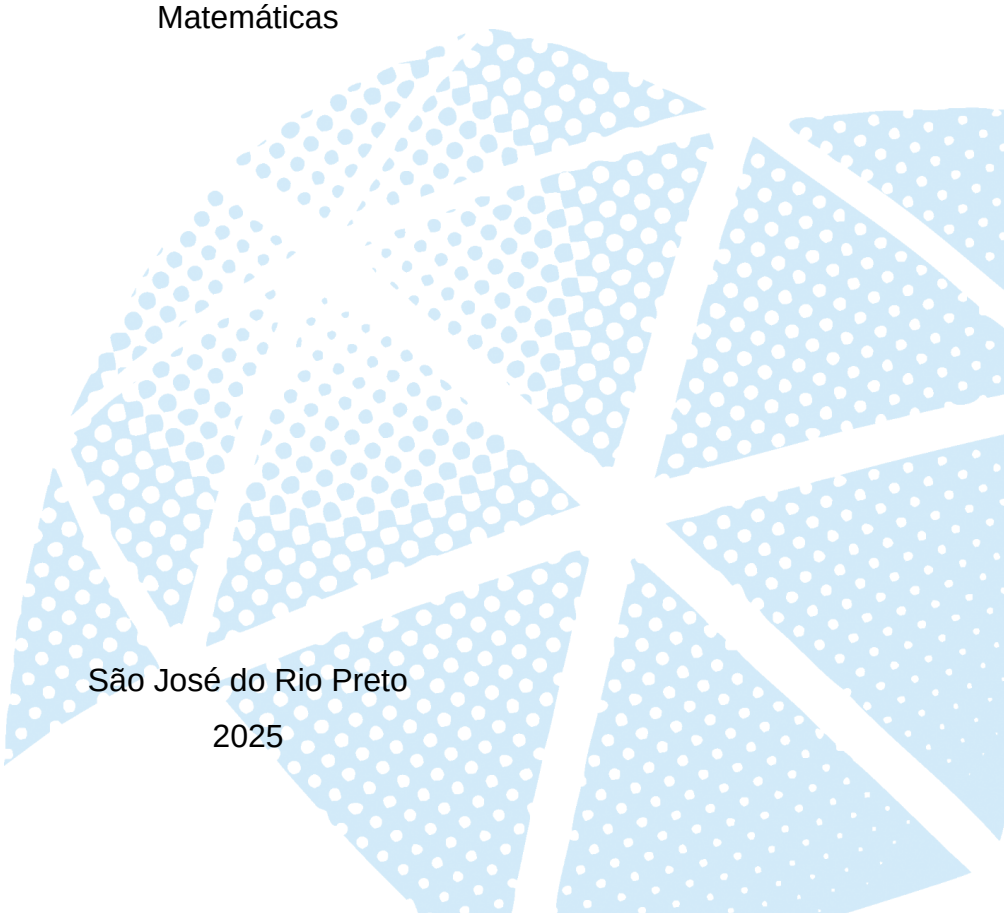
UNIVERSIDADE ESTADUAL PAULISTA – UNESP

Instituto de Biociências, Letras e Ciências Exatas − São José do Rio Preto

JOÃO OTAVIO FURTADO DA SILVA

CRIPTOGRAFIA NO ENSINO DE MATEMÁTICA:

Teoria, História e Atividades Didáticas para o Desenvolvimento de Competências Matemáticas



São José do Rio Preto

2025

JOÃO OTAVIO FURTADO DA SILVA

CRIPTOGRAFIA NO ENSINO DE MATEMÁTICA:

Teoria, História e Atividades Didáticas para o Desenvolvimento de Competências Matemáticas

Dissertação apresentada à Universidade Estadual Paulista (UNESP), Instituto de Biociências, Letras e Ciências Exatas, São José do Rio Preto, para obtenção do título de Grau acadêmico Mestre em Mestrado Profissional.

Área de Concentração: Programa de Pós−Graduação em Matemática em Rede Nacional

Orientador: Prof. Dr. Jéfferson Luiz Rocha Bastos

São José do Rio Preto

2025

S586c

Silva, João Otavio Furtado
CRIPTOGRAFIA NO ENSINO DE MATEMÁTICA : Teoria, História e Atividades Didáticas para o Desenvolvimento de Competências Matemáticas / João Otavio Furtado Silva. -- São José do Rio Preto, 2025
122 p.

Dissertação (mestrado profissional) - Universidade Estadual Paulista (UNESP), Instituto de Biociências Letras e Ciências Exatas, São José do Rio Preto
Orientador: Jéfferson Luiz Rocha Bastos

1. Criptografia. 2. Ensino de Matemática. 3. Cifra de César. I. Título.

JOÃO OTAVIO FURTADO DA SILVA

CRIPTOGRAFIA NO ENSINO DE MATEMÁTICA:

Teoria, História e Atividades Didáticas para o Desenvolvimento de Competências Matemáticas

Dissertação apresentada à Universidade Estadual Paulista (UNESP), Instituto de Biociências, Letras e Ciências Exatas, São José do Rio Preto, para obtenção do título de Grau acadêmico Mestre em Mestrado Profissional.
Área de Concentração: Programa de Pós−Graduação em Matemática em Rede Nacional

Data da defesa: 19 / 11 / 2025

Banca Examinadora:

_____

Prof. Dr. Jéfferson Luiz Rocha Bastos
UNESP − Instituto de Biociências, Letras e Ciências Exatas − São José do Rio Preto

_____

Prof. Dr. Flávia Souza Machado da Silva
UNESP − Instituto de Biociências, Letras e Ciências Exatas − São José do Rio Preto

_____

Prof. Dr. Oyran Silva Raizzaro
UEMS − Universidade Estadual de Mato Grosso do Sul – Nova Andradina

Dedico este trabalho à minha família, pelo amor incondicional e apoio constante. Aos meