é de aproximadamente 10^{26}), conseguimos determinar o valor de seu resto empregando apenas a teoria desenvolvida até o momento.

Exemplo 2.8. *Determine o resto da divisão de 10^{33} por 99.*
Solução.
Como $10^2 \equiv 1 \pmod{99}$, segue da Proposição 2.18, item *iii*, que

$$(10^2)^{16} \equiv 1^{16} = 1 \pmod{99}.$$

Assim,

$$10^{33} = (10^2)^{16} 10 \equiv 10 \pmod{99}.$$

Concluímos que o resto da divisão de 10^{33} por 99 é 10. □

Os inversos modulares desempenham um papel essencial no funcionamento do método de criptografia RSA. O conceito é semelhante ao da divisão, como explica Coutinho [1].

Dado dois números reais a e b, com $b \neq 0$, dividir a por b equivale a multiplicar a por $1/b$, que é chamado de inverso de b. Esse inverso é caracterizado pela equação

$$b \cdot \frac{1}{b} = 1.$$

É importante destacar um ponto sutil que muitas vezes passa despercebido: para que um número real possua um inverso, ele deve ser, necessariamente, diferente de zero. Vamos agora verificar como essa ideia é desenvolvida no contexto de $\mathbb{Z}_m$.

Em $\mathbb{Z}_m$, o inverso de uma classe $\bar{a}$ é outra classe $\bar{y}$ tal que $\bar{a}\bar{y} = \bar{1}$. Assim, o problema de encontrar o inverso de a é equivalente a determinar um inteiro y tal que $ay \equiv 1 \pmod{m}$. Evidentemente, se $\bar{a} = \bar{0}$, então a não possui inverso.

No entanto, em $\mathbb{Z}_m$, podem existir outros elementos sem inverso além de $\bar{0}$. Para compreender melhor essa ideia, veja a Tabela 2.4, que apresenta os possíveis produtos entre as classes de resto em $\mathbb{Z}_8$ e $\mathbb{Z}_5$.

$\bullet$	$\bar{1}$	$\bar{2}$	$\bar{3}$	$\bar{4}$	$\bar{5}$	$\bar{6}$	$\bar{7}$
$\bar{1}$	$\bar{1}$	$\bar{2}$	$\bar{3}$	$\bar{4}$	$\bar{5}$	$\bar{6}$	$\bar{7}$
$\bar{2}$	$\bar{2}$	$\bar{4}$	$\bar{6}$	$\bar{0}$	$\bar{2}$	$\bar{4}$	$\bar{6}$
$\bar{3}$	$\bar{3}$	$\bar{6}$	$\bar{1}$	$\bar{4}$	$\bar{7}$	$\bar{2}$	$\bar{5}$
$\bar{4}$	$\bar{4}$	$\bar{0}$	$\bar{4}$	$\bar{0}$	$\bar{4}$	$\bar{0}$	$\bar{4}$
$\bar{5}$	$\bar{5}$	$\bar{2}$	$\bar{7}$	$\bar{4}$	$\bar{1}$	$\bar{6}$	$\bar{3}$
$\bar{6}$	$\bar{6}$	$\bar{4}$	$\bar{2}$	$\bar{0}$	$\bar{6}$	$\bar{4}$	$\bar{2}$
$\bar{7}$	$\bar{7}$	$\bar{6}$	$\bar{5}$	$\bar{4}$	$\bar{3}$	$\bar{2}$	$\bar{1}$

$\bullet$	$\bar{1}$	$\bar{2}$	$\bar{3}$	$\bar{4}$
$\bar{1}$	$\bar{1}$	$\bar{2}$	$\bar{3}$	$\bar{4}$
$\bar{2}$	$\bar{2}$	$\bar{4}$	$\bar{1}$	$\bar{3}$
$\bar{3}$	$\bar{3}$	$\bar{1}$	$\bar{4}$	$\bar{2}$
$\bar{4}$	$\bar{4}$	$\bar{3}$	$\bar{2}$	$\bar{1}$

Tabela 2.4: Tabela de produtos das classes residuais de $\mathbb{Z}_8$ e $\mathbb{Z}_5$

Analisando os produtos entre as classes de $\mathbb{Z}_8$, verifica-se facilmente que há classes que não são invertíveis: $\bar{2}, \bar{4}, \bar{6}$. Por outro lado, ao examinarmos as classes de $\mathbb{Z}_5$, observamos que todas as classes são invertíveis.

Com base nessa análise, podemos conjecturar algumas ideias. Primeiramente, as classes invertíveis de $\mathbb{Z}_8$ possuem uma característica em comum: todas elas são coprimas com 8. No caso de $\mathbb{Z}_5$, como 5 é primo, todas as classes são necessariamente invertíveis.

Vamos agora sistematizar essa observação por meio do teorema a seguir.

Teorema 2.21 (Teorema da Inversão)**.** *A classe $\bar{a} \in \mathbb{Z}_m$ é invertível se, e somente se, $(a, m) = 1$.*

Um fato importante a ser mencionado é que, além de todas as classes $\bar{a} \in \mathbb{Z}_m \setminus \{\bar{0}\}$, com $m \in \mathbb{Z}$, terem inverso se $(a, m) = 1$, o conjunto dessas classes invertíveis, que

denominaremos aqui como $\mathbb{U}_m = \{\bar{a} \in \mathbb{Z}_m : (a, m) = 1\}$, embora não seja fechado para a soma, é fechado para a multiplicação. Por exemplo, considerando $\bar{1}$ e $\bar{5} \in \mathbb{U}_{12}$, sua soma $\bar{1} + \bar{5} = \bar{6} \notin \mathbb{U}_{12}$. No entanto, seu produto $\bar{1} \cdot \bar{5} = \bar{5} \in \mathbb{U}_{12}$.

Lema 2.22. *O conjunto* $\mathbb{U}_m = \{\bar{a} \in \mathbb{Z}_m : (a, m) = 1\}$ *é fechado para a multiplicação. Ou seja, o produto de duas classes que possuem inverso também possui inverso.*

As próximas ferramentas que estudaremos são de grande importância para a teoria dos números. Uma delas é a função ϕ de Euler, também conhecida como função totiente de Euler, e a outra é o Teorema de Euler. Vale destacar a relevância do trabalho desse matemático. Como observa Roque [16], Euler foi um dos responsáveis pelas mudanças que moldaram a imagem da matemática que temos hoje, consolidada principalmente ao longo do século XIX e início do XX.

Nesse período, a história da análise, ou do cálculo infinitesimal, desempenhou um papel central nessas transformações, especialmente em um dos estágios de transformação, o analítico ou algébrico, que teve início por volta de 1740. Foi com Euler que o cálculo passou a ser entendido como uma teoria das funções, vistas como entidades distintas das curvas. A concepção de que a análise matemática é uma ciência geral das variáveis e de suas funções exerceu grande influência sobre a matemática do século XVIII, especialmente após a publicação de sua obra *Introductio in analysin infinitorum* (Introdução à Análise dos Infinitos), publicada em 1748.

Definição 2.23 (Função de Euler)**.** *Chama-se função de Euler a* $\phi : \mathbb{N} \to \mathbb{N}$*, definida como:*

$$\phi(n) = \text{ Numero de elementos do conjunto } \{x \in \mathbb{N} : 1 \leq x \leq n \quad e \quad (x, n) = 1\}.$$

Por exemplo, $\phi(1) = 1$, pois $(1, 1) = 1$. Além disso, $\phi(30) = 8$, já que os únicos números naturais menores que 30 e coprimos com ele são $1, 7, 11, 13, 17, 19, 23$ e 29. Note que, para valores elevados de n, calcular $\phi(n)$ pode se tornar uma tarefa complexa sem o uso das ferramentas que serão apresentadas a seguir:

Teorema 2.24. *Sejam* p *um número primo e* $n, m \in \mathbb{N}$*. Então:*

i. $\phi(p^n) = p^n - p^{n-1}$*, e, em particular,* $\phi(p) = p - 1$*;*

ii. $\phi(mn) = \phi(m)\phi(n)$*, desde que* $(m, n) = 1$*;*

iii. Se $n = p_1^{\alpha_1} \cdot p_2^{\alpha_2} \cdots p_k^{\alpha_k}$, *em que* $p_1 < p_2 < \cdots < p_k$ *são primos distintos, então:*

$$\phi(n) = n \cdot \left(1 - \frac{1}{p_1}\right) \cdot \left(1 - \frac{1}{p_2}\right) \cdots \left(1 - \frac{1}{p_k}\right).$$

iv. Se n *é livre de quadrados, então para todo* $k \in \mathbb{N}$ *tem-se que*

$$m^{k\phi(n)+1} \equiv m \pmod{n}.$$

Demonstração.
Aqui apresentaremos a demonstração do item *iii.* as demonstrações dos demais itens podem ser consultadas em Hefez [7].

Dos itens *ii* e *i*, temos:

$$\begin{aligned}
\phi(n) &= \phi\left(p_1^{\alpha_1} p_2^{\alpha_2} \cdots p_k^{\alpha_k}\right) = \phi(p_1^{\alpha_1})\phi(p_2^{\alpha_2}) \cdots \phi(p_k^{\alpha_k}) \\
&= p_1^{\alpha_1}\left(1 - \frac{1}{p_1}\right) p_2^{\alpha_2}\left(1 - \frac{1}{p_2}\right) \cdots p_k^{\alpha_k}\left(1 - \frac{1}{p_k}\right) \\
&= p_1^{\alpha_1} p_2^{\alpha_2} \cdots p_k^{\alpha_k}\left(1 - \frac{1}{p_1}\right)\left(1 - \frac{1}{p_2}\right) \cdots \left(1 - \frac{1}{p_k}\right).
\end{aligned}$$

□

Observação 2.25. *Na criptografia RSA, a chave pública* $[n, e]$ *e a chave privada* $[n, d]$ *estão matematicamente relacionadas pela congruência*

$$ed \equiv 1 \pmod{\phi(n)}, \tag{2.5}$$

onde $(e, \phi(n)) = 1$ *e* $n = pq$, *com* p *e* q *sendo primos. Como a chave* $[n, e]$ *é de conhecimento geral, qualquer pessoa pode utilizá-la para criptografar mensagens. No entanto, apenas quem possui a chave* $[n, d]$ *pode descriptografá-las.*

De fato, quem gerou as chaves conhece a fatoração de n *e pode, sem dificuldades, usando o Teorema 2.24, item (iii), calcular:*

$$\phi(n) = \phi(pq) = \phi(p)\,\phi(q) = (p-1)(q-1).$$

Assim, o inverso modular de e, que é d, pode ser facilmente determinado a partir de (2.5), utilizando, por exemplo, o algoritmo euclidiano estendido. Por outro lado, para quem não conhece a fatoração de n, será necessário fatorá-lo antes de calcular $\phi(n)$, o que, para n suficientemente grande, representa um problema computacional extremamente difícil. Assim, a segurança do RSA é garantida, pois, sem a fatoração de n, a determinação de d se torna inviável.

A seguir, exploraremos outra ferramenta fundamental nas operações da aritmética modular: o **Teorema de Euler**. Para sua demonstração, utilizaremos o seguinte lema, que desempenha um papel essencial no desenvolvimento da prova.

Lema 2.26. *Sejam a e $n > 1$ inteiros tais que $(a, n) = 1$. Se $a_1, a_2, a_3, \ldots, a_{\phi(n)}$ são inteiros positivos menores que n e coprimos com n, então cada um dos produtos:*

$$aa_1,\ aa_2,\ aa_3,\ \cdots,\ aa_{\phi(n)}$$

é congruente módulo n a um dos inteiros $a_1, a_2, a_3, \ldots, a_{\phi(n)}$, não necessariamente na mesma ordem.

Teorema 2.27 (Teorema de Euler)**.** *Sejam $n \in \mathbb{N}$ e $a \in \mathbb{Z}$ tal que $(a, n) = 1$. Então:*

$$a^{\phi(n)} \equiv 1 \pmod{n}.$$

Demonstração.
Para $n = 1$, a proposição é verdadeira, pois

$$a^{\phi(1)} = a \equiv 1 \pmod{1}.$$

Agora, consideremos agora $n > 1$ e sejam a_1, a_2, $a_3, \cdots, a_{\phi(n)}$ inteiros positivos menores que n e coprimos com n. Como $(a, n) = 1$, pelo Lema 2.26, os produtos aa_1, aa_2, $aa_3, \cdots, aa_{\phi(n)}$ são congruentes módulo n a algum dos a_1, a_2, $a_3, \cdots, a_{\phi(n)}$.

Assim, pela Proposição 2.18, item *ii*, podemos multiplicar todas essas congruências lado a lado, obtendo:

$$aa_1 \cdot aa_2 \cdots aa_{\phi(n)} \equiv a_1 \cdot a_2 \cdots a_{\phi(n)} \pmod{n}.$$

Reorganizando os fatores no lado esquerdo, temos:

$$a^{\phi(n)} a_1 a_2 \cdots a_{\phi(n)} \equiv a_1 \cdot a_2 \cdots a_{\phi(n)} \pmod{n}.$$

Como $(a_1 \cdot a_2 \cdots a_{\phi(n)}, n) = 1$, pela Proposição 2.18, item iv, podemos cancelar o fator $a_1 \cdot a_2 \cdots a_{\phi(n)}$ em ambos lados da congruência. Assim, obtemos:

$$a^{\phi(n)} \equiv 1 \pmod{n}.$$

□

Observe que, para um número primo p, temos $\phi(p) = p - 1$. Assim, pelo Teorema de Euler, segue que

$$a^{\phi(p)} = a^{p-1} \equiv 1 \pmod{p},$$

o que corresponde ao Pequeno Teorema de Fermat.

Com a discussão deste teorema, concluímos a apresentação dos conceitos básicos, que possibilitarão uma melhor compreensão e análise do sistema de criptografia RSA, a ser estudado no próximo capítulo.

Capítulo

3

O Método RSA: Uma Introdução Simplificada

Neste capítulo, apresenta-se a essência do funcionamento do método de criptografia RSA. Em particular, demonstra-se que, ao descriptografar um texto criptografado utilizando esse método, é possível recuperar a mensagem original. Além disso, discute-se por que o método RSA é amplamente utilizado e considerado seguro para a transmissão de dados sensíveis, como informações bancárias e transações comerciais. A abordagem apresentada busca, também, oferecer aos professores do ensino básico uma perspectiva que enriqueça o ensino de conceitos matemáticos, contextualizando-os de forma prática e envolvente.

Este capítulo fundamenta-se nas obras de Coutinho [1] e Hefez [7]. Destaca-se, em particular, a contribuição de Coutinho, que aborda o tema de forma intuitiva, sem perder a formalidade matemática necessária para a compreensão do método.

3.1 Processo de Precodificação

Para a utilização do método RSA, é necessário, primeiramente, converter a mensagem em uma sequência de números. Para simplificar o entendimento do método, a mensagem que será cifrada conterá apenas letras, sem números ou outros símbolos. Dessa forma, a mensagem será composta apenas por letras e espaços entre as palavras.

Seguindo a denominação de Coutinho [1], chamaremos esta etapa de **precodificação**, para distingui-la do processo de codificação propriamente dito. Nessa etapa, os caracteres básicos e o espaço entre as palavras, representado por (-), serão convertidos em números

de acordo com a seguinte tabela de conversão:

A	**B**	**C**	**D**	**E**	**F**	**G**	**H**	**I**	**J**	**K**	**L**	**M**
10	11	12	13	14	15	16	17	18	19	20	21	22

N	**O**	**P**	**Q**	**R**	**S**	**T**	**U**	**V**	**W**	**X**	**Y**	**Z**	–
23	24	25	26	27	28	29	30	31	32	33	34	35	36

Tabela 3.1: Tabela de Conversão de Caracteres em Valores Numéricos

Por exemplo, utilizando a Tabela 3.1, a frase **PROFMAT NA UFMT** corresponde à seguinte sequência de números:

25 27 24 15 22 10 29 36 23 10 36 30 15 22 29

Quando representada como um único número contínuo, obtemos:

$$252724152210293610363015222 9 \tag{3.1}$$

É importante destacar que, segundo Coutinho [1], na construção da Tabela 3.1, cada caractere foi associado a um número de dois algarismos, o que evita ambiguidades. Por exemplo, suponha que a letra **A** fosse representada pelo número 1. Seguindo essa lógica, **B** corresponderia ao número 2 e assim sucessivamente. Nesse caso, a sequência numérica 21 poderia ser interpretada tanto como **BA** quanto como **U**, que ocupa a posição 21 no alfabeto, tornando a interpretação ambígua. O uso de números de dois algarismos elimina esse problema, garantindo uma representação clara e sem confusão.

Outro aspecto importante é a **quebra do número contínuo em blocos**, ou seja, o processo de segmentação de uma sequência numérica longa em partes menores, chamadas **blocos**. Por exemplo, na criptografia RSA, ao converter uma mensagem em números, o resultado pode ser um valor muito grande, conforme ilustrado em (3.1). Para viabilizar a criptografia, essa sequência numérica é dividida em blocos de tamanho adequado, menores que o parâmetro $n = pq$, e organizados de forma que o primeiro algarismo de cada bloco seja diferente de zero. Essa divisão garante que cada bloco possa ser processado individualmente sem exceder os limites do sistema criptográfico, além de evitar a perda de dados na encriptação devido a possíveis ambiguidades.

Assim, se tomarmos, por exemplo, $p = 11$ e $q = 13$, temos que $n = 143$. Dessa forma, podemos dividir a expressão (3.1) da seguinte maneira:

Bloco 01	Bloco 02	Bloco 03	Bloco 04	Bloco 05	Bloco 06	Bloco 07
25	27	24	15	22	102	93

Bloco 08	Bloco 09	Bloco 10	Bloco 11	Bloco 12	Bloco 13	Bloco 14
6	103	6	30	15	22	29

Tabela 3.2: Blocos numéricos formados no processo de precodificação.

3.2 Implementação do Processo de Codificação

Com a precodificação da frase **PROFMAT NA UFMT**, conforme (3.1), podemos prosseguir para a etapa de codificação. A partir da Observação 2.25, além do número $n = 143$, definido anteriormente como o produto dos números primos $p = 11$ e $q = 13$, é necessário definir outro número, denotado por e. Esse número deve satisfazer a condição $(\phi(n), e) = 1$, garantindo que e seja inversível módulo $\phi(n)$.

Cada bloco da Tabela 3.2 será codificado separadamente, e a mensagem codificada será a sequência dos blocos codificados. Vale destacar que a posição de cada bloco é essencial para preservar a integridade da mensagem. Assim, a união desses blocos para formar um único número contínuo codificado não deve ser realizada, pois isso inviabilizará a decodificação correta.

Dada a chave pública $[n, e]$, também conhecida como chave de criptografia do sistema RSA, procederemos à codificação.

Processo de Codificação.

A codificação de um bloco b da Tabela 3.2 será representada por $C(b)$ e é realizada da seguinte forma:

- Escolha números primos distintos p e q, e calcule $n = pq$, $\phi(n) = (p-1)(q-1)$. Em seguida, escolha um número inteiro $1 < e < \phi(n)$, com $(\phi(n), e) = 1$;
- Calcule $C(b)$ utilizando a seguinte expressão:

$$C(b) = \text{resto da divisão de } b^e \text{ por } n, \text{ ou seja, } C(b) \equiv b^e \pmod{n}. \quad (3.2)$$

Assim, para $p = 11$, $q = 13$ temos que $n = 143$ e

$$\phi(143) = (11 - 1)\,(13 - 1) = 12(10) = 120.$$

Em seguida, escolha $e = 7$ que é o menor inteiro entre 1 e 120 tal que $(7, 120) = 1$. Agora, usando (3.2), codificaremos os blocos da Tabela 3.2.

Bloco 01: Para $b = 25$,

$$C(25) \equiv 25^7 = (25^3)^2 \cdot 25 \equiv (38)^2 \cdot 25 \equiv 14 \cdot 25 \equiv 350 \equiv 64 \pmod{143}.$$

Portanto, $C(25) = 64$.

Bloco 02: Para $b = 27$,

$$C(27) \equiv 27^7 \equiv (27^3)^2 \cdot 27 \equiv (92)^2 \cdot 27 \equiv 27 \cdot 27 \equiv 14 \pmod{143}.$$

Portanto, $C(27) = 14$.

Bloco 03: Para $b = 24$,

$$C(24) \equiv 24^7 \equiv (24^3)^2 \cdot 24 \equiv (96)^2 \cdot 24 \equiv 64 \cdot 24 \equiv 106 \pmod{143}.$$

Portanto, $C(24) = 106$.

Bloco 04: Para $b = 15$,

$$C(15) \equiv 15^7 \equiv (15^3)^2 \cdot 15 \equiv (86)^2 \cdot 15 \equiv 103 \cdot 15 \equiv 115 \pmod{143}.$$

Portanto, $C(15) = 115$.

Bloco 05: Para $b = 22$,

$$C(22) \equiv 22^7 \equiv (22^3)^2 \cdot 22 \equiv (66)^2 \cdot 22 \equiv 66 \cdot 22 \equiv 22 \pmod{143}.$$

Portanto, $C(22) = 22$.

Bloco 06: Para $b = 102$,

$$C(102) \equiv 102^7 \equiv (102^3)^2 \cdot 102 \equiv (5)^2 \cdot 102 \equiv 25 \cdot 102 \equiv 119 \pmod{143}.$$

Portanto, $C(102) = 119$.

Bloco 07: Para $b = 93$,

$$C(93) \equiv 93^7 \equiv (93^3)^2 \cdot 93 \equiv (125)^2 \cdot 93 \equiv 38 \cdot 93 \equiv 102 \pmod{143}.$$

Portanto, $C(93) = 102$.

Seguindo o mesmo procedimento desenvolvido até então, a mensagem codificada será formada pelos blocos gerados sucessivamente até o bloco 14:

Bloco 01	**Bloco 02**	**Bloco 03**	**Bloco 04**	**Bloco 05**	**Bloco 06**	**Bloco 07**
64	14	106	115	22	119	102

Bloco 08	**Bloco 09**	**Bloco 10**	**Bloco 11**	**Bloco 12**	**Bloco 13**	**Bloco 14**
85	38	85	134	115	22	94

Tabela 3.3: Blocos numéricos resultantes da codificação da mensagem

3.3 Processo de Decodificação

Para decodificar a mensagem dada pela Tabela 3.3, é necessário utilizar dois números que correspondem à chave de decodificação, ou chave privada: $[n, d]$. Conforme a Observação 2.25, o número d é o inverso de e módulo $\phi(n)$.

Processo de Decodificação

A decodificação de um bloco a da Tabela 3.3 será representada por $D(a)$ e é realizada da seguinte forma:

- Como e e $\phi(n)$ são conhecidos a partir do processo de codificação, calcule d resolvendo a congruência:

$$ed \equiv 1 \pmod{\phi(n)}. \tag{3.3}$$

- Em seguida, determine $D(a)$ utilizando a seguinte expressão:

$$D(a) = \text{resto da divisão de } a^d \text{ por } n, \text{ ou seja, } D(a) \equiv a^d \pmod{n}. \tag{3.4}$$

Observação 3.1. *Sendo $a = C(b)$ a codificação de um bloco b da Tabela 3.2, temos que*

$$D(a) \equiv b \pmod{n},$$

ou seja, o processo de decodificação nos permite recuperar a mensagem original. De fato, pela equação (3.3), existe um inteiro $k \in \mathbb{N}$ tal que $ed = k\phi(n) + 1$. Assim, usando (3.4)

e (3.2), *juntamente com o item iv do Teorema* 2.24, *obtemos*

$$D(a) \equiv a^d \equiv (b^e)^d = b^{k\phi(n)+1} \equiv b \pmod{n}.$$

Para compreender melhor esse processo, continuaremos com a decodificação dos blocos da Tabela 3.3. Como vimos, tomando $p = 11$ e $q = 13$ determinamos $n = 143$, $\phi(143) = 120$ e $e = 7$. A seguir, determinamos o valor de d resolvendo $7d \equiv 1 \pmod{120}$. Para isso utilizamos o algoritmo estendido de Euclides. Dividindo 120 por 7, obtemos:

$$120 = 7 \cdot 17 + 1 \Rightarrow 120 - 17 \cdot 7 = 1.$$

A partir dessa equação, concluímos que o inverso modular de 7 módulo 120 é -17. Como d deve ser um número positivo, ajustamos o valor somando 120:

$$d = -17 + 120 = 103.$$

Assim, o inverso de 7 módulo 120 é $d = 103$.

Com isso, usando (3.4), iniciamos a decodificação da mensagem, bloco por bloco:

Bloco 01: Para $a = 64$, como

$$\textbf{(i)}\ 64^2 \equiv 92 \pmod{143}, \qquad \textbf{(ii)}\ 64^3 \equiv 25 \pmod{143},$$
$$\textbf{(iii)}\ 64^5 \equiv 12 \pmod{143}, \qquad \textbf{(iv)}\ 64^{10} \equiv 1 \pmod{143},$$

temos que

$$D(64) = 64^{103} \equiv (64^{10})^{10}64^3 \equiv (1)^{10} \cdot 25 \equiv 25 \pmod{143}.$$

Portanto, $D(64) = 25$.

Bloco 02: Para $a = 14$, como

$$\textbf{(i)}\ 14^2 \equiv 53 \pmod{143}, \qquad \textbf{(ii)}\ 14^3 \equiv 27 \pmod{143},$$
$$\textbf{(iii)}\ 14^5 \equiv 1 \pmod{143}, \qquad \textbf{(iv)}\ 14^{10} \equiv 1 \pmod{143},$$

temos que

$$D(14) = 14^{103} \equiv (14^{10})^{10}14^3 \equiv (1)^{10} \cdot 27 \equiv 27 \pmod{143}.$$

Portanto, $D(14) = 27$.

Bloco 03: Para $a = 106$, como

$$\textbf{(i)}\ 106^2 \equiv 82 \pmod{143}, \qquad \textbf{(ii)}\ 106^3 \equiv 112 \pmod{143},$$
$$\textbf{(iii)}\ 106^5 \equiv 32 \pmod{143}, \qquad \textbf{(iv)}\ 106^{10} \equiv 23 \pmod{143},$$

temos que

$$\begin{aligned} D(106) &= 106^{103} \equiv (106^{10})^{10}106^3 \equiv (23^2)^5 \cdot 112 \equiv 529^5 \cdot 112 \\ &\equiv 100^5 \cdot 112 \equiv 24 \pmod{143}. \end{aligned}$$

Portanto, $D(106) = 24$.

Bloco 04: Para $a = 115$, como

$$\textbf{(i)}\ 115^2 \equiv 69 \pmod{143}, \qquad \textbf{(ii)}\ 115^3 \equiv 70 \pmod{143},$$
$$\textbf{(iii)}\ 115^5 \equiv 111 \pmod{143}, \qquad \textbf{(iv)}\ 115^{10} \equiv 23 \pmod{143},$$

temos que

$$D(115) = 115^{103} \equiv (115^{10})^{10}115^3 \equiv 23^{10} \cdot 70 \equiv 133 \cdot 70 \equiv 15 \pmod{143}.$$

Portanto, $D(115) = 15$.

Bloco 05: Para $a = 22$, como

$$\textbf{(i)}\ 22^2 \equiv 484 \equiv 55 \pmod{143}, \qquad \textbf{(ii)}\ 22^3 \equiv 22 \cdot 55 \equiv 66 \pmod{143},$$
$$\textbf{(iii)}\ 22^5 \equiv 55 \cdot 66 \equiv 55 \pmod{143}, \qquad \textbf{(iv)}\ 22^{10} \equiv (55)^2 \equiv 22 \pmod{143},$$

temos que

$$\begin{aligned} D(22) &= 22^{103} \equiv (22^{10})^{10}22^3 \equiv (22^{10})^{10} \cdot 22^3 \equiv 22^{10} \cdot 66 \\ &\equiv 22 \cdot 66 \equiv 22 \pmod{143}. \end{aligned}$$

Portanto, $D(22) = 22$.

Bloco 06: Para $a = 119$, como

$$\textbf{(i)}\ 119^2 \equiv 14161 \equiv 4 \pmod{143}, \qquad \textbf{(ii)}\ 119^3 \equiv 4 \cdot 119 \equiv 47 \pmod{143},$$
$$\textbf{(iii)}\ 119^5 \equiv 4 \cdot 47 \equiv 45 \pmod{143}, \qquad \textbf{(iv)}\ 119^{10} \equiv (45)^2 \equiv 23 \pmod{143},$$

temos que

$$\begin{aligned} D(119) &= 119^{103} \equiv (119^{10})^{10}119^3 \equiv (23^2)^5 \cdot 47 \equiv 100^5 \cdot 47 \\ &\equiv 133 \cdot 47 \equiv 102 \pmod{143}. \end{aligned}$$

Portanto, $D(119) = 102$.

Bloco 07: Para $a = 102$, como

(i) $102^2 \equiv 10404 \equiv 108 \pmod{143}$, **(ii)** $102^3 \equiv 108 \cdot 102 \equiv 5 \pmod{143}$,
(iii) $102^5 \equiv 108 \cdot 111 \equiv -32 \pmod{143}$, **(iv)** $102^{10} \equiv (-32)^2 \equiv 23 \pmod{143}$,

temos que

$$D(102) = 102^{103} \equiv (102^{10})^{10}102^3 \equiv (23^2)^5 \cdot 5 \equiv 100^5 \cdot 5 \equiv 93 \pmod{143}.$$

Portanto, $D(102) = 93$.

Seguindo o mesmo procedimento, a mensagem decodificada será composta pelos seguintes blocos:

$$25 - 27 - 24 - 15 - 22 - 102 - 93 - 6 - 103 - 6 - 30 - 15 - 22 - 29.$$

Transformando-os em uma sequência contínua,

$$25272415221029361036301522 29$$

Agora separando-os em números de dois algarismos e fazendo a correspondência via a Tabela 3.1, obtemos:

$25 = \mathbf{P},\ 27 = \mathbf{R},\ 24 = \mathbf{O},\ 15 = \mathbf{F},\ 22 = \mathbf{M},\ 10 = \mathbf{A},\ 29 = \mathbf{T},\ 36 = (\text{Espaço}),$
$23 = \mathbf{N},\ 10 = \mathbf{A},\ 36 = (\text{Espaço}),\ 30 = \mathbf{U},\ 15 = \mathbf{F},\ 22 = \mathbf{M},\ 29 = \mathbf{T}.$

Assim, obtemos a frase decodificada utilizando o princípio do método RSA. É importante destacar um aspecto essencial desse processo: a validação de um dos princípios fundamentais da segurança da informação, a integridade. Ou seja, o algoritmo não alterou as informações durante os processos de codificação e decodificação. A frase decodificada é: **PROFMAT NA UFMT**.

3.4 Como o RSA Garante a Segurança

Ao longo deste trabalho, verificamos que a essência do método RSA consiste na utilização de duas chaves. Uma **chave pública**, representada pelo par de números inteiros $[n, e]$, empregada para a codificação da mensagem e acessível a todos os usuários do método. Outra **chave privada**, representada pelo par de números inteiros $[n, d]$, exclusivamente destinada à decodificação da mensagem e de uso restrito do destinatário. Além disso, vimos na Observação 2.25 como essas chaves se relacionam.

A questão da segurança está diretamente relacionada à escolha dos números primos p e q, que devem ser grandes e significativamente distintos em magnitude, conforme destacado na Seção 2.4. Dessa forma, o produto desses primos resulta em um semiprimo n de difícil fatoração. Por outro lado, a definição do parâmetro d, conforme explicado na Observação 2.25, depende diretamente dos valores de p e q. Surge então a pergunta: qual deve ser o tamanho mínimo do parâmetro n para garantir a segurança do sistema?

Coutinho [3] ressalta que implementações comerciais do RSA geralmente utilizam chaves públicas com aproximadamente 200 dígitos, embora algumas versões suportem chaves com até 2467 dígitos. Essa variação no tamanho das chaves é justamente o que assegura a robustez da segurança do método.

Visto que até o presente momento, não existe um algoritmo eficiente capaz de fatorar n em um número de operações que seja concluído em um tempo aceitável. Isso também pode ser entendido como a ausência de computadores suficientemente potentes para processar tal quantidade de dados em um tempo razoável. Dessa forma, a consistência da segurança do método RSA depende diretamente dessa limitação.

Um exemplo disso é apresentado em um artigo publicado em agosto de 2010, por um grupo de pesquisadores de diversas universidades [17]. O artigo descreve o método utilizado e o tempo necessário para fatorar um número n de 768 bits, equivalente a 232 dígitos. O número fatorado, conhecido como RSA-768, está representado a seguir:

$$123018668453011775513049495838496272077285356959533479219732245215172$$
$$640050726365751874520219978646938995647494277406384592519255732630345$$
$$373154826850791702612214291346167042921431160222124047927473779408066$$
$$535141959745985690214341 3.$$

O mesmo artigo relata que, para fatorar esse número, o grupo de pesquisadores utilizou centenas de máquinas, trabalhando em conjunto, e o processo levou dois anos para ser concluído. Segundo os pesquisadores, caso a fatoração fosse realizada em apenas um núcleo

de um processador AMD Opteron de 2,2 GHz com 2 GB de RAM, o tempo necessário para completar a tarefa seria de aproximadamente 1.500 anos.

Além disso, os pesquisadores estimam que fatorar um número n de um RSA com 1024 bits, equivalente a 309 dígitos decimais, seria cerca de mil vezes mais difícil do que fatorar um número de 768 bits.

Ainda nesse sentido, a Tabela (3.4), extraída de um artigo de fevereiro de 1978 [20] elaborada pelos próprios autores do método RSA, apresenta estimativas sobre o tempo e a quantidade de operações necessárias para fatorar um número n em função de sua quantidade de dígitos.

Quantidade de dígitos no número n	**Número de operações**	**Tempo estimado**
50	$1,4 \cdot 10^{10}$	$3,9$ **horas**
70	$9 \cdot 10^{12}$	104 **dias**
100	$2,3 \cdot 10^{15}$	74 **anos**
200	$1,2 \cdot 10^{23}$	$3,8 \cdot 10^{9}$ **anos**
300	$1,5 \cdot 10^{29}$	$4,9 \cdot 10^{15}$ **anos**
500	$1,3 \cdot 10^{39}$	$4,2 \cdot 10^{25}$ **anos**

Tabela 3.4: Estimativa do tempo de fatoração em função da quantidade de dígitos de n

Antes de encerrar este capítulo, apresentaremos um breve comentário sobre a determinação de números primos extremamente grandes. Conforme verificado no Teorema 2.14, existem infinitos números primos. Embora possa parecer que determinar números primos grandes seja uma tarefa difícil, a realidade é que ela não é tão complexa. Como observa Coutinho [3], é possível determinar se um número é composto ou primo sem necessariamente fatorá-lo. Isso é realizado por meio de métodos matemáticos combinados com técnicas computacionais, que permitem essa verificação de maneira eficiente. Um exemplo disso foi o Teste de Composição descrito no capítulo anterior (Seção 2.4).

Um exemplo recente que demonstra a facilidade em encontrar números primos grandes foi a descoberta de um número primo com quase 25 milhões de dígitos. Segundo uma notícia publicada no site do IMPA [21], matemáticos do projeto de pesquisa **Great Internet Mersenne Prime Search (GIMPS)** confirmaram a descoberta de um número primo com exatamente 24.862.048 dígitos. Esse número supera o recorde anterior, estabelecido em 2017, em mais de 1,5 milhão de dígitos.

Esse número pertence à classe especial de números primos raros, conhecidos como **primos de Mersenne**, que recebem esse nome em homenagem ao monge francês Marin Mersenne, que estudou tais primos há cerca de 350 anos. Os primos de Mersenne seguem

a fórmula simples $2^n - 1$, em que n é um número inteiro positivo. O número descoberto, já apelidado de **M82589933**, é o $51^{\underline{o}}$ primo de Mersenne conhecido. Ele foi identificado após computadores calcularem $2^{82.589.933} - 1$.

A descoberta foi realizada com o auxílio do projeto GIMPS, que permite aos usuários 'baixar' um software especial que é executado em segundo plano para buscar números primos. Desde sua criação, o projeto já descobriu os últimos 17 primos de Mersenne. A revelação do **M82589933** foi feita por Patrick Laroche, de Ocala (Flórida, EUA), um profissional de TI que utilizava o software disponibilizado pelo projeto como um "teste de estresse" gratuito para validar suas compilações de computador. Encontrado em 7 de dezembro, o M82589933 passou duas semanas sendo verificado por matemáticos do projeto e foi oficialmente anunciado em 21 de dezembro 2019.

Neste capítulo, exploramos como o método RSA garante sua segurança por meio da dificuldade de fatorar números extremamente grandes e da escolha criteriosa dos primos p e q. Exemplos como a fatoração do RSA-768 e a descoberta de grandes números primos, como o $M82589933$, demonstram a resistência do método frente aos avanços computacionais. No próximo capítulo, exploraremos alguns dos conceitos matemáticos do método RSA na Educação Básica, aliados à programação de algoritmos e ao uso de planilhas eletrônicas.

Capítulo

4

Explorando os Conceitos Matemáticos do Método RSA na Educação Básica

O principal objetivo deste capítulo é apresentar propostas para que os professores de matemática tornem a disciplina mais atrativa e significativa para os alunos. Essas propostas visam proporcionar a aquisição de conceitos matemáticos desenvolvidos ao longo deste trabalho, relacionando-os aos conteúdos presentes na educação básica, de forma a conectar o aprendizado teórico com a prática.

Além disso, busca-se ampliar o interesse e o engajamento dos alunos, que frequentemente veem a matemática como algo monótono, desinteressante e desconectado de sua realidade. A ideia é oferecer alternativas que tornem o ensino mais dinâmico e próximo do cotidiano dos estudantes, incentivando uma abordagem mais interativa e prática.

Considera-se ainda, na elaboração das propostas, algumas particularidades relacionadas ao ensino público do estado de **Mato Grosso (MT)**, as quais podem ser facilmente adaptadas a outras realidades. Uma delas é que grande parte das escolas públicas do estado dispõe de Chromebooks em quantidade suficiente para que cada aluno tenha acesso a um dispositivo. Esses aparelhos possuem, de forma nativa, o Google Sheets, permitindo que os alunos utilizem a ferramenta tanto na escola quanto em casa. Considerando que em alguns casos, os alunos podem levar os aparelhos para casa em regime de comodato, o que possibilita seu uso com acesso à internet. Essa infraestrutura tecnológica favorece a implementação de atividades pedagógicas mais dinâmicas e integradas ao uso de ferramentas digitais.

Nesse sentido, as propostas apresentadas estão alinhadas às diretrizes da **Base**

Nacional Comum Curricular (BNCC) [22] e ao seu complemento referente à **Computação** [23], sendo indicadas as **habilidades** previstas ao final de cada exercício e exemplo apresentado. Nesse contexto, destaca-se a importância de difundir o uso de recursos tecnológicos, os quais podem ser ferramentas eficazes para aprimorar o processo de ensino e aprendizagem. Sobre esse aspecto, a BNCC [22] afirma:

> "...a ampliação e o aprofundamento das aprendizagens essenciais desenvolvidas até o 9º ano do Ensino Fundamental. Para tanto, coloca em jogo, de modo mais inter-relacionado, os conhecimentos já explorados na etapa anterior, possibilitando que os estudantes construam uma visão mais integrada da matemática, ainda na perspectiva de sua aplicação à realidade."

Além de consolidar os conceitos fundamentais, a **BNCC** enfatiza a importância da tecnologia como um dos elementos-chave para o desenvolvimento das competências matemáticas. Recomenda que tecnologias, como planilhas eletrônicas e calculadoras, sejam introduzidas desde os **primeiros anos do Ensino Fundamental**. Essa abordagem tem como objetivo não apenas preparar os estudantes para as demandas do mundo atual, mas também promover o desenvolvimento do pensamento computacional. Sobre isso, a **BNCC** [22] afirma:

> Tal valorização possibilita que, ao chegarem aos anos finais, eles possam ser estimulados a desenvolver o pensamento computacional, por meio da interpretação e da elaboração de fluxogramas e algoritmos.

A BNCC que integra a Computação [23] a Educação Básica definindo os conteúdos e habilidades relacionados à Educação Digital que devem ser abordados no Ensino Básico. Tem como um dos eixos estruturantes o desenvolvimento do pensamento computacional, especialmente do 6º ao 9º ano, com foco em dois objetos de conhecimento: Programação e Estratégias de solução de problemas. Assim apresentaremos aqui uma síntese dos objetos de conhecimento e das habilidades associadas, conforme a estrutura sugerida pelo documento.

OBJETO DE CONHECIMENTO		HABILIDADE	
PROGRAMAÇÃO	Tipos de dados	Construir e analisar soluções computacionais de problemas de diferentes áreas do conhecimento, de forma individual ou colaborativa, selecionando as estruturas de dados adequadas (registros, matrizes, listas e grafos), aperfeiçoando e articulando saberes escolares.	(EF69CO01) Classificar informações, agrupando-as em coleções (conjuntos) e associando cada coleção a um 'tipo de dado
	Linguagem de Programação		(EF69CO02) Elaborar algoritmos que envolvam instruções sequenciais, de repetição e de seleção usando uma linguagem de programação.
			(EF69CO03) Descrever com precisão a solução de um problema, construindo o programa que implementa a solução descrita.

Tabela 4.1: Habilidades associadas ao objeto de conhecimento Programação. retirada da BNCC [23].

O objeto de conhecimento **Programação** promove o aprendizado de conceitos fundamentais da computação, como algoritmos, estruturas de dados e linguagens de programação. Além disso, o uso da programação no currículo contribui para que os estudantes desenvolvam competências práticas que os preparem para demandas futuras em diversas áreas do conhecimento.

OBJETO DE CONHECIMENTO		HABILIDADE	
ESTRATÉGIAS DE SOLUÇÃO DE PROBLEMAS.	Decomposição	Empregar diferentes estratégias da Computação (decomposição, generalização e reúso) para construir a solução de problemas	(EF69CO04) Construir soluções de problemas usando a técnica de decomposição e automatizar tais soluções usando uma linguagem de programação.
	Generalização		(EF69CO05) Identificar os recursos ou insumos necessários (entradas) para a resolução de problemas, bem como os resultados esperados (saídas), determinando os respectivos tipos de dados, e estabelecendo a definição de problema como uma relação entre entrada e saída.
			(EF69CO06) Comparar diferentes casos particulares (instâncias) de um mesmo problema, identificando as semelhanças e diferenças entre eles, e criar um algoritmo para resolver todos, fazendo uso de variáveis (parâmetros) para permitir o tratamento de todos os casos de forma genérica.

Tabela 4.2: Habilidades associadas a Resolução de Problemas retirada da BNCC [23].

Observa-se, no objeto de conhecimento **Estratégias de Solução de Problemas**, a ideia de capacitar os alunos a enfrentar desafios complexos por meio do raciocínio lógico, da análise crítica e da criatividade. Essa abordagem inclui a decomposição de problemas em partes menores, a identificação de padrões e a aplicação de técnicas para criar soluções eficientes e inovadoras. Ao trabalhar essas habilidades, os estudantes desenvolvem não apenas competências técnicas, mas também competências socioemocionais, como o

trabalho em equipe, a persistência e a tomada de decisão baseada em evidências.

Apesar de o desenvolvimento do pensamento computacional apresentado aqui estar relacionado à etapa do Ensino Fundamental II, acredito que seja possível explorar essas habilidades também no Ensino Médio, visto que muitos estudantes não desenvolveram plenamente essas habilidades. Nesse sentido, a integração desses objetos de conhecimento tanto ao Ensino Fundamental II quanto ao Ensino Médio, conforme as diretrizes da BNCC, desempenha um papel importante na formação dos estudantes. Essas habilidades não apenas fortalecem a capacidade de resolver problemas e pensar de forma lógica, mas também preparam os alunos para um futuro em que a tecnologia desempenha um papel cada vez mais central nas mais diversas áreas da vida e do trabalho.

No mesmo sentido das orientações da **BNCC**, a Lei de Diretrizes e Bases da Educação Nacional (**LDB**)[24] faz referência ao ensino médio, estabelecendo que ele possui as seguintes finalidades:

- Preparação básica para o trabalho e a cidadania do educando, para continuar aprendendo, de modo a ser capaz de se adaptar com flexibilidade a novas condições de ocupação ou aperfeiçoamento posteriores.

- Compreensão dos fundamentos científico-tecnológicos dos processos produtivos, relacionando teoria e prática no ensino de cada disciplina.

É importante destacar que, para introduzir as propostas de programação, utilizaremos uma metodologia ativa baseada na sala de aula invertida. Segundo Castellar [25], essa metodologia tem como principais aspectos a reorganização de dois elementos essenciais da chamada cultura escolar: a ordem de realização das atividades e a organização do tempo e espaço dessa sequência. Castellar ainda propõe o seguinte diagrama para ilustrar essa reorganização:

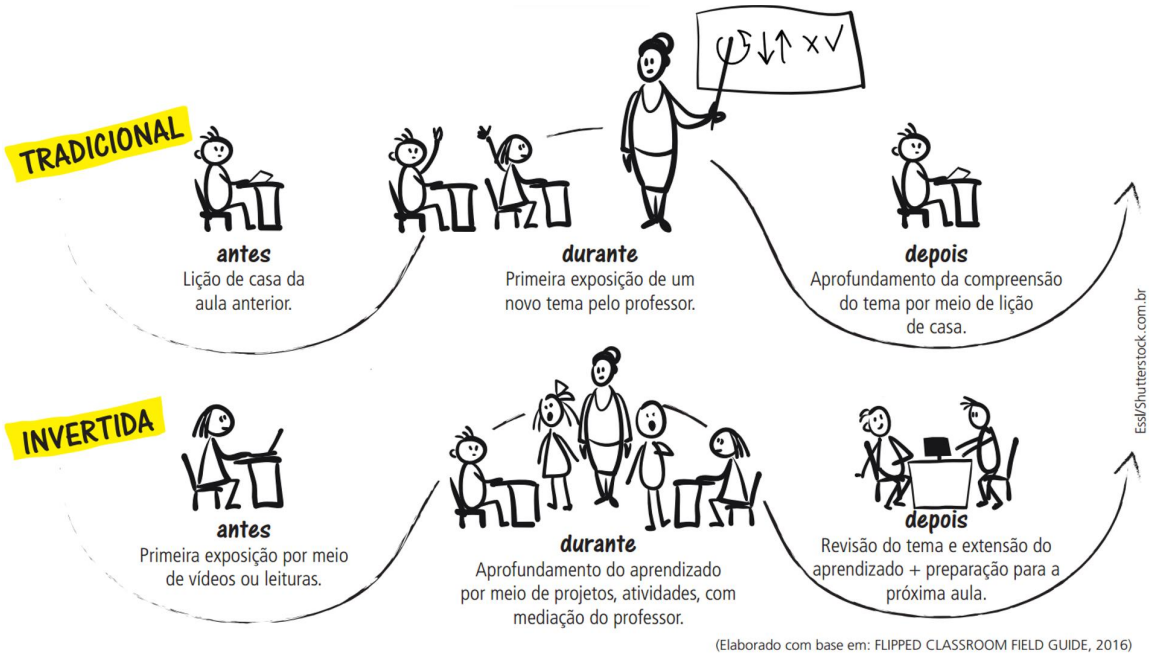

Figura 4.1: Aula invertida em comparação com o método tradicional. Apresentada por Castellar [25].

Inicialmente, apresentaremos as propostas de exercícios de programação utilizando a planilha eletrônica, permitindo que os alunos pesquisem e se familiarizem com o tema. Normalmente, essa etapa ocorrerá na última aula da proposta anterior. Nas aulas seguintes, daremos continuidade à discussão dos conceitos matemáticos e à resolução dos exercícios. A aplicação dessa metodologia, mesmo que de forma parcial, tem como objetivo promover uma maior interação entre alunos e professor, além de motivar os alunos a buscar mais informações sobre o tema. Conforme comentário extraído de Castellar [25], referenciando Lage, Platt e Treglia (2000), sobre a metodologia de sala de aula invertida:

> "Invertendo a sala de aula: um portal para a criação de um ambiente de aprendizagem inclusiva"(2000), os economistas Maureen Lage, Glenn Platt e Michael Treglia relatam uma experiência levada a cabo por eles com os estudantes de uma disciplina de Microeconomia em uma universidade pública de Ohio (Universidade de Miami) em 1996. Havia uma grande discrepância entre os estilos de aprendizagem dos alunos e os estilos de ensino dos professores, o que resultava em menor interesse (para não falar em desinteresse) nas aulas, por parte dos alunos... O resultado, comparado a outras turmas da mesma disciplina, foi a percepção de maior motivação dos alunos envolvidos na experiência, tanto por parte dos professores responsáveis quanto pela própria avaliação da disciplina feita pelos estudantes.

Dessa forma, apresentaremos algumas propostas de atividades que visam contemplar

as recomendações dos documentos norteadores, com foco na integração entre a matemática e a Tecnologia.

As propostas apresentadas podem ser aplicadas a partir do Ensino Fundamental II. Elas não estão classificadas para uma série específica, pois podem ser adaptadas para contemplar a série em que se deseja desenvolver a atividade.

4.1 Proposta 01: Cálculo do MDC com o Algoritmo de Euclides

4.1.1 Tema da Aula:

Utilização de Planilhas Eletrônicas para o Cálculo do MDC com o Algoritmo de Euclides.

4.1.2 Objetivos Gerais Abordados nesta Aula:

Compreender o significado de algoritmo e o conceito de Máximo Divisor Comum (MDC). Explorar o Algoritmo de Euclides por meio de cálculos realizados manualmente. Utilizar recursos computacionais, como planilhas eletrônicas, para automatizar o cálculo do MDC com o Algoritmo de Euclides.

4.1.3 Desenvolvimento do Tema:

Primeiro Momento

A fim de tornar o conteúdo mais atrativo e interessante para o aluno desde o início, é importante que ele tenha um contato direto com a planilha eletrônica, conferindo um aspecto mais dinâmico e envolvente ao início e desenvolvimento das atividades.

Inicialmente, será realizada uma explanação introdutória sobre o uso de planilhas eletrônicas, destacando suas principais funcionalidades. Os alunos aprenderão a inserir fórmulas matemáticas, representar operações básicas e utilizar a fórmula

```
=MOD(A1;B1),
```

que retorna o resto da divisão de `A1` por `B1`. Além disso, será apresentada a função de teste lógico

```
=IF(teste_logico;[valor_se_verdadeiro];[valor_se_falso]),
```

que permite realizar comparações lógicas entre um valor e o resultado esperado. Para consolidar os conceitos, os alunos realizarão exercícios práticos que reforçam a lógica e a aplicabilidade no uso das planilhas eletrônicas.

Exemplo 4.1. *Como exemplo, o professor poderá confeccionar, junto com os alunos, uma tabuada automática. Ao inserir o valor da tabuada desejada, os cálculos serão realizados automaticamente. Por exemplo, se for desejada a tabuada do 5, o aluno deverá digitar o número 5, e os resultados serão gerados automaticamente, aplicando a mesma lógica para outros valores.* (Habilidades: EF69CO01, EF69CO02, EF69CO03, EF69CO04, EF69CO05, EF69CO06, EF07MA13, EF09MA08, EM13MAT301, EM13MAT401 e EM13MAT405)

A principal intenção desse exemplo é familiarizar o aluno com a utilização de elementos e fórmulas em planilhas eletrônicas, que geralmente são iniciadas com o símbolo de igualdade, como na multiplicação

```
=MULTIPLY(C2;1).
```

Além disso, será explorada a funcionalidade de clicar e arrastar, permitindo que a fórmula seja repetida automaticamente ao longo das células selecionadas. Por fim, o exercício demonstrará como fixar uma célula específica utilizando o símbolo `$` na fórmula, como em

```
=MULTIPLY($C$2;1).
```

Solução.

A princípio, criaremos o layout da planilha mesclando as células `A1`, `B1` e `C1`. Mesclaremos também as células `A2` e `B2`, definindo `C2` como a célula de entrada, onde digitaremos o valor desejado. É importante que o professor reserve um tempo para explicar de forma detalhada esses procedimentos, lembrando que muitos alunos podem estar tendo seu primeiro contato com o conceito e o uso de planilhas eletrônicas.

Uma vez definida a célula de entrada, replicaremos o valor ao longo da tabela para indicar qual tabuada está sendo calculada. O procedimento adotado será colocar, em função da célula de entrada, as células da coluna `B3` até `B12`. Para isso, basta digitar a função `=C2` na célula `B3`. Em seguida, apresentaremos a função de clicar e arrastar até `B12`, preenchendo as células.

Entretanto, como feito até aqui, a planilha apresentará um erro, pois o programa entenderá que as demais células devem ser preenchidas com uma sequência, partindo de `=C2` até `=C12`. No entanto, o desejado é que seja replicado apenas o valor de `C2` ao longo dessas células. Assim, a função inserida em `B3` deve ser ajustada utilizando o símbolo `$`,

sendo reescrita da seguinte maneira: `=$C$2`. A função do símbolo `$` é fixar a coluna `C` e a linha `2`, de forma que, ao arrastar, o valor de `=$C$2` seja mantido ao longo das demais células.

De `B3` até `B12`, o layout será preenchido. Já de `C3` até `C12`, utilizaremos a função de multiplicação, que relaciona dois parâmetros: o valor de entrada e outro valor, fazendo o produto entre os dois, como `=MULTIPLY($C$2;1)`. Neste caso, o valor `$C$2` será multiplicado por `1`. Repetiremos o processo ao longo das células até a última célula, onde teremos o produto `=MULTIPLY($C$2;10)`. Assim, o exercício será concluído. □

Esse exemplo deverá ser confeccionado passo a passo com os alunos, desde a formatação das colunas até a inserção dos dados. Na planilha eletrônica apresentada, a célula de entrada será a `C2`. Ressalta-se que este exercício é apenas uma das várias possibilidades que podem ser exploradas utilizando a planilha eletrônica para a assimilação desses conceitos.

É recomendável que a planilha fique disponível para os alunos, permitindo que a utilizem como referência para explorar melhor o software. Assim, disponibilizá-la online é uma opção interessante, pois possibilita que os alunos verifiquem as fórmulas utilizadas e os demais passos realizados.

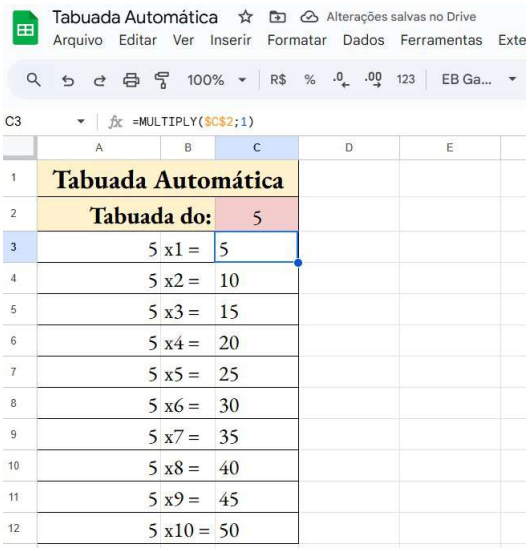

Figura 4.2: Exemplificação da planilha eletrônica da tabuada automática. Autoria própria. A planilha pode ser acessada clicando neste link

Com o término desse exercício, deve-se introduzir o tema do próximo. Nesse momento, é importante apresentar o novo exercício, que consiste em utilizar a planilha eletrônica para programar o cálculo do MDC utilizando o Algoritmo de Euclides. Além disso, solicitar que os alunos pesquisem e relembrem o conceito de MDC, a definição de algoritmo, bem como investiguem sobre Euclides, sua importância no desenvolvimento

da matemática e o funcionamento de seu algoritmo.

Segundo Momento

Nesse momento, será proposto um exercício em que os estudantes utilizem a planilha eletrônica para programar o cálculo do MDC utilizando o Algoritmo de Euclides.

Intencionalmente, o exercício será apresentado antes da discussão do conteúdo matemático, como uma forma de desafio e de exploração da metodologia ativa da sala de aula invertida. Embora seja muito provável que poucos alunos, ou nenhum, tenham tido contato prévio com esse algoritmo, o objetivo principal é despertar a curiosidade e o interesse dos alunos em compreender o conceito matemático necessário para realizar a programação.

Alguns alunos podem pesquisar e encontrar a fórmula pronta para a determinação do MDC, como

```
=GCD(A1; B1)
```

no Google Sheets, entre dois números. Nesse momento, o professor deve incentivar a pesquisa e a utilização dessa fórmula, mas deve reforçar que o exercício exige a implementação do cálculo do MDC utilizando uma abordagem específica: o **Algoritmo de Euclides**.

Além disso, é fundamental enfatizar que os estudantes devem criar campos específicos para a inserção dos dois números, garantindo que os cálculos subsequentes sejam realizados de forma automática. Isso inclui implementar a lógica necessária para a parada do algoritmo, assegurando uma programação funcional e eficiente.

Exercício 4.1. *Utilizando uma planilha eletrônica, programe o cálculo do MDC para dois números inteiros usando o Algoritmo de Euclides. A planilha deve conter campos para a entrada dos dois números e implementar a lógica completa do algoritmo, incluindo a condição de parada.* (Habilidades: EF69CO01, EF69CO02, EF69CO03, EF69CO04, EF69CO05, EF69CO06, EF06MA04, EF06MA23, EF07MA03, EF07MA04, EF07MA05, EM13MAT101, EM13MAT102, EM13MAT104 e EM13MAT405)

Terceiro Momento

Após a apresentação do exercício, é aconselhável que o professor inicie a aula promovendo a seguinte indagação: "Como se calcula o MDC utilizando o Algoritmo de Euclides?"Para responder a essa questão, o professor deve começar explicando a definição de Máximo Divisor Comum (MDC) e o conceito de algoritmo.

É importante incluir uma breve explanação sobre Euclides e sua relevância para o desenvolvimento da matemática, seguida da introdução ao Algoritmo da divisão, ao Lema de Euclides e, posteriormente, ao Algoritmo de Euclides. A aula deve incluir a realização de exercícios que estimulem a compreensão lógica do algoritmo, além de cálculos manuais que explorem seu funcionamento de forma prática.

Exemplos que podem ser desenvolvidos com os alunos: no primeiro exercício, utilizaremos um exemplo apresentado por Hefez no Programa de Iniciação Científica da OBMEP [2].

Exemplo 4.2. *Determine o Máximo Divisor Comum (MDC) de* 372 *e* 162. *Primeiramente, utilize o Algoritmo de Euclides, em seguida, o Lema de Euclides.* (Habilidades: EF06MA04, EF07MA05, e EM13MAT103)

Solução.
Usando o Algoritmo de Euclides: a ideia é que essa solução seja apresentada no quadro branco, mostrando o preenchimento da tabela passo a passo.

372	162	48	18	12	6
48	18	12	6	0	

Portanto, o $\text{mdc}(372, 162) = 6$.

Usando o Lema de Euclides, temos:

$$\begin{aligned}
\text{mdc}(372, 162) &= \text{mdc}(372 - 162 \cdot 2, 162) = \text{mdc}(48, 162) = \text{mdc}(48, 162 - 48 \cdot 3) \\
&= \text{mdc}(48, 18) = \text{mdc}(48 - 18 \cdot 2, 18) = \text{mdc}(12, 18) \\
&= \text{mdc}(12, 18 - 12) = \text{mdc}(12, 6) = \text{mdc}(12 - 6 \cdot 2, 6) \\
&= \text{mdc}(0, 6) = 6.
\end{aligned}$$

□

O próximo exemplo foi retirado do portal da OBMEP [4] e faz parte de um módulo de exercícios indicado para alunos do $6^{\underline{o}}$ ano do Ensino Fundamental.

Exemplo 4.3. *Uma folha de papel retangular de* 360 *mm de comprimento por* 210 *mm de largura deve ser quadriculada, com todos os quadrados idênticos, como em um tabuleiro de xadrez. Se a quantidade de quadrados deve ser a menor possível e a medida dos lados precisa ser um número inteiro, em milímetros, qual deve ser essa medida?* (Habilidades: EF06MA06, EF07MA01, EF07MA05, EM13MAT101 e EM13MAT103)

Solução.
Para resolver o exercício, basta calcular o MDC entre 360 e 210. Para isso, utilizaremos o Lema de Euclides:

$$\begin{aligned}\text{mdc}(360,210) &= \text{mdc}(360-210\cdot 1,210) = \text{mdc}(150,210) = \text{mdc}(150,210-150\cdot 1)\\ &= \text{mdc}(150,60) = \text{mdc}(150-60\cdot 2,60) = \text{mdc}(30,60)\\ &= \text{mdc}(30,60-30\cdot 2) = \text{mdc}(30,0)\\ &= \text{mdc}(0,30) = 30.\end{aligned}$$

Portanto, a medida dos lados dos quadrados deve ser 30 mm. □

Seguindo a ideia discutida nos exemplos, solicitaremos que os alunos pratiquem e fixem o conceito realizando os seguintes exercícios:

Exercício 4.2 (Extraído do ENEM 2015, adaptado, [4]). *O gerente de um cinema fornece anualmente ingressos gratuitos para escolas. Este ano, serão distribuídos* 400 *ingressos para uma sessão vespertina e* 320 *ingressos para uma sessão noturna de um mesmo filme. Várias escolas podem ser escolhidas para receber os ingressos, mas há alguns critérios para a distribuição:*

1. *Cada escola deverá receber ingressos para uma única sessão.*
2. *Todas as escolas contempladas deverão receber o mesmo número de ingressos.*
3. *Não haverá sobra de ingressos (ou seja, todos os ingressos serão distribuídos).*

O número mínimo de escolas que podem ser escolhidas para obter ingressos, segundo os critérios estabelecidos, é:

a) 2. *b) 4.* *c) 9.* *d) 40.* *e) 80.*

(Habilidades: EM13MAT314, EM13MAT301, EM13MAT304, EF06MA06, EF07MA01 e EM13MAT201)

Ao término da discussão e realização dos exercícios, solicite que os alunos apliquem as ideias discutidas para resolver o exercício de programação. É importante destacar que não é necessário que os alunos comecem a pensar na programação apenas após a realização dos exercícios. Ao longo das aulas de explanação do conteúdo, o professor deve estimular os alunos a pesquisarem sobre o tema e a tentarem realizar o exercício de programação de forma progressiva.

Uma vez concluída a programação da planilha anterior, deve-se orientar os alunos a realizar uma pesquisa sobre os principais aspectos da próxima proposta. É importante

apresentar o próximo exercício: a utilização da Cifra de César e sua aplicação. Nesse contexto, recomenda-se solicitar que os alunos investiguem como funciona a Cifra de César, explorando seu contexto histórico, e incentivá-los a identificar possíveis relações entre a cifra e operações matemáticas.

4.2 Proposta 02: Aplicação da Cifra de César

4.2.1 Temas da Aula:

Programação da Cifra de César em uma planilha eletrônica.

4.2.2 Objetivos Gerais Abordados neste tema:

Compreensão do funcionamento da Cifra de César e sua relação com a matemática, além da utilização de recursos tecnológicos para a programação da cifra.

4.2.3 Desenvolvimento do Tema:

Primeiro Momento

Com a mesma intenção da proposta anterior, primeiramente apresentaremos o problema para incentivar os estudantes, que deverão aplicar as ideias pesquisadas e discutidas para criar um programa utilizando uma planilha eletrônica. O objetivo é desenvolver um código capaz de criptografar e descriptografar uma mensagem com até 120 caracteres, utilizando a Cifra de César e considerando os espaços entre as palavras como caracteres válidos.

Exercício 4.3. *Usando uma planilha eletrônica, desenvolva um programa capaz de criptografar e descriptografar mensagens utilizando a Cifra de César. A mensagem deve conter pelo menos* 120 *caracteres, considerando também os espaços como parte da contagem.*

Além disso, implemente um campo de entrada que permita definir o ciclo de deslocamento a ser utilizado na cifra. No exemplo apresentado, o deslocamento padrão foi 3*, mas o valor máximo do deslocamento será determinado com base no número de caracteres definido por cada aluno. O alfabeto utilizado deve ter, no mínimo,* 27 *caracteres, incluindo as letras sem distinção entre maiúsculas e minúsculas, além de um caractere reservado para o espaço.* (Habilidades: EF69CO01, EF69CO02, EF69CO03, EF69CO04, EF69CO05, EF69CO06, EF07MA05, EM13MAT101, EF07MA18, EF09MA08, EM13MAT406, EM13MAT203,EM13MAT315 e EM13MAT314)

Após a conclusão da primeira parte do exercício, pode-se propor o segundo exercício utilizando a planilha programada, conforme descrito a seguir:

Exercício 4.4. *Usando a planilha eletrônica criada no exercício anterior, proponha que os alunos se dividam em pares. Cada dupla deve realizar a seguinte atividade: um dos alunos enviará uma mensagem criptografada contendo* 120 *caracteres, sem revelar o ciclo de deslocamento utilizado na cifra. O outro aluno, utilizando o método de análise de frequência das letras, deverá tentar decifrar a mensagem.*

(Habilidades: EF69CO01, EF69CO02, EF69CO03, EF69CO04, EF69CO05, EF69CO06, EM13MAT101, EM13MAT103, EM13MAT405, EM13MAT402, EF08MA24, EF07MA36, EF06MA33 e EF07MA15)

Segundo Momento

Apresentar uma introdução ao contexto histórico e ao funcionamento da Cifra de César. Uma sugestão didática é dividir a turma em duplas e propor que um integrante criptografe uma mensagem enquanto o outro a descriptografa. Dois aspectos importantes devem ser enfatizados. A relação entre a quantidade de letras do alfabeto e o processo de criptografia. Elabora perguntas que induzam os alunos a percepção das operações matemáticas envolvidas no algoritmo, destacando conceitos como deslocamento e ciclos.

Terceiro Momento

Propor e discutir exemplos de exercícios que proporcionem maior fixação da ideia de ciclos, e ao menos um outro exercício que relacione a Cifra de César com o conceito de análise de frequência. Aqui, citaremos dois exemplos, embora existam muitos outros que podem ser realizados para essa finalidade.

Exemplo 4.4 (Retirado do Exame Nacional do Ensino Médio - ENEM 2021)**.** *A Cifra de César é um exemplo de método de codificação de mensagens usado por Júlio César para se comunicar com seus generais. Nesse método, cada letra é trocada por uma letra que aparece no alfabeto um número fixo de posições à frente (ou atrás), de forma cíclica. A seguir, temos um exemplo onde cada letra é substituída por aquela que está três posições à frente.*

Original	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
Codificado	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C

Para quebrar um código como esse, a análise de frequência das letras de um texto é

uma ferramenta importante. Uma análise do texto do romance O Guarani, de José de Alencar, composto por 491631 *letras, gerou o seguinte gráfico de frequências:*

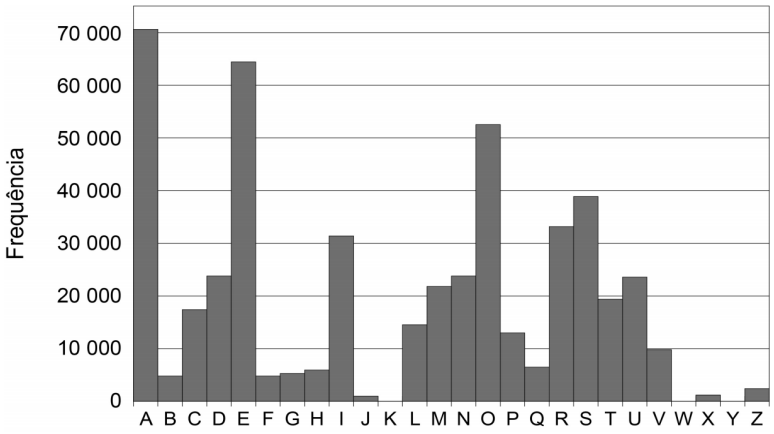

Figura 4.3: Gráfico retirado do ENEM 2021, 2º dia.
Link para a prova.

Após codificar esse texto com a regra do exemplo fornecido, realiza-se uma nova análise de frequência no texto codificado. As quatro letras mais frequentes, em ordem decrescente de frequência, no texto codificado são:

a) A, E, O e S. *b) D, E, F e G.* *c) D, E, F e G.* *d) D, H, R e V.*
e) X, B, L e P.

(Habilidades: EF09MA23 EF09MA21, EM13MAT406 e EM13MAT507)

Solução.

Ao analisar a figura apresentada, identificamos que as quatro letras mais frequentes, em ordem decrescente, são *A, E, O* e *S*. Contudo, é importante destacar que o enunciado solicita a codificação dessas letras, seguindo a regra da Cifra de César. Essa regra determina que cada letra deve ser deslocada no alfabeto por um número fixo de posições, conforme indicado na tabela de conversão/codificação.

Ao aplicar a regra com o deslocamento especificado no problema, as letras *A, E, O* e *S* são transformadas em *D, E, F* e *G*, respectivamente.

Assim, a solução correta para o exercício é a sequência ***D, E, F*** e ***G***. □

Exemplo 4.5 (Retirado da OBMEP. Nível 3 - 2003)**.** *Considere a sequência oscilante:* $1, 2, 3, 4, 5, 4, 3, 2, 1, 2, 3, 4, 5, 4, 3, 2, 1, 2, 3, 4, \ldots$. *O* 2003^{o} *termo desta sequência é:*

A) 1. *B) 2.* *C) 3.* *D) 4.* *E) 5.*

(Habilidades: EF08MA10, EF08MA11, EM13MAT507 e EF06MA03)

Solução.

Para compreender este exercício e extrair dele a ideia que utilizaremos, enfatize que, independentemente da posição pedida, o valor será sempre um dos seguintes algarismos: 1, 2, 3, 4, 5, 4, 3, 2. Além disso, destaque a importância de observar a posição que cada número ocupa na sequência.

Após essa explicação inicial, explore a seguinte ideia: escreva uma sequência menor, por exemplo, com 11 dígitos. Mostre que a sequência completa (com 8 dígitos) cabe exatamente uma vez, sobrando 3 dígitos. Conclua que a posição do $11^{\underline{o}}$ termo equivale à $3^{\underline{a}}$ posição, ou seja, é o número 3. Em seguida, escreva uma sequência com 19 dígitos e mostre que os 8 algarismos se repetem exatamente duas vezes, restando novamente 3 dígitos. Mais uma vez, conclua que a posição do $19^{\underline{o}}$ termo equivale à $3^{\underline{a}}$ posição, ou seja, é o número 3.

A intenção é induzir os alunos a perceberem que basta dividir a posição desejada pelo número de elementos da repetição (neste caso, 8). O resto da divisão indica a posição do algarismo na sequência repetida.

Assim, para determinar o $2003^{\underline{o}}$ termo da sequência, dividimos 2003 por 8. O quociente indica o número de repetições completas da sequência, e o resto será a posição do algarismo correspondente. Como o resto da divisão é 3, o $2003^{\underline{o}}$ termo corresponde ao terceiro elemento da sequência que se repete, ou seja, 3. □

Exemplo 4.6. *As contribuições para o desenvolvimento da matemática moderna do francês Pierre de Fermat são inúmeras. Ele é conhecido como um dos fundadores da teoria dos números, além de ter contribuído em diversas outras áreas da matemática. Fermat também formulou o famoso Último Teorema de Fermat, que desafiou gerações de matemáticos.*

Considerando a seguinte sequência `FERMATFERMATFERMATFERMATFERMATF...`*, na qual a palavra* `FERMAT` *se repete várias vezes, responda às perguntas a seguir:*

a) Que letra ocupa a 211 *posição?*

b) Quantas vezes a palavra `FERMAT` *foi repetida até a* 211 *letra?*

(Habilidades: EF08MA10, EF09MA08, EF06MA07, EF06MA03, EM13MAT507 e EM13MAT501)

Solução.

a) Neste exemplo, como no exercício anterior, é importante explorar a ideia de periodicidade e ciclos, ajudando os alunos a perceberem a relação entre o período formado pela quantidade de letras da palavra e sua aplicação na solução do exercício.

   A solução consiste em dividir o valor 211 por 6, que é a quantidade de letras da palavra `FERMAT`. Escreva o resultado da divisão no formato do algoritmo da divisão: $211 = 6 \cdot 35 + 1$, ou seja, o resto dessa divisão é 1. Assim, a letra que está na posição $211^{\underline{a}}$ é equivalente à primeira letra da palavra, que é `F`.

   É interessante explorar outras situações como forma de provocação, perguntando, por exemplo: "E se o resto fosse zero? Qual seria a letra?"

b) Quantas vezes a palavra `FERMAT` foi repetida até a $211^{\underline{a}}$ letra? Novamente, utilize o algoritmo da divisão para explicar. A expressão $211 = 6 \cdot 35 + 1$ indica que a palavra de 6 letras foi repetida exatamente 35 vezes até alcançar a $211^{\underline{a}}$ letra.

□

Ao término da realização e discussão dos exercícios relacionados à Cifra de César, oriente os alunos a aplicar os conceitos aprendidos para resolver o exercício de programação utilizando a Cifra de César.

Uma vez concluída a atividade com a Cifra de César, deve-se introduzir a próxima proposta: o uso da Tabela de Vigenère para codificar e decodificar mensagens. Nesse momento, recomenda-se que os alunos pesquisem sobre a Tabela de Vigenère, sua relevância no desenvolvimento da criptografia e como essa técnica incorpora operações matemáticas, como a soma modular. Incentive-os a comparar as duas cifras, identificando suas semelhanças, vantagens e limitações.

4.3 Proposta 03: Construção de uma Planilha para Criptografia com Vigenère

4.3.1 Temas da Aula:

Programação usando a Tabela de Vigenère para codificar e decodificar mensagens.

4.3.2 Objetivos Gerais Abordados neste tema:

Compreensão do funcionamento da Tabela de Vigenère para codificar e decodificar mensagens e sua relação com a matemática, além da utilização de recursos tecnológicos

para a programação da cifra.

4.3.3 Desenvolvimento do Tema:

Primeiro Momento

Seguindo a ideia das propostas anteriores, primeiramente apresentaremos o problema para incentivar os estudantes, que deverão aplicar as ideias discutidas na criação da planilha da Cifra de César para desenvolver um programa utilizando uma planilha eletrônica capaz de codificar e decodificar mensagens com a Tabela de Vigenère. A abordagem para desenvolver esse programa é muito semelhante à utilizada no desenvolvimento da Cifra de César. O objetivo é implementar um código que permita codificar e decodificar mensagens utilizando a Tabela de Vigenère, considerando a possibilidade de trabalhar com mensagens de até 120 caracteres e incluindo os espaços entre as palavras como caracteres válidos.

Exercício 4.5. *Crie uma planilha eletrônica, capaz de criptografar e descriptografar mensagens utilizando o conceito da Tabela de Vigenère. A mensagem deve conter, no mínimo,* 120 *caracteres, considerando também os espaços como parte da contagem.*

Além disso, implemente um campo de entrada que permita definir a palavra-chave para a criptografia e descriptografia, com até 12 letras. O número máximo de linhas e colunas da planilha será determinado em função do número de caracteres definido por cada aluno. O alfabeto utilizado deve conter, no mínimo, 27 *caracteres, incluindo as letras (sem distinção entre maiúsculas e minúsculas) e um caractere reservado para o espaço.* (Habilidades: EF69CO01, EF69CO02, EF69CO03, EF69CO04, EF69CO05, EF69CO06, EF06MA33, EF09MA22, EF09MA23, EF08MA10, EM13MAT314, EM13MAT406, EM13MAT203 e EM13MAT315)

Segundo Momento

Apresente uma introdução ao contexto histórico e ao funcionamento da Tabela de Vigenère. Uma abordagem didática eficiente é dividir a turma em duplas, incentivando cada dupla a criar sua própria tabela baseada na Tabela de Vigenère. Essa tabela pode conter mais ou menos caracteres, dependendo da proposta. Proponha que um integrante criptografe uma mensagem enquanto o outro a descriptografa, promovendo a compreensão prática do processo.

É fundamental destacar alguns aspectos essenciais que devem ser enfatizados durante a atividade: a relação entre a quantidade de caracteres no alfabeto e o número de conjuntos de caracteres defasados que compõem a tabela, bem como sua ligação com

o processo de criptografia. Elabore perguntas que estimulem os alunos a compreenderem as operações matemáticas envolvidas na defasagem dos alfabetos, ressaltando conceitos como deslocamento, ciclos e sua conexão com a aritmética modular.

A solução de um exemplo é muito útil para ilustrar o processo. Para isso, considere a tabela apresentada no Exemplo 1.1 do primeiro capítulo. Usando a criptografia da Tabela de Vigenère, a mensagem "PROFMAT NA UFMT" é codificada com a palavra-chave "FERMAT". Como resultado, obtemos a palavra criptografada "UVFRMTYRRGFFY".

É importante apresentar exemplos como este de forma detalhada, para que os alunos compreendam melhor o mecanismo de funcionamento. Além disso, recomenda-se destacar a associação com a aritmética modular e o cálculo de restos, evidenciando os conceitos matemáticos envolvidos no processo de codificação e decodificação.

Exemplo 4.7. *Utilizando o conceito da Tabela de Vigenère (Tabela 1.3) e operações matemáticas para produzir um algoritmo mais eficiente, codifique e decodifique a mensagem "PROFMAT NA UFMT" com a palavra-chave "FERMAT".*

(Habilidades: EF09MA08, EF06MA33, EF07MA36, EF07MA18, EF08MA10,EM13MAT315, EM13MAT203 e EM13MAT314)

Solução.

Note que, de forma mais prática, podemos realizar o seguinte procedimento. Com base no exercício citado, consideraremos a seguinte tabela que associa cada letra do alfabeto a um número:

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

Tabela 4.3: Associação das Letras do Alfabeto com Números

Comece demonstrando o processo de codificação com um exemplo simples. Pegue a primeira letra da palavra que desejamos converter, a letra `P` (que possui o valor 15 na tabela), e some ao valor correspondente à primeira letra da palavra-chave, `F` (valor 5). Quando somamos $15 + 5 = 20$ e dividimos o total pelo número de letras do alfabeto (26), o resto da divisão é 20. Na tabela, o número 20 corresponde à letra `U`, que será o valor criptografado.

Repita o mesmo processo para a próxima letra. A letra `R` (valor 17) somada à letra `E` (valor 4) resulta em 21. Na tabela, o número 21 corresponde à letra `V`. Assim, continuamos codificando cada letra da mensagem usando a palavra-chave.

Agora, para descriptografar, o processo será inverso. Pegue o valor criptografado, como 21 (letra `V`), e subtraia o valor correspondente à letra da palavra-chave, `E` (valor 4). Subtraindo $21 - 4 = 17$, o número 17 corresponde novamente à letra `R`, restaurando a mensagem original.

Esse processo simples de soma e subtração modular permite que as mensagens sejam codificadas e decodificadas com precisão, e pode ser facilmente expandido para mensagens maiores e palavras-chave mais longas. □

A partir da discussão sobre solução deste exercício, os alunos podem direcionar sua atenção para a resolução do Exercício 4.5, aplicando os conceitos e estratégias exploradas.

Ao concluir a proposta anterior, o professor deve introduzir a ideia da próxima atividade, contextualizando-a como um desafio prático. A proposta seguinte será uma tentativa de quebrar os códigos da criptografia RSA, a ideia é simular um cenário de investigação. Recomenda-se que os alunos: Pesquisem sobre os fundamentos matemáticos da criptografia RSA. Assistam ao filme *O Jogo da Imitação* (2014), que aborda a quebra de códigos na Segunda Guerra Mundial, para entender a relevância histórica e social da criptoanálise.

4.4 Proposta 04: Decifrando os Segredos da Criptografia RSA

4.4.1 Tema da Aula

Compreender o método de fatoração de Fermat para números semiprimos e sua aplicação prática, implementando o algoritmo em planilhas eletrônicas.

4.4.2 Objetivos Gerais Abordados neste Tema

Analisar o contexto histórico da evolução da criptografia, seu desenvolvimento e sua importância até os dias atuais. Compreender a relevância do método de criptografia RSA e seu princípio de funcionamento. Revisar a definição de números primos, compreender o conceito de paridade dos números e a relação entre os números primos e a segurança desse sistema criptográfico. Além disso, estudar o algoritmo de fatoração de Fermat, seus critérios de parada e implementá-lo em uma planilha eletrônica para a fatoração de números semiprimos.

4.4.3 Desenvolvimento do Tema

Primeiro Momento

Seguindo a ideia das demais propostas, o professor deve, inicialmente, apresentar o novo exercício aos alunos, estimulando neles o espírito de investigação. A ideia é que implementem o algoritmo de Fermat, de forma que ele seja capaz de fatorar um número semiprimo qualquer com até seis algarismos.

Exercício 4.6. *Crie uma planilha eletrônica que realize automaticamente a fatoração de um número semiprimo ímpar, n, ou seja, o produto de dois números primos que resulte em um número com até seis algarismos, utilizando o método de fatoração de Fermat. A planilha deve conter um campo de entrada para a inserção do número a ser fatorado. Caso o número seja par, o menor fator será, evidentemente, 2. O algoritmo deve funcionar corretamente para qualquer valor n no intervalo $1 \leq n \leq 999\,999$. Além disso, a planilha deve obedecer aos critérios de parada do algoritmo e identificar se o número digitado é primo ou composto; se composto, deverá exibir os fatores p e q.*
Exemplo de teste:
Para $n = 100\,489$, a planilha deve retornar $p = q = 317$ (quadrado de um primo).
Para $n = 15$, deve retornar $p = 5$ e $q = 3$.

(Habilidades: EF69CO01, EF69CO02, EF69CO03, EF69CO04, EF69CO05, EF69CO06,EF06MA02, EF06MA05, EF06MA03, EF07MA01, EF07MA18, EF08MA02, EF07MA08, EF06MA33, EF09MA18, EF09MA09, EM13MAT314, EM13MAT315, EM13MAT203, EM13MAT405 e EM13MAT301)

Segundo Momento

Fazer uma breve recapitulação do contexto histórico dos outros métodos de criptografia discutidos até aqui, evidenciando que todos utilizavam o método de chave simétrica, em que a mesma chave é empregada tanto para criptografar quanto para descriptografar. Destacar que essa dinâmica representa uma grande fragilidade, pois a exposição da chave é um risco significativo para a segurança dos métodos criptográficos.

É interessante que os alunos tenham assistido ao filme *O Jogo da Imitação* (2014), que aborda a quebra de códigos na Segunda Guerra Mundial, para que possam compreender melhor a relevância histórica e social da criptoanálise. Além disso, o filme pode contribuir para estimular o interesse pelo tema, proporcionando uma imersão em um ambiente de investigação e pesquisa.

Terceiro Momento

Por meio de perguntas provocativas, instigar os alunos a refletirem: como é possível resolver essa fragilidade? Como restringir o acesso à chave criptográfica? E se, em vez de uma, utilizássemos duas chaves, de forma que cada pessoa tivesse no mínimo duas chaves: uma para encriptar e outra para descriptografar? Isso poderia ajudar a resolver o problema?

Gradativamente e de maneira sutil, o professor pode responder às questões reformulando-as: como isso poderia ajudar a resolver o problema? A ideia aqui é levar os alunos a compreenderem o conceito de chave pública e chave privada, reconhecendo que a chave pública, como o próprio nome sugere, é de acesso a todos, enquanto a chave privada deve ser mantida em sigilo. Explicar que a chave pública contém um parâmetro com uma informação oculta, um número semiprimo formado pelo produto de dois números primos. Aqui, é importante garantir que os alunos compreendam que a quebra da chave consiste na fatoração desse número composto.

A partir dessa introdução, o professor pode apresentar exemplos de exercícios que aumentem gradativamente o nível de dificuldade da fatoração, abordando critérios de divisibilidade e aprimorando a compreensão sobre números primos.

A discussão desse exercício deve servir como base para que os alunos revisem e analisem critérios de divisibilidade, além de relembrar o uso do Crivo de Eratóstenes para determinar se um número é primo.

Exemplo 4.8 (Extraído do Módulo Resolução de Exercícios, ver [18]). *Considere as afirmativas:*

I - *Todo número múltiplo de 5 e 15 é múltiplo de 75.*

II - *O número 1147 não é primo.*

III - *Todo o número da forma abba, em que a e b são algarismos, é divisível por 11.*

IV - *O número de divisores naturais de 576 é divisor de 63.*

(Habilidades: EF09MA03, EF06MA05, F06MA06, EF07MA01, EF06MA07, EF07MA16, EM13MAT315 e EM13MAT501 EF09MA04)

Solução.

I - A ideia neste item é revisar pelo menos os critérios básicos de divisibilidade. Mesmo que o exercício não os solicite, é interessante que o professor comente, ao menos, os critérios de divisibilidade por 2, 3, 4, 5 e 9.

Para resolver o exercício, explique aos alunos que o objetivo aqui é utilizar um contraexemplo, ou seja, encontrar um número que seja múltiplo de 5 e de 15, mas que

não seja múltiplo de 75. Nesse sentido, pode-se usar o próprio número 15, que é múltiplo de 5 e de si mesmo, mas não é múltiplo de 75. Logo, a afirmação **I** é falsa.

II - Antes de resolver este item, dê um tempo para que os alunos tentem fatorar esse número, que é um semiprimo com fatores de dois algarismos. Desde o início, eles perceberão a dificuldade de encontrar esses números primos.

Comece explorando a ideia de potências de números com base 10, pois a multiplicação é mais intuitiva e direta, permitindo que o aluno compreenda como pode refinar a busca pelos fatores de um número ou determinar se ele é primo. Nesse sentido, podemos escrever: $10^2 < 20^2 < 30^2 < 40^2$. Pode-se afirmar que o número que procuramos está no intervalo $30^2 < \ \mathbf{1147} < 40^2$, que pode ser reduzido a $33^2 < \mathbf{1147} < 34^2$. Mostre que:

$$33^2 = 1089 < \mathbf{1147} < 1156 = 34^2.$$

Após isso, use o Crivo de Eratóstenes para verificar se 1147 é um número primo ou composto. Para isso, basta dividir 1147 por todos os primos menores que 33. Se uma dessas divisões for exata, então 1147 é composto; caso contrário, é primo. Fazendo essa verificação, encontramos $1147 = 31 \times 37$, ou seja, 1147 não é um número primo. Logo, a afirmação **II é falsa.**

Antes de seguir para o próximo item, é importante chamar a atenção dos alunos para o fato de que o processo de fatoração utilizado aqui está otimizado. Em vez de dividir 1147 por todos os números menores que ele para verificar se é primo, reduzimos essa análise para números menores que 33. Mesmo com essa otimização, a fatoração manual ainda demandaria um tempo considerável. Essa ideia pode ser útil na programação da planilha eletrônica.

III - Todo número da forma *abba*, em que a e b são algarismos, é divisível por 11.

Explique o procedimento para verificar se um número é divisível por 11. Um número é divisível por 11 se a diferença entre a soma dos dígitos nas posições ímpares e a soma dos dígitos nas posições pares for um múltiplo de 11. Apresente um exemplo numérico para que a ideia fique mais clara.

Por exemplo, considere o número 6172947: $6 - 1 + 7 - 2 + 9 - 4 + 7 = 22$. Como 22 é divisível por 11, então 6172947 também será. De fato:$561177 \times 11 = 6172947$.

Utilizaremos essa ideia no item proposto para um número da forma *abba* (com quatro dígitos):

- Posições ímpares ($1^{\underline{a}}$ e $3^{\underline{a}}$): $a + b$ - Posições pares ($2^{\underline{a}}$ e $4^{\underline{a}}$): $b + a$

A diferença será:$(a+b)-(b+a) = 0$. Como 0 é múltiplo de 11, todo número da forma *abba* é divisível por 11. Portanto, a afirmação **III é verdadeira**.

IV - O número de divisores naturais de 576 é um divisor de 63.

Fatorando 576, temos $576 = 2^6 \cdot 3^2$. A partir dessa expressão, calculamos a quantidade de divisores de 576 da seguinte maneira: acrescenta-se 1 a cada expoente dos fatores primos e, em seguida, multiplicamos os resultados, ou seja, $(6+1) \cdot (2+1) = 7 \cdot 3 = 21$. Isso significa que 576 tem 21 divisores. Como $21 \times 3 = 63$, pode-se afirmar que o número de divisores de 576 divide 63. Portanto, **IV é verdadeira.**

É interessante chamar a atenção dos alunos, comparando a diferença entre o número de divisores deste caso e o da **alternativa I**. □

A ideia do próximo exemplo é estimular a habilidade de decompor números, levando os alunos a manipular fatores primos para contabilizar o número de dígitos, compreender a estrutura de números grandes e determinar a quantidade de dígitos de um número.

Exemplo 4.9 (Extraído do Módulo Resolução de Exercícios, ver [5])**.** *Para registrar o resultado da operação* $2^{101} \times 5^{97}$*, o número de dígitos necessários é:*

a) 96 *b)* 97 *c)* 98 *d)* 99 *e)* 100

(Habilidades: EF06MA11, EF09MA04, EF06MA02, EM13MAT301, EF09MA18, EF08MA02, EM13MAT304, EM13MAT313 e EF08MA01)

Solução.
Vamos reescrever a expressão:

$$\begin{aligned} 2^{101} \times 5^{97} &= 2^4 \times 2^{97} \times 5^{97} = 16 \times (2^{97} \times 5^{97}) = 16 \times 10^{97} \\ &= \underbrace{16}_{\text{2 dígitos}} \underbrace{00\ldots00}_{\text{97 zeros}}. \end{aligned}$$

Portanto, o número total de dígitos é $2 + 97 = 99$. □

Quarto momento

A ideia aqui é que o professor crie um ambiente de entusiasmo ao apresentar aos alunos uma ferramenta poderosa para quebrar códigos e desafiar a segurança do sistema RSA. Explicar que o método se baseia na fatoração de um número semiprimo, um conceito essencial para a criptografia moderna.

O professor deve iniciar a aula explicando o **Algoritmo de Fatoração de Fermat**, destacando sua lógica e seu funcionamento. Para isso, é importante abordar:

O conceito de números semiprimos, ressaltando que são formados pelo produto de dois números primos e que a segurança do sistema RSA depende da dificuldade de fatorá-los.

A estrutura do algoritmo, explicando que ele se baseia na tentativa de expressar o número semiprimo n como a diferença de dois quadrados:

$$n = x^2 - y^2 \Rightarrow (x - y)(x + y) = n$$

Os critérios de parada, esclarecendo as condições em que o algoritmo pode ser interrompido, seja porque n é primo ou porque os fatores foram encontrados.

Para consolidar a explicação, o professor pode resolver alguns exemplos com os alunos, demonstrando o passo a passo do método. Pode-se iniciar com números menores para ilustrar o raciocínio e, em seguida, aplicar o algoritmo a um número semiprimo maior, como 17367, utilizando a estrutura definida anteriormente no exemplo 2.6.

Após a discussão dos exemplos, os alunos podem direcionar sua atenção para a resolução do Exercício 4.6, aplicando os conceitos e as estratégias exploradas para implementação computacional. Essa abordagem não apenas reforça a compreensão do método, mas também estimula o interesse pela matemática e pela programação.

Por fim, o professor pode conduzir uma discussão sobre a relevância da fatoração de números grandes para a segurança digital, relacionando o tema com aplicações reais na criptografia moderna.

Considerações Finais

Este trabalho apresentou uma proposta alternativa para o ensino da matemática, explorando conceitos de criptografia e o uso de ferramentas computacionais acessíveis aos estudantes. A proposta busca tornar o aprendizado mais dinâmico e envolvente, além de demonstrar a aplicabilidade da matemática em contextos reais, aproximando os conteúdos escolares das demandas do mundo moderno. Dessa forma, espera-se que essa abordagem contribua para uma melhor compreensão da matemática, tornando o aprendizado mais significativo e motivador.

As atividades foram planejadas para serem implementadas pelos professores em sala de aula, permitindo que os alunos desenvolvam habilidades matemáticas por meio de exercícios práticos e interativos. Para isso, considerou-se a infraestrutura tecnológica disponível nas escolas, garantindo que a proposta seja viável dentro da realidade educacional do estado de Mato Grosso. Dessa maneira, os professores podem diversificar suas estratégias de ensino, tornando os conceitos matemáticos mais acessíveis e estimulantes para os estudantes.

Entre as metodologias sugeridas, destaca-se a utilização parcial da **sala de aula invertida**, uma abordagem que visa otimizar o tempo em sala e incentivar a participação ativa dos alunos na construção do conhecimento. No entanto, reconhece-se que essa mudança pode enfrentar resistência por parte dos estudantes. Caso isso ocorra, recomenda-se que os professores combinem o ensino tradicional para introdução dos conteúdos com o uso de planilhas eletrônicas para promover a exploração dos conceitos matemáticos.

A escolha da criptografia RSA como tema central justifica-se por sua relevância no mundo moderno e pelo seu potencial de conectar os conteúdos matemáticos ao cotidiano dos estudantes. A segurança digital é uma área de grande impacto social e crescente interesse, o que pode motivar os alunos a aprofundarem seus estudos matemáticos por meio de aplicações práticas.

No entanto, um dos principais desafios enfrentados na implementação da proposta é o **engessamento do currículo escolar**. Atualmente, as escolas públicas estaduais de Mato Grosso seguem um cronograma curricular rígido denominado *Sistema Estruturado de Ensino*, o que pode dificultar a aplicação completa da metodologia devido à carga horária restrita. Muitos professores enfrentam dificuldades para incluir novas abordagens devido à necessidade de cumprir o planejamento curricular obrigatório, limitando a exploração de atividades como as definidas aqui.

Diante desse cenário, algumas alternativas foram consideradas. Uma possibilidade seria a aplicação da proposta nas turmas contempladas com o itinerário formativo **Matemática e suas Tecnologias**, especificamente por meio da disciplina de **Aprofundamento em Matemática**. Essa abordagem permitiria ao professor utilizar três aulas semanais ao longo do ano para desenvolver as atividades propostas. No entanto, essa solução restringiria sua implementação apenas ao ensino médio.

Outra alternativa, que permitiria sua aplicação também no ensino fundamental, seria o desenvolvimento da proposta por meio de um programa de pesquisa e extensão. No contexto do estado de Mato Grosso, o **Programa Pesquisa e Inovação na Escola (PIE)**, promovido pela **Fundação de Amparo à Pesquisa do Estado de Mato Grosso (FAPEMAT)**, poderia viabilizar essa iniciativa. Esse programa tem como objetivo despertar nos professores e estudantes da rede estadual de ensino a vocação para a pesquisa, o desenvolvimento tecnológico e a inovação, sendo um meio eficaz para aplicação das atividades propostas.

Além de proporcionar uma nova abordagem para os alunos, este trabalho também tem como objetivo **fornecer aos professores uma proposta metodológica viável para ser implementada em sala de aula**. A estrutura apresentada busca auxiliar os docentes na modernização de suas práticas pedagógicas, oferecendo alternativas que integrem tecnologia e matemática de maneira acessível e eficaz. Com isso, espera-se que os professores tenham um ponto de partida para desenvolver atividades capazes de despertar maior interesse e engajamento nos estudantes.

Outro aspecto relevante a ser considerado é a ampliação do uso de ferramentas tecnológicas no ensino da matemática. Além das planilhas eletrônicas, destacam-se softwares específicos para ensino matemático e linguagens de programação como *Python*, que permitem uma abordagem mais aprofundada e interativa dos conceitos matemáticos. O uso dessas ferramentas pode favorecer o pensamento computacional e tornar o ensino mais alinhado às atuais demandas tecnológicas.

Por fim, ressalta-se que as propostas apresentadas devem ser objeto de futuras pesquisas e experimentações para que possam ser validadas na prática. A avaliação de sua

eficácia e o aprimoramento de suas aplicações no contexto educacional poderão contribuir significativamente para o ensino da matemática, tornando-o mais acessível, interativo e conectado às realidades contemporâneas.

Referências Bibliográficas

[1] Coutinho, S. C. *Números Inteiros e Criptografia RSA*. IMPA, Série de Computação e Matemática, 226 páginas, Rio de Janeiro, 2005.

[2] Hefez, A. *Iniciação à Aritmética*. Programa de Iniciação Científica da OBMEP, Volume 01. IMPA/OBMEP, 127 páginas, Rio de Janeiro, 2016.

[3] Coutinho, S. C. *Criptografia*. Programa de Iniciação Científica da OBMEP, Volume 07. IMPA/OBMEP, 217 páginas, Rio de Janeiro, 2012.

[4] Assis, C. e Miranda, T. *Módulo: Resolução de Exercícios Máximo Divisor Comum e Mínimo Múltiplo Comum, 6^o ano E.F.* Produzido pelo IMPA - Instituto de Matemática Pura e Aplicada. Retirado do portal da OBMEP. Disponível em: https://cdnportaldaobmep.impa.br/portaldaobmep/uploads/material/5h3eihokpiosg.pdf. Acessado em 02 de janeiro de 2025.

[5] Assis, C. e Miranda, T. *Módulo: Resolução de Exercícios Operações com Números Naturais, 6^o ano E.F.* Produzido pelo IMPA - Instituto de Matemática Pura e Aplicada. Retirado do portal da OBMEP. Disponível em: https://cdnportaldaobmep.impa.br/portaldaobmep/uploads/material/8m6trk3rs1s0g.pdf. Acessado em 31 de janeiro de 2025.

[6] Domingues, Hygino; Iezzi, Gelson. *Álgebra Moderna: Volume Único*. 4ª ed., Editora Atual, 368 páginas, São Paulo, 2003.

[7] Hefez, Abramo. *Aritmética*. 2ª ed., SBM, Coleção PROFMAT 08, 298 páginas, Rio de Janeiro, 2016.

[8] Alencar Filho, Edgard de. *Teoria Elementar dos Números*. 4ª ed., Editora Nobel, 383 páginas, São Paulo, 1913.

[9] Fisher, Steven R. *História da Leitura* (tradução: Cláudia Freire). Editora UNESP, 384 páginas, São Paulo, 2006.

[10] Wikipédia. Biografia de Jean-François Champollion. *Wikipédia, a enciclopédia livre*, acessado em 11 de novembro de 2024. Disponível em: https://pt.wikipedia.org/wiki/Jean-Fran%C3%A7ois_Champollion.

[11] Singh, Simon. *O Livro dos Códigos* (tradução: Jorge Calife). 14ª ed., Editora Record, 446 páginas, Rio de Janeiro, 2022.

[12] Ziviani, Nivio. *Projeto de Algoritmos com Implementações em Pascal e C*. 4ª ed., Pioneira, 267 páginas, São Paulo, 1999.

[13] Turing, Alan. *On Computable Numbers, With an Application to the Entscheidungsproblem*. Acessado em 12 de novembro de 2024. Disponível em: https://www.cs.virginia.edu/~robins/Turing_Paper_1936.pdf.

[14] Wikimedia. Retrato de Alan Mathison Turing. *Wikimedia Commons*, acessado em 12 de novembro de 2024. Disponível em: https://commons.wikimedia.org/wiki/File:Alan_Turing_(1951).jpg.

[15] Carneiro, Framilson José Ferreira. *Criptografia e Teoria dos Números*. Editora Ciência Moderna, 121 páginas, Rio de Janeiro, 2017.

[16] Roque, Tatiana. *História da Matemática: Uma visão crítica, desfazendo mitos e lendas*. Editora Zahar, 121 páginas, Rio de Janeiro, 2012.

[17] Kleinjung, Thorsten; Aoki, Kazumaro; Franke, Jens; Lenstra, Arjen; Thomé, Emmanuel; Bos, Joppe; Gaudry, Pierrick; Kruppa, Alexander; Montgomery, Peter; Osvik, Dag Arne; Riele, Herman; Timofeev, Andrey; Zimmermann, Paul. *Factorization of a 768-Bit RSA Modulus*. Publicado em agosto de 2010, páginas 333–350. Disponível em: https://www.researchgate.net/publication/221354652_Factorization_of_a_768-Bit_RSA_Modulus. Acessado em 9 de janeiro de 2025.

[18] Parente, U. Lima *Módulo: Resolução de Exercícios O Regras de Divisibilidade- Parte 2, 6º ano E.F.* Produzido pelo IMPA - Instituto de Matemática Pura e Aplicada. Retirado do portal da OBMEP. Disponível em: https://cdnportaldaobmep.impa.br/portaldaobmep/uploads/material_teorico/5nbirw980dssw.pdf. Acessado em 31 de janeiro de 2025.

[19] Sant'Ana Júnior, B. *Introdução à Matemática Aplicada à Criptografia*. Edição do Kindle, 87 páginas. Taubaté-SP, 2024.

[20] Rivest, R. L., Shamir, A., Adleman, L. *A method for obtaining digital signatures and public-key cryptosystems*. Association for Computing Machinery, New York, NY, USA, número 2, ISSN 0001-0782, publicado em fevereiro de 1978. Disponível em: https://doi.org/10.1145/359340.359342. Acessado em 9 de janeiro de 2025.

[21] IMPA. *Descoberto número primo com quase 25 milhões de dígitos*. Publicado no site do IMPA. Disponível em: https://impa.br/noticias/descoberto-numero-primo-com-quase-25-milhoes-de-digitos/. Acessado em 10 de janeiro de 2025.

[22] Brasil. *BNCC: Base Nacional Comum Curricular*. Ministério da Educação. Brasília, DF: MEC, 2017. Disponível em: http://basenacionalcomum.mec.gov.br/images/BNCC_EI_EF_110518_versaofinal_site.pdf. Acessado em 11 de janeiro de 2025.

[23] Brasil. *BNCC: Base Nacional Comum Curricular - Computação*. Ministério da Educação. Brasília, DF: MEC, 2022. Disponível em: http://basenacionalcomum.mec.gov.br/images//historico/anexo_parecer_cneceb_n_2_2022_bncc_computacao.pdf. Acessado em: 16 de janeiro de 2025.

[24] Brasil. *LDB: Lei de Diretrizes e Bases da Educação Nacional - Lei nº 9.394/1996*. 7. ed. Brasília, DF: Senado Federal, 2023. 64 p. Disponível em: https://www2.senado.leg.br/bdsf/bitstream/handle/id/642419/LDB_7ed.pdf. Acessado em 11 de janeiro de 2025.

[25] CASTELLAR, Sonia M. Vanzella (Org.). *Metodologias ativas: sala de aula invertida*. 1. ed. São Paulo: FTD, 2016.

Apêndice – Material Adicional

Solução dos Exercícios Propostos

Nesta primeira parte do **Apêndice**, apresentamos propostas de solução para os exercícios. Vale destacar que as soluções aqui expostas representam apenas uma dentre as diversas abordagens possíveis. O objetivo é ilustrar uma das formas de resolver os problemas.

Exercício 4.1

Solução.
Apresentamos uma possibilidade para a criação da planilha.

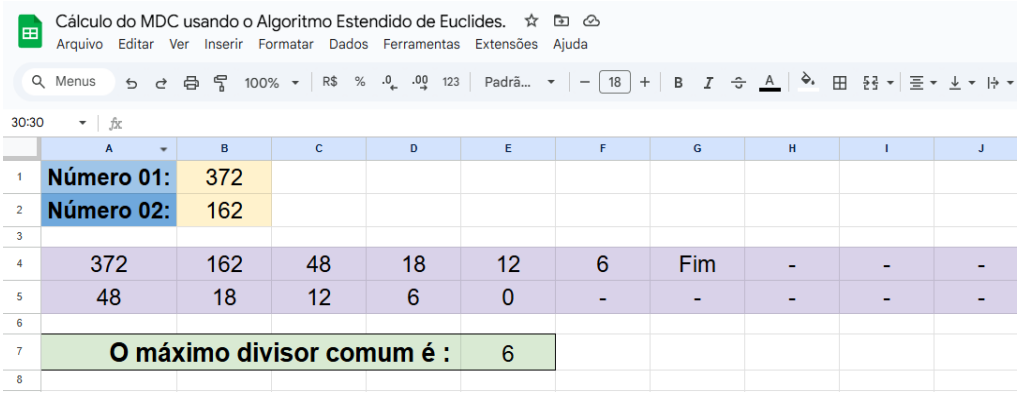

Figura 4.4: Planilha para o cálculo do MDC usando o Algoritmo de Euclides. Autoria própria. A planilha pode ser acessada clicando neste link.

A princípio, definiremos as células de entrada para os dois números dos quais se deseja obter o MDC. As células serão `B1` e `B2`. Devido à forma como a planilha foi configurada, há uma limitação no número de colunas, resultando em um máximo de 26 colunas. Assim,

o número máximo de divisões que poderemos realizar será 25, o que é suficiente para este propósito.

O próximo passo será indexar os valores de `B1` e `B2`, respectivamente, nas células `A4` e `B4`.

Feito isso, utilizaremos a função `=MOD(A4;B4)` na célula `A5`. Dessa forma, o resto da divisão de `A4` por `B4` será retornado em `A5`.

Agora, na célula `C4`, utilizaremos a função de teste lógico da seguinte maneira: `=IF(A5=0;"Fim";A5)`. Essa função verificará se `A5=0`. Caso verdadeiro, isso indica que os números são múltiplos, e o algoritmo será encerrado. Caso contrário, o valor de `A5` será repetido em `C4`, e o próximo passo será determinar o resto da divisão entre `B4` e `C4`.

É importante observar que é necessário verificar quando o resto da divisão for zero, para que o algoritmo possa ser encerrado. Para isso, utilizaremos novamente a função de teste lógico, definindo que, se o resto for zero, a célula correspondente seja preenchida com ‘‘-‘‘. Note, entretanto, que células adicionais poderão apresentar o valor ‘‘-‘‘. Para lidar com isso, utilizaremos a função lógica `OR`, que verificará se `B5=‘‘-‘‘` ou `B5=0`. Se uma dessas condições for verdadeira, a célula será preenchida com ‘‘-‘‘. Caso contrário, será realizada novamente a divisão, retornando o resto correspondente. A função completa fica assim expressa: `=IF(OR(B5=‘‘-‘‘);(B5=0));‘‘-‘‘;MOD(C4;D4))`.

Agora, basta clicar e arrastar até o final da coluna para que o procedimento seja repetido automaticamente. Com isso, a etapa de definição do algoritmo estará concluída.

Por fim, faremos com que o algoritmo identifique o máximo divisor comum (MDC) e o exiba na célula `E7`. Para realizar esse procedimento, seguiremos os passos a seguir. Na célula `E7`, será necessário identificar a posição na linha 4 onde aparece a palavra `Fim`. Para isso, utilizaremos a seguinte função, `=MATCH("Fim";4:4;0)`.

Essa função localizará a palavra `"Fim"` ao longo da linha 4, `4:4`, e retornará a posição da célula que contém essa palavra. Com isso, temos um referencial que nos permitirá identificar o MDC, que estará em uma célula anterior a essa. Para acessar esse valor, utilizaremos a função de indexação, `=INDEX(4:4;MATCH("Fim";4:4;0)-1)`.

Nessa função, subtraímos uma unidade do valor retornado pela função `MATCH`, exibindo o valor da célula imediatamente anterior à que contém a palavra `"Fim"`. Assim, conseguimos determinar e exibir o MDC, concluindo o desenvolvimento do nosso algoritmo. O resultado pode ser visualizado na Figura (4.4), que também apresenta um link para acesso descrito na legenda. □

Exercício 4.2

Solução.

Uma das formas de determinar o número mínimo de escolas é calculando o máximo de ingressos que cada escola poderá receber. Esse valor pode ser obtido pelo Máximo Divisor Comum (MDC) entre 400 e 320. Temos, então:

$$\begin{aligned} \text{mdc}(400, 320) &= \text{mdc}(400 - 320, 320) = \text{mdc}(80, 320) \\ &= \text{mdc}(80, 320 - 4 \cdot 80) = \text{mdc}(80, 0) = 80. \end{aligned}$$

Para determinar o número mínimo de escolas, basta calcular:

- Manhã: $400 \div 80 = 5$
- Vespertino: $320 \div 80 = 4$

Assim, o total de escolas será $5 + 4 = 9$. □

Exercício 4.3

Solução.

Apresentamos a seguir uma proposta para a criação da planilha, conforme ilustrado na imagem abaixo e descrito pelos procedimentos adotados. À primeira vista, a planilha pode parecer complexa; entretanto, com a explicação fornecida, verificaremos que os elementos nela presentes serão facilmente compreendidos.

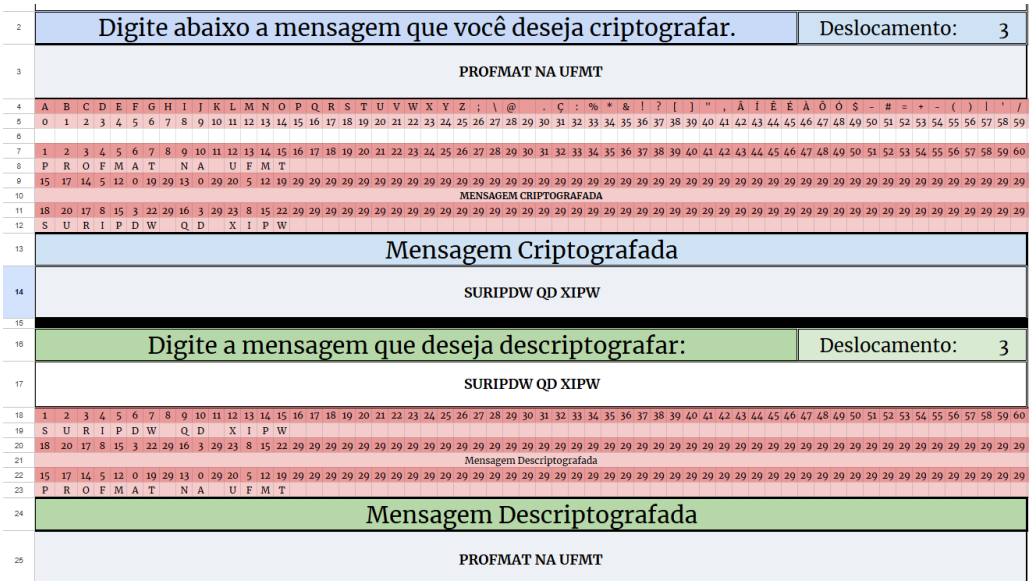

Figura 4.5: Estrutura Completa da Planilha: Cifra de César
Incluindo as linhas auxiliares. Fonte: Autoria própria. A planilha pode ser acessada clicando neste link.

Primeiramente, definimos o layout para atender ao enunciado do exercício. Após essa etapa, especificaremos duas células de entrada: uma delas será destinada ao deslocamento aplicado na cifra de César e a outra será para a inserção da mensagem. Para destacar esses parâmetros de entrada, aumentaremos o tamanho das células, mesclando aquelas que estão próximas. Assim, o valor de referência será o da primeira célula.

Por exemplo, na célula destinada ao valor do deslocamento, mesclaremos as células `BF3`, `BG3` e `BH3`, sendo que o valor de referência para a inserção de fórmulas será a célula `BF3`. De maneira similar, no outro campo de entrada, mesclaremos o intervalo `A12` até `BH12`.

Apos isso especificaremos algumas linhas para que possamos fazer as programações auxiliares, apos a conclusão da programação essas linha serão ocultadas, para que o programa fique mais apresentável.

Primeira Linha Auxiliar (Linha 4)

Definimos aqui os principais caracteres que serão extraídos e identificados na mensagem escrita. No exemplo, foram utilizados 60 caracteres. Essa quantidade pode ser ajustada conforme a planilha de cada aluno.

É importante que o caractere destinado a identificar o espaço entre as palavras seja especificado corretamente. Para isso, na célula `AD24`, utilizamos a seguinte configuração:

=“ “ (um espaço vazio). Essa especificação garante que os espaços na mensagem sejam reconhecidos e tratados adequadamente.

Lembre-se de que, se um caractere não estiver listado aqui, ele também não será identificado na mensagem.

Segunda Linha Auxiliar (Linha 5)

Cada célula dessa linha corresponderá ao valor numérico associado ao caractere indicado na linha anterior.

Terceira Linha Auxiliar (Linha 7)

Essa linha servirá como referencial para a posição dos caracteres extraídos da mensagem. Ela foi enumerada de 1 a 120, antecipando a extração dessa quantidade de caracteres.

Quarta Linha Auxiliar (Linha 8))

Cada célula dessa linha armazenará a extração de um caractere da mensagem, conforme a posição na da coluna da celula em que ela esta sento digitada. Por exemplo, considerando a mensagem `PROFMAT NA UFMT`, cada caractere, incluindo os espaços entre palavras, será extraído e realocado nas células correspondentes da linha 8.

Para automatizar esse processo, utilizamos a seguinte função: `=MID($A$3; COLUMN(); 1)`. Essa função extrai o caractere da célula `$A$3`, de acordo com a coluna (`COLUMN()`) na qual a função foi inserida. Por exemplo, na célula `A8`, que corresponde à primeira posição da linha, a fórmula retornará o primeiro caractere da célula `$A$3`, que, nesse caso, é a letra `P`. O número 1 no final da fórmula, `=MID($A$3; COLUMN(); 1)`, especifica que apenas um caractere será extraído para cada célula.

No entanto, se utilizarmos a função dessa forma, enfrentaremos um problema caso a mensagem digitada tenha menos de 120 caracteres. Nessa situação, a fórmula definida retornará um erro nas células que excedem o número de caracteres ocupados pela mensagem. Para solucionar esse problema, faremos uma verificação na posição de onde será extraído o caractere, verificando se ela não está vazia. Para isso, utilizaremos uma composição entre as funções `MID` e `IF`. Assim, a fórmula será ajustada da seguinte maneira:`=IF(MID($A$3; COLUMN(); 1)<>“ “; MID($A$3; COLUMN(); 1);“ “)`.

Essa função realiza um teste lógico verificando se a célula de onde se pretende extrair o valor contém um caractere, ou seja, se o valor extraído pela função `MID` é diferente (`<>`) de vazio (“ “). A lógica funciona da seguinte forma:
- Caso o teste seja verdadeiro, a função retorna o valor extraído pela fórmula `MID($A$3; COLUMN(); 1)`.
- Caso contrário (quando o teste for falso), a função retorna um espaço vazio (“ “).

Esse ajuste evita erros nas células ao lidar com mensagens que possuem menos caracteres do que o previsto, preenchendo as posições excedentes com espaços vazios, (“ “) de forma automática e garantindo a consistência do resultado.

Quinta Linha Auxiliar (Linha 9)

Essa linha receberá o valor numérico correspondente a cada letra extraída. Por exemplo, a letra `P` tem o valor 15, que será atribuído à célula correspondente.

Para realizar essa extração, utilizaremos a composição de duas funções. Inserindo a fórmula `=MATCH(A8; 4:4; 0)` na célula `A9`, estamos indicando que a função deve procurar o valor de `A8` ao longo da linha 4. O parâmetro `0` significa que a função retornará exatamente a posição do valor correspondente na linha. Assim, no exemplo, a fórmula retornará o valor `16`, que é a posição da célula correspondente à letra `P`.

Note que esse valor (`16`) não corresponde ao valor que definimos na linha auxiliar 5. Para solucionar esse problema, faremos com que o algoritmo retorne o valor associado à posição `16`, conforme definido na linha 5. Para isso, utilizaremos a fórmula: `=INDEX(5:5; MATCH(A8; 4:4; 0))` Essa função `INDEX` retornará o valor que está na linha 5 (`5:5`) na posição fornecida pela função `MATCH(A8; 4:4; 0)`.

Sexta Linha Auxiliar (Linha 11)

Nessa linha, os valores da linha auxiliar anterior serão utilizados para codificar a mensagem de acordo com o deslocamento informado na célula de entrada.

Na célula `A11`, a fórmula utilizada para realizar a codificação será:

```
=MOD(A9+$BF$2;60)
```

Essa fórmula realiza o seguinte cálculo:

i Soma o valor presente na célula `A9` ao valor digitado na célula de entrada `$BF$2`.

ii Calcula o resto da divisão dessa soma por 60, garantindo o comportamento cíclico necessário, já que temos um total de 60 caracteres definidos.

A divisão por 60 é essencial para evitar valores fora do intervalo dos caracteres definidos, criando um ciclo contínuo que "reinicia"após alcançar o limite de 60.

No entanto, ao utilizar a fórmula dessa forma, poderíamos ter um problema relacionado à repetição excessiva do caractere correspondente ao espaço, dependendo do deslocamento aplicado. Essa repetição tornaria a decodificação mais fácil, pois seria evidente qual símbolo representa o espaço entre as palavras.

Para resolver esse problema, empregaremos um teste lógico por meio da função `IF`, conforme mostrado abaixo:

```
=IF(A9=29; $AD$17; MOD(A9+$BF$2; 60))
```

Essa fórmula funciona da seguinte maneira:

i Verifica se o valor de `A9` é igual a `29`, que corresponde ao código do caractere espaço.

ii Se o teste for verdadeiro (`A9 = 29`), a função retorna diretamente o valor da célula `$AD$17`, mantendo o espaçamento original na mensagem codificada.

iii Caso contrário (teste falso), a função executa o cálculo `MOD(A9+$BF$2; 60)` normalmente, retornando o valor númerico codificado correspondente.

Dessa forma, o espaçamento entre as palavras é mantido na mensagem codificada, evitando padrões óbvios que poderiam facilitar a decodificação.

Sétima Linha Auxiliar (Linha 12))

Essa linha retornará o caractere correspondente ao valor numérico codificado na linha 11. Para isso, utilizaremos novamente a combinação de duas funções já utilizadas:

```
=INDEX(4:4; MATCH(A11; 5:5; 0))
```

A lógica de utilização é a mesma aplicada na quinta linha. Porém, neste caso, a função retornará uma letra correspondente ao valor codificado. Assim, obteremos a mensagem criptografada, onde cada uma das letras criptografadas estará em uma célula.

Apresentando a Mensagem Criptografada em uma Única Célula

Por fim, reuniremos as letras criptografadas presentes em cada célula da linha 12 em uma única célula, `A14`, que apresentará a mensagem criptografada. Para realizar este procedimento, na célula `A14`, utilizaremos a função `=TEXTJOIN(" "; TRUE; A12:BH12)`, que reúne as letras contidas nas células da linha 12 (`A12:BH12`), preservando os espaços.

Como definido anteriormente, para mensagens com menos de 120 caracteres, os caracteres faltantes serão representados por espaços. Para remover esses caracteres excedentes, faremos a composição da função `TEXTJOIN` com a função `TRIM`, que remove espaços à esquerda, à direita e duplicados no texto. Assim, a função combinada fica da seguinte maneira: `=TRIM(TEXTJOIN(" "; TRUE; A12:BH12))`.

Dessa forma finalizamos a programação que compõem a parte que encriptação da mensagem. Para descriptografar a mensagem a ideia será a mesma utilizada para encriptação a única mudança é que para reverter a encriptação utilizaremos o inverso aditivo da fórmula utilizada para realizar a codificação será:

```
=IF(A20=$AD$5; $AD$5; MOD(A20-$BF$16; 60))
```

para que o programa tenho um aspecto melhor ocultaremos as linhas auxiliares ficando apenas com as principais partes do programa conforme a seguinte figura.

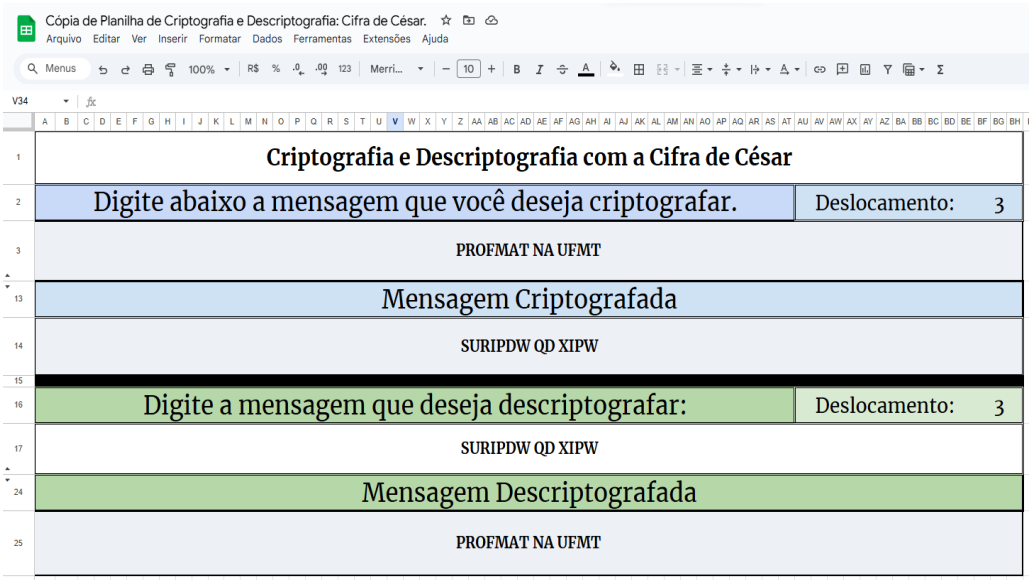

Figura 4.6: Configuração final da planilha: Cifra de César.
Fonte: Autoria própria. A planilha pode ser acessada clicando neste link.

□

Exercício 4.4

Solução.

Esse exercício pode ser considerado tanto uma atividade exploratória quanto uma avaliação diagnóstica, pois, ao apresentar uma pergunta subjetiva, permite avaliar a compreensão dos alunos sobre os processos de decodificação, o padrão de repetição das letras e a familiaridade com o método de análise de frequência. Por exemplo, os alunos podem perceber que letras como "A", "E"e "O"são mais frequentes na língua portuguesa e podem fornecer pistas sobre o ciclo de deslocamento utilizado na cifra.

Alguns alunos podem relatar dificuldade em identificar padrões, especialmente quando o texto é curto. Outro ponto importante que deve ser explorado é a fragilidade desse método de criptografia. Por meio de tentativas e erros, modificando o parâmetro de entrada, é possível verificar quantos deslocamentos podem ser realizados para descriptografar a mensagem.

Além disso, é interessante questionar os alunos com perguntas que os levem a refletir. Por exemplo: "Por que o número 63 na caixa de deslocamento produz a mesma mensagem criptografada que o número 3?"Isso os incentivará a pensar sobre a relação entre deslocamentos e aritmética modular, enriquecendo a análise crítica e o aprendizado. □

Exercício 4.5

Solução.

Como sugestão para a solução do exercício, novamente começaremos pensando no layout da planilha, formatando as principais células de entrada, bem como as áreas onde serão apresentadas a mensagem criptografada e a mensagem descriptografada.

A confecção desta planilha seguirá muitos dos passos utilizados na programação da Cifra de César. Para evitar explicações repetitivas e desnecessárias, faremos referência, sempre que possível, às ideias utilizadas no Exercício 4.2.

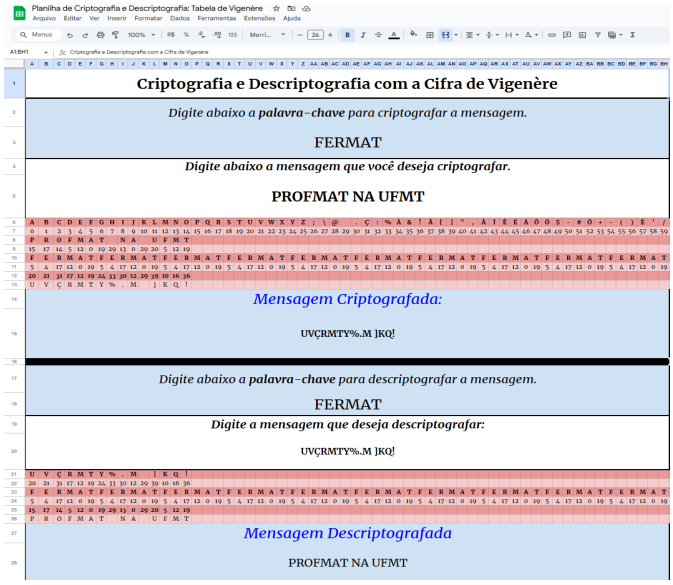

Figura 4.7: Esquema da planilha para cifragem com Vigenère.
A planilha pode ser acessada clicando neste link.

As linhas em vermelho são utilizadas para programações auxiliares e, após a conclusão do algoritmo, serão ocultadas para melhorar o aspecto visual da planilha. Conforme ilustrado na figura, utilizaremos as linhas 6 e 7 para indexar o alfabeto e os valores numéricos correspondentes.

Observe que as frases codificadas aqui serão diferentes das apresentadas no Exercício 1.1, pois, nesta configuração, utilizaremos uma tabela expandida com 60 caracteres, incluindo o espaço entre as palavras como um caractere válido.

Na linha 8 da planilha, extrairemos as letras da mensagem. Na célula `A8`, utilizaremos a fórmula `=MID($A$5; COLUMN(); 1)` combinada com o teste lógico `IF`, para verificar se o espaço correspondente não está vazio antes de extrair as letras. Isso evita possíveis erros de criptografia. Após a composição, a fórmula final será: `=IF(MID($A$5; COLUMN(); 1)<>" "; MID($A$5; COLUMN(); 1); " ")` Essa fórmula verifica se o caractere na posição especificada não é vazio. Caso seja um caractere válido, ele será extraído; caso contrário, a célula permanecerá em branco.

Na linha 9 da planilha, indexaremos a letra extraída ao seu número correspondente, conforme definido nas linhas 6 e 7. Para isso, utilizaremos a fórmula `=if(A8<> ;INDEX(7:7;MATCH(A8;6:6;0));)`, conforme explicado no exercício 1.1.

No entanto, antes de aplicá-la, é necessário verificar se a célula `A8` não está vazia. Caso esteja preenchida, a fórmula associará a letra extraída ao valor especificado na tabela. Assim, a fórmula completa será: `=IF(A8<>" ";INDEX(7:7;MATCH(A8;6:6;0));" ")`. Essa verificação garante que a planilha funcione corretamente, evitando erros causados

por células vazias.

Na linha 10 da planilha, será realizada a repetição da palavra "FERMAT" de forma sucessiva e contínua. Para isso, utilizaremos uma ideia explorada no exercício 4.6. Primeiramente, precisamos contar o número de caracteres da palavra-chave, o que pode ser feito facilmente com a função `LEN($A$3)`. Essa função retorna o número de caracteres da palavra definida na célula `$A$3`, que, no caso da figura 4.7, é "FERMAT"e possui seis letras.

Para associar cada letra da palavra digitada aos seus possíveis restos, utilizaremos a numeração das colunas em uma determinada linha, obtida pela função `=COLUMN()`. A partir disso, podemos combinar essas duas funções para produzir uma sequência oscilante com período igual ao número de caracteres da palavra-chave.

A sequência oscilante é gerada pela função:

```
=MOD(COLUMN(); LEN($A$3))
```

Essa fórmula retorna a seguinte sequência periódica: $1, 2, 3, 4, 5, 0, 1, 2, 3, 4, 5, 0, \ldots$.

No entanto, há um problema com o valor zero na função `=MID`, pois ela só aceita parâmetros maiores ou iguais a 1. Para contornar isso, utilizaremos um teste lógico para ajustar o valor 0 ao último caractere da palavra-chave.

A fórmula ajustada será:

```
=IF(MOD(COLUMN();LEN(A3))<>0;mid(A3;MOD(COLUMN();LEN(A3));1);mid(A3;LEN(A3);1))
```

Essa função verifica se o valor de `=MOD(COLUMN(), LEN($A$3))` é diferente de zero. Se for, utiliza o valor diretamente na função `MID`; caso contrário, utiliza o comprimento total da palavra-chave (`LEN($A$3)`) para retornar o último caractere. Dessa forma, a palavra será repetida continuamente, como foi digitada originalmente.

Na linha 11 da planilha, utilizaremos uma combinação já explorada no exercício anterior: `=INDEX(7:7; MATCH(A10; 6:6; 0))`. Essa fórmula indexará cada letra da linha 10 ao seu valor numérico correspondente na tabela previamente definida.

Na linha 12 da planilha, realizaremos a criptografia de cada letra da mensagem. Para exemplificar, utilizaremos a célula `A12`, combinando o parâmetro numérico da letra da mensagem a ser convertida com o valor correspondente da letra da palavra-chave. Antes de realizar o cálculo, é necessário verificar se as células contêm valores numéricos. Para isso, utilizaremos uma função de teste lógico que verifica se a célula `A8` não está vazia. Caso esteja preenchida, a conversão será realizada; caso contrário, o espaço vazio será mantido. A fórmula utilizada será:

```
=IF(A8<>" "; MOD(A9+A11; 60); " ")
```

Essa fórmula calcula o resto da divisão da soma dos dois valores numéricos. O resultado obtido será o valor correspondente à letra criptografada. Por exemplo, no caso da fórmula `=MOD(A9+A11; 60)`, o valor resultante é 20, que corresponde à letra `U` na tabela definida.

É importante discutir com os alunos o motivo de utilizar a função `MOD`, que retorna o resto da divisão, em vez de usar diretamente a soma dos valores. Essa discussão ajuda a compreender como a periodicidade é mantida, garantindo que o valor resultante esteja sempre dentro do intervalo válido de 60 caracteres, evitando erros ou valores fora da tabela.

Na linha 13, indexaremos o valor numérico criptografado com a respectiva letra predefinida na tabela. Para isso, utilizaremos a fórmula `=INDEX(6:6;MATCH(A12;7:7;0))`. É essencial usar um teste lógico, pois essa função só deve ser executada quando a célula na linha 12 não estiver vazia. Caso contrário, o valor da célula na linha 13 deve permanecer vazio. Assim, a fórmula ficará da seguinte maneira: `=IF(A12<>'' ''; INDEX(6:6; MATCH(A12;7:7;0));'' '')`.

Para finalizar a parte de criptografia da mensagem, basta inserir na célula `A15` a fórmula conhecida, que concatena em uma única célula as letras criptografadas ao longo da linha 13: `=TRIM(TEXTJOIN('' ''; TRUE; A13:DP13))`.

Essa fórmula organiza a mensagem criptografada, eliminando espaços desnecessários e garantindo um formato limpo e legível. Assim, concluímos a parte de criptografia da mensagem.

Para descriptografar a mensagem, o procedimento é muito semelhante ao utilizado na encriptação. Primeiramente, definiremos a célula de entrada onde a palavra-chave será digitada (`A18`) e a célula onde a mensagem criptografada será inserida (`A20`).

Na linha 21 da planilha, utilizaremos novamente a expressão: `=IF(MID($A$20, COLUMN(), 1)<>'' ''; MID($A$20, COLUMN(), 1); '' '')`. Essa fórmula extrairá as letras da mensagem digitada para as células da linha 21.

Na linha 22, indexaremos as letras extraídas aos valores numéricos da tabela predefinida. Caso a célula esteja vazia, ela permanecerá vazia. Assim, a fórmula, mais uma vez, utilizada será: `=IF(A21<>;INDEX(7:7;MATCH(A21;6:6;0));)`.

Na linha 23, utilizaremos a mesma ideia aplicada na linha 10, apenas ajustando o referencial das células de entrada. Combinaremos duas funções para gerar uma sequência oscilante com período igual ao número de caracteres da palavra-chave, utilizando a seguinte fórmula:

`=IF(MOD(COLUMN();LEN(`$A18$`))<>0;mid(`$A18$`;MOD(COLUMN();LEN(`$A18$`));1);mid(`$A18$`;LEN(`$A18$`);1))`

Na linha 24, utilizaremos uma combinação já explorada no exercício anterior: `=INDEX(7:7; MATCH(A23; 6:6; 0))`. Essa fórmula indexará cada letra da linha 10 ao

seu valor numérico correspondente na tabela previamente definida.

Na linha 25, utilizaremos o mesmo procedimento adotado no processo de criptografia, descrito na linha 10, com uma pequena diferença: agora será realizado o processo inverso, correspondente à determinação do resto da subtração dos valores, dividido pelo número de letras da palavra-chave. A fórmula ficará da seguinte forma:

```
=IF(A22<>; MOD(A22-A24;60); ” ”)
```

Na linha 26, indexaremos o valor numérico descriptografado com a respectiva letra predefinida na tabela. Para isso, basearemos a explicação no procedimento realizado na linha 13, ajustando para os novos valores de entrada. A fórmula utilizada será: `=IF(A25<>” ”; INDEX(6:6; MATCH(A25;7:7;0)); ” ”)`.

Para concluir o exercício, teremos a mensagem descriptografada ao longo da linha 26, com a primeira letra localizada na célula `A26`. Para reunir todas as letras em uma única célula (`A28`), utilizaremos a seguinte função ja explanada: `=TRIM(TEXTJOIN(” ”; TRUE; A26:BH26))`.

Por fim, ocultaremos as linhas auxiliares para que a planilha fique mais apresentável.

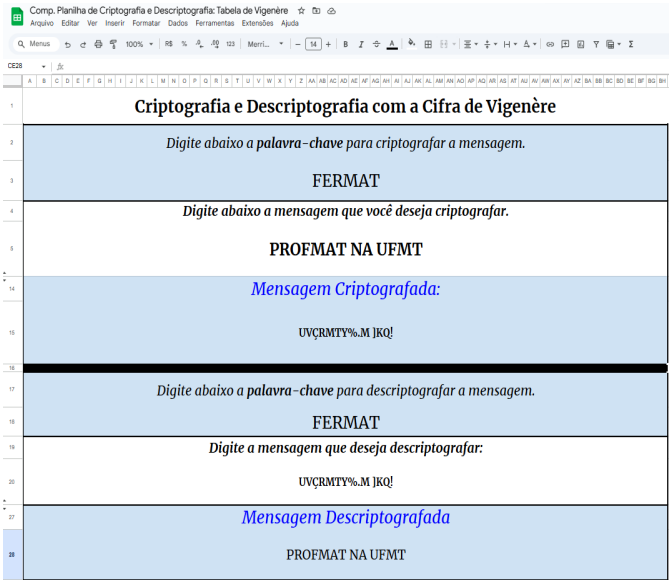

Figura 4.8: Planilha de Criptografia e Descriptografia usando a Tabela de Vigenère sem as linhas auxiliares. A planilha pode ser acessada clicando neste link.

□

Exercício 4.6

Solução.

Apresentamos uma proposta para a criação de uma planilha de fatoração por Fermat. Assim como nos demais exemplos, começaremos definindo a entrada do nosso algoritmo, que, neste caso, será apenas o número ímpar n. Na Figura 4.9 abaixo, o leitor pode acessar a planilha para compreender melhor a ideia.

Definiremos a célula `C1` como entrada. O próximo passo é configurar as células de saída, que serão: `E1`, a qual indicará se o número é composto, primo ou par; `B2`, representando o valor de X; `E2`, representando o valor de Y; `B4`, representando o valor do primeiro fator, $(X + Y) = f^1$; e `E4`, representando o valor do segundo fator, $(X - Y) = f^2$.

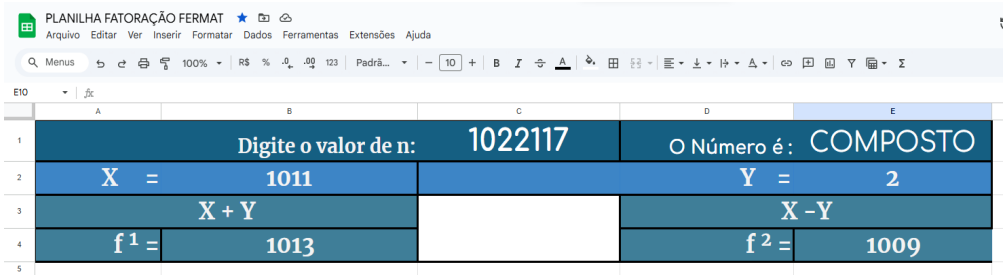

Figura 4.9: Fatoração por Fermat em planilha eletrônica
Autoria própria. A planilha pode ser acessada clicando neste link.

Nessa programação, diferente das demais propostas, utilizaremos duas abas, conforme pode ser verificado na Figura 4.10. A aba `PA` comportará o layout da planilha, contendo as entradas e saídas, enquanto a aba `CAL` será destinada à implementação dos cálculos. Essa alternativa se faz necessária devido à quantidade de linhas que utilizaremos para a execução dos cálculos e para as verificações lógicas.

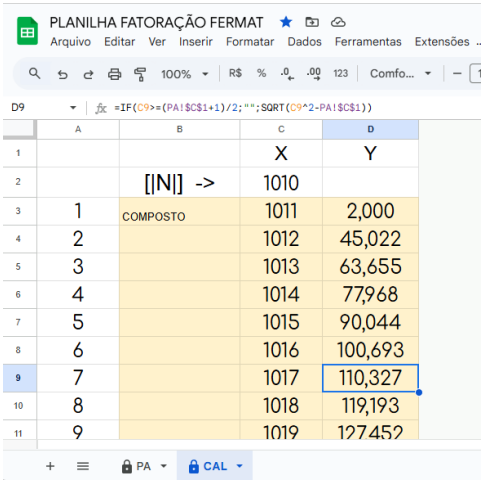

Figura 4.10: Guia de cálculos auxiliares para a planilha de fatoração por Fermat. Autoria própria. A planilha pode ser acessada clicando neste link.

Uma vez definido o layout da apresentação, começaremos determinando o número de células que utilizaremos para realizar os cálculos. Como o exercício solicita que seja possível calcular qualquer número de seis algarismos, e o maior número de seis algarismos é 999 999, utilizaremos o critério de parada para definir a quantidade máxima de células. Assim, temos: $X = \frac{999\,999+1}{2} = \mathbf{500\,000}$ linhas, garantindo que qualquer número semiprimo de seis algarismos seja determinado pelo algoritmo.

Feita a adição das linhas para completar a quantidade necessária, definiremos a primeira coluna, `A`, como parâmetro de enumeração. A coluna `B` será utilizada para verificar se o número é composto, primo ou PAR. Para isso, é necessária uma fórmula específica, a ser inserida na célula `B2`, que verificará se o número inserido é par, se nenhum número foi inserido, se o respectivo valor de y é um inteiro ou se o valor digitado é um número primo.

Note que aqui serão realizadas várias verificações na célula. Para essa função, utilizaremos a função de teste lógico `IF`, já empregada anteriormente. Para utilizá-la, faremos sua composição, aplicando-a quatro vezes. É importante mencionar que a ordem das verificações deve ser considerada para que a lógica não apresente falhas. Abaixo segue a fórmula responsável por essa verificação.

```
=IF(C2="PAR";"PAR";IF(PA!$C$1=“ ”;“ ”;IF(D2=INT(D2);"COMPOSTO";IF(C2>=(PA!$C$1+1)/2;"PRIMO";“ ”))))
```

Faremos uma breve explanação do seu funcionamento. Inicialmente, ele verificará se a célula `C2` contém a palavra `“PAR”`; se sim, repetirá essa palavra. Caso contrário, realizará um novo teste, verificando se a célula `C1` da aba `PA` está vazia; se estiver, repetirá o vazio. Se não, verificará se a célula `D2` contém um número inteiro, exibindo a palavra `COMPOSTO`.

Se não, fará um novo teste, verificando se a célula C2 é maior ou igual à divisão do número de entrada mais um por 2, exibindo a palavra PRIMO; caso contrário, exibirá vazio.

A célula C2 é a única da coluna C que terá esse comando, pois é necessário verificar, logo no início, se o número de entrada não é par, já que o algoritmo não funcionaria nesse caso. Para as demais células ao longo da coluna C, teremos:

=IF(D3=“ ”;“ ”;IF(D3=INT(D3);"COMPOSTO";IF(C3>=(PA!$C$1+1)/2;"PRIMO";“ ”)))

Vamos para a inserção das funções na coluna C. Novamente, utilizaremos alguns testes lógicos com a função IF. A expressão ficará da seguinte maneira:

=IF(MOD(PA!$C$1;2)<>0;INT(SQRT(PA!C1));IF(PA!$C$1=“ ”;“ ”;"PAR"))

Logo de início, precisamos verificar se o usuário não inseriu, de forma equivocada, um número par na entrada. Para tanto, utilizaremos a combinação da função MOD(PA!$C$1;2), que verifica se o número inserido na entrada é divisível por dois. Caso não seja, a função deve retornar apenas a parte inteira da raiz quadrada do número de entrada, ou seja, INT(SQRT(PA!C1)). Se o número de entrada for divisível por 2, deve-se verificar se a célula de entrada está vazia; se estiver, repetir o valor vazio, caso contrário, exibir a palavra “PAR”.

Novamente, essa expressão é inserida apenas na célula C2, devido à possibilidade de o número ser par. Para as demais células dessa coluna, utilizaremos a seguinte expressão:

=IF(C2=“PAR”;“ ”;IF(C2=“ ”;“ ”;IF(C2+1>=(PA!$C$1+1)/2;“ ”;C2+1)))

Começaremos verificando se a célula C2 contém o valor PAR; se sim, o algoritmo não deverá continuar e será apresentado o valor vazio “ ”. Caso contrário, realizaremos um novo teste, verificando se o valor inserido em C2, incrementado de uma unidade, é maior ou igual à divisão do valor de entrada (também incrementado de uma unidade) por dois. Se verdadeiro, o algoritmo deverá parar, apresentando um espaço vazio; caso contrário, deverá retornar o valor de C2 incrementado de uma unidade.

Cabe aqui um pequeno comentário de otimização: a única célula que poderá apresentar a palavra “PAR” ao longo da coluna C é a célula C2, devido à lógica apresentada. Sendo assim, não é necessário realizar essa verificação a partir da célula C3. A expressão pode ser utilizada para todas as células da seguinte maneira:

=IF(C3=“ ”;“ ”;IF(C3+1>=(PA!$C$1+1)/2;“ ”;C3+1))

O próximo passo é inserir a logica de programação ao longo da coluna D. Novamente Utilizaremos o teste lógico IF, além disso a celula D2 terá a fórmula com algumas diferenças quando comparada com as demais da coluna C. A fórmula que utilizaremos nela é a seguinte:

=IF(SQRT(PA!$C1$)=int(SQRT((PA!$C1$)));0;“ ”)

A princípio, faremos o seguinte teste: verificaremos se o número de entrada é um quadrado perfeito, isto é, se =SQRT(PA!$C1$) é igual a =INT(SQRT(PA!$C1$)). Caso seja, X é um fator primo de n e o algoritmo se encerra, retornando 0 para o valor de y, caso contraio retornara um espaço vazio “ ”.

Para as células da coluna, utilizaremos a seguinte expressão:

=IF(C3>=(PA!$C$1+1)/2;“”;SQRT(C3-PA!$C$1))

A ideia é verificar se o valor de x contido na célula C3 satisfaz o Teste 1 do critério de parada do algoritmo, apresentando “”, caso contrário, deve-se calcular o valor de $Y = \sqrt{X^2 - N}$, conforme realizado no exemplo 2.6. Essa expressão será replicada ao longo das demais linhas. Finalizamos nossa programação na aba CAL.

Agora, na aba PA, programaremos as respostas das células de saída. Começaremos pela célula E1, a qual deverá indicar se o número é PAR, PRIMO ou COMPOSTO. Para realizar esse trabalho, utilizaremos uma função semelhante à =IFERROR(...), composta com a função =VLOOKUP(chave_de_pesquisa; intervalo; indice; [classificado]). Vamos compreender a lógica dessas duas funções.

Começaremos pela função =VLOOKUP("COMPOSTO";CAL!B2:D500000;1;FALSE). Essa função realizará uma busca vertical da palavra “COMPOSTO” ao longo da tabela formada pelas células da aba CAL (CAL!B2:D500000), retornando exatamente o valor contido na célula que contém a palavra, ou seja, a própria palavra. Se a palavra não for encontrada, a função retornará um erro.

Quando ocorrer um erro na função anterior, trataremos esse erro utilizando uma função derivada do =IF(...), ou seja, a função IFERROR(valor;[valor_se_erro]). A ideia é muito semelhante à do =IF(...). Por exemplo, a função =IFERROR(VLOOKUP("COMPOSTO";A1:D18;1;FALSE);"PRIMO") executará o seguinte teste lógico: ela procurará verticalmente a palavra “COMPOSTO” ao longo da tabela, conforme mencionado anteriormente; caso a palavra não seja encontrada, em vez de retornar um erro, retornará a palavra PRIMO.

O que temos que fazer agora é combinar essa lógica para que, em vez de retornar a palavra "PRIMO", a função retorne o resultado de um novo teste =IFERROR(...) que busque a palavra "PAR" ao longo da tabela informada. Caso ocorra um novo erro, isto é, se a palavra não existir, a função deverá retornar o resultado de outro teste =IFERROR(...) que verifica se a célula de entrada está vazia, ou seja, se (PA!$C1$ = “ ”), e, em caso de erro, deverá retornar a palavra PRIMO. Apesar do tamanho da função, a ideia é bem simples: em vez de retornarmos uma palavra na ocorrência de um erro, utilizaremos

novamente a `=IFERROR(...)` para procurar a próxima, de modo a contemplar essa lógica. A fórmula ficará da seguinte maneira:

```
=IFERROR(VLOOKUP("COMPOSTO";CAL!B2:D500000;1;FALSE);
IFERROR(VLOOKUP("PAR";CAL!B2:D500000;1;FALSE);IF(PA!$C$1=“ ”;“ ”;"PRIMO")))
```

Note que o intervalo de busca definido nas funções é composto pelas colunas responsáveis pelo cálculo de determinação de parada do algoritmo, que vai da célula `B2` até a célula `D500000` (`CAL!B2:D500000`).

Agora, vamos apresentar os valores de saída X e Y. Para tanto, utilizaremos novamente a mesma composição de funções `=IFERROR(...)` e `=VLOOKUP(...)`. A única mudança significativa será o número do índice da função de busca, que será modificado para 2 para exibição de X e para 3 para exibição de Y. Assim, na célula `B2`, a função retornará o valor da primeira célula à direita da que contém a palavra procurada `"COMPOSTO"`, que é o valor de X:

```
=IFERROR(VLOOKUP("COMPOSTO";CAL!B2:D500000;2;FALSE);“ ”)
```

Para a apresentação do valor de saída de Y na célula `E2`, basta alterar o índice para 3, retornando o valor da segunda célula à direita da célula que contém a palavra procurada. Caso não seja encontrada, será exibido o valor vazio, representado por `=IFERROR(...;“ ”)`:

```
=IFERROR(VLOOKUP("COMPOSTO";CAL!B2:D500000;3;FALSE);“ ”)
```

Finalmente, vamos apresentar os valores dos dois fatores, f^1 e f^2, do número semiprimo n de entrada. Para tanto, usaremos a função de teste lógico `=IF(...)`. A princípio, testaremos a célula de saída `E1`. Se o resultado for a palavra “PRIMO”, sabemos que o número terá apenas dois fatores: o próprio valor de entrada e o número 1. Assim, para esse primeiro caso, a função retornará o valor de `C1`; caso contrário, fará outro teste, verificando se a diferença entre os valores $x - y > 0$. Se essa condição for satisfeita, a função retornará a soma $x+y$, correspondente ao primeiro fator; caso contrário, retornará o valor de `E2`. Esse ajuste garante que, se n for um quadrado perfeito, os fatores serão iguais.

```
=IF(E1="Primo";C1;IF((B2-E2)>0;B2+E2;E2))
```

Para a exibição do segundo fator, a fórmula será a mesma, com uma sutil modificação. Agora, se `E1` for “PRIMO”, a função retornará o valor 1, pois, nesse caso, o número é primo. Caso contrário, verificará se a diferença entre os valores $x - y > 0$ é positiva; se

for, retornará essa diferença; caso contrário, retornará o valor de `E2` (caso do quadrado perfeito).

```
=IF(E1="Primo";1;IF((B2-E2)>0;B2-E2;E2))
```

□

Assim, concluímos a programação desta última proposta, que pode ser desenvolvida com os alunos a fim de aumentar o interesse na matemática, verificando sua relação com as tecnologias de criptografia moderna.

Descrição das Principais Habilidades Contempladas na Base Nacional Comum Curricular (BNCC)

Habilidades referentes à competência Matemática e suas Tecnologias [22]

EF06MA02: Reconhecer o sistema de numeração decimal, como o que prevaleceu no mundo ocidental, e destacar semelhanças e diferenças com outros sistemas, de modo a sistematizar suas principais características (base, valor posicional e função do zero), utilizando, inclusive, a composição e decomposição de números naturais e números racionais em sua representação decimal.

EF06MA03: Resolver e elaborar problemas que envolvam cálculos (mentais ou escritos, exatos ou aproximados) com números naturais, por meio de estratégias variadas, com compreensão dos processos neles envolvidos com e sem uso de calculadora.

EF06MA04: Construir algoritmo em linguagem natural e representá-lo por fluxograma que indique a resolução de um problema simples (por exemplo, se um número natural qualquer é par).

EF06MA06: Resolver e elaborar problemas que envolvam as ideias de múltiplo e de divisor.

EF06MA07: Compreender, comparar e ordenar frações associadas às ideias de partes de inteiros e resultado de divisão, identificando frações equivalentes.

EF06MA23: Construir algoritmo para resolver situações passo a passo (como na construção de dobraduras ou na indicação de deslocamento de um objeto no plano segundo pontos de referência e distâncias fornecidas etc.).

EF06MA33: Planejar e coletar dados de pesquisa referente a práticas sociais escolhidas pelos alunos e fazer uso de planilhas eletrônicas para registro, representação e interpretação das informações, em tabelas, vários tipos de gráficos e texto.

EF07MA01: Resolver e elaborar problemas com números naturais, envolvendo as noções de divisor e de múltiplo, podendo incluir máximo divisor comum ou mínimo múltiplo comum, por meio de estratégias diversas, sem a aplicação de algoritmos.

EF07MA03: Comparar e ordenar números inteiros em diferentes contextos, incluindo o histórico, associá-los a pontos da reta numérica e utilizá-los em situações que envolvam adição e subtração.

EF07MA04: Resolver e elaborar problemas que envolvam operações com números inteiros.

EF07MA05: Resolver um mesmo problema utilizando diferentes algoritmos.

EF07MA13: Compreender a ideia de variável, representada por letra ou símbolo, para expressar relação entre duas grandezas, diferenciando-a da ideia de incógnita.
EF07MA18: Resolver e elaborar problemas que possam ser representados por equações polinomiais de 1^{o} grau, redutíveis à forma $ax + b = c$, fazendo uso das propriedades da igualdade.
EF08MA01: Efetuar cálculos com potências de expoentes inteiros e aplicar esse conhecimento na representação de números em notação científica.
EF08MA02: Resolver e elaborar problemas usando a relação entre potenciação e radiciação, para representar uma raiz como potência de expoente fracionário.
EF08MA10: Identificar a regularidade de uma sequência numérica ou figural não recursiva e construir um algoritmo por meio de um fluxograma que permita indicar os números ou as figuras seguintes.
EF08MA11: Identificar a regularidade de uma sequência numérica recursiva e construir um algoritmo por meio de um fluxograma que permita indicar os números seguintes.
EF09MA04: Resolver e elaborar problemas com números reais, inclusive em notação científica, envolvendo diferentes operações.
EF09MA08: Resolver e elaborar problemas que envolvam relações de proporcionalidade direta e inversa entre duas ou mais grandezas, inclusive escalas, divisão em partes proporcionais e taxa de variação, em contextos socioculturais, ambientais e de outras áreas.
EF09MA18: Reconhecer e empregar unidades usadas para expressar medidas muito grandes ou muito pequenas, tais como distância entre planetas e sistemas solares, tamanho de vírus ou de células, capacidade de armazenamento de computadores, entre outros
EF09MA21: Analisar e identificar, em gráficos divulgados pela mídia, os elementos que podem induzir, às vezes propositadamente, erros de leitura, como escalas inapropriadas, legendas não explicitadas corretamente, omissão de informações importantes (fontes e datas), entre outros.
EF09MA22: Escolher e construir o gráfico mais adequado (colunas, setores, linhas), com ou sem uso de planilhas eletrônicas, para apresentar um determinado conjunto de dados, destacando aspectos como as medidas de tendência central.
EF09MA23: Planejar e executar pesquisa amostral envolvendo tema da realidade social e comunicar os resultados por meio de relatório contendo avaliação de medidas de tendência central e da amplitude, tabelas e gráficos adequados, construídos com o apoio de planilhas eletrônicas.
EM13MAT101: Interpretar criticamente situações econômicas, sociais e fatos relativos às Ciências da Natureza que envolvam a variação de grandezas, pela análise dos gráficos

das funções representadas e das taxas de variação, com ou sem apoio de tecnologias digitais.

EM13MAT102: Analisar tabelas, gráficos e amostras de pesquisas estatísticas apresentadas em relatórios divulgados por diferentes meios de comunicação, identificando, quando for o caso, inadequações que possam induzir a erros de interpretação, como escalas e amostras não apropriadas.

EM13MAT103: Interpretar e compreender textos científicos ou divulgados pelas mídias, que empregam unidades de medida de diferentes grandezas e as conversões possíveis entre elas, adotadas ou não pelo Sistema Internacional (SI), como as de armazenamento e velocidade de transferência de dados, ligadas aos avanços tecnológicos.

EM13MAT104: Interpretar taxas e índices de natureza socioeconômica (índice de desenvolvimento humano, taxas de inflação, entre outros), investigando os processos de cálculo desses números, para analisar criticamente a realidade e produzir argumentos.

EM13MAT201: Propor ou participar de ações adequadas às demandas da região, preferencialmente para sua comunidade, envolvendo medições e cálculos de perímetro, de área, de volume, de capacidade ou de massa.

EM13MAT203: Aplicar conceitos matemáticos no planejamento, na execução e na análise de ações envolvendo a utilização de aplicativos e a criação de planilhas (para o controle de orçamento familiar, simuladores de cálculos de juros simples e compostos, entre outros), para tomar decisões

EM13MAT301: Resolver e elaborar problemas do cotidiano, da Matemática e de outras áreas do conhecimento, que envolvem equações lineares simultâneas, usando técnicas algébricas e gráficas, com ou sem apoio de tecnologias digitais.

EM13MAT302: Construir modelos empregando as funções polinomiais de 1^{o} ou 2^{o} graus, para resolver problemas em contextos diversos, com ou sem apoio de tecnologias digitais.

EM13MAT304: Resolver e elaborar problemas com funções exponenciais nos quais seja necessário compreender e interpretar a variação das grandezas envolvidas, em contextos como o da Matemática Financeira, entre outros.

EM13MAT304: Resolver e elaborar problemas com funções exponenciais nos quais seja necessário compreender e interpretar a variação das grandezas envolvidas, em contextos como o da Matemática Financeira, entre outros.

EM13MAT313: Utilizar, quando necessário, a notação científica para expressar uma medida, compreendendo as noções de algarismos significativos e algarismos duvidosos, e reconhecendo que toda medida é inevitavelmente acompanhada de erro.

EM13MAT314: Resolver e elaborar problemas que envolvem grandezas determinadas

pela razão ou pelo produto de outras (velocidade, densidade demográfica, energia elétrica etc.).

EM13MAT315: Investigar e registrar, por meio de um fluxograma, quando possível, um algoritmo que resolve um problema.

EM13MAT401: Converter representações algébricas de funções polinomiais de 1^{o} grau em representações geométricas no plano cartesiano, distinguindo os casos nos quais o comportamento é proporcional, recorrendo ou não a softwares ou aplicativos de álgebra e geometria dinâmica.

EM13MAT405: Utilizar conceitos iniciais de uma linguagem de programação na implementação de algoritmos escritos em linguagem corrente e/ou matemática.

EM13MAT406: Construir e interpretar tabelas e gráficos de frequências com base em dados obtidos em pesquisas por amostras estatísticas, incluindo ou não o uso de softwares que inter-relacionem estatística, geometria e álgebra.

EM13MAT501: Investigar relações entre números expressos em tabelas para representá-los no plano cartesiano, identificando padrões e criando conjecturas para generalizar e expressar algebricamente essa generalização, reconhecendo quando essa representação é de função polinomial de 1^{o} grau

EM13MAT507: Identificar e associar progressões aritméticas (PA) a funções afins de domínios discretos, para análise de propriedades, dedução de algumas fórmulas e resolução de problemas.

Habilidades referentes à Computação [23]

EF69CO01: Classificar informações, agrupando-as em coleções (conjuntos) e associando cada coleção a um 'tipo de dado'.

EF69CO02: Elaborar algoritmos que envolvam instruções sequenciais, de repetição e de seleção usando uma linguagem de programação.

EF69CO03: Descrever com precisão a solução de um problema, construindo o programa que implementa a solução descrita.

EF69CO04: Construir soluções de problemas usando a técnica de decomposição e automatizar tais soluções usando uma linguagem de programação.

EF69CO05: Identificar os recursos ou insumos necessários (entradas) para a resolução de problemas, bem como os resultados esperados (saídas), determinando os respectivos tipos de dados, e estabelecendo a definição de problema como uma relação entre entrada e saída.

EF69CO06: Comparar diferentes casos particulares (instâncias) de um mesmo problema, identificando as semelhanças e diferenças entre eles, e criar um algoritmo para resolver

todos, fazendo uso de variáveis (parâmetros) para permitir o tratamento de todos os casos de forma genérica.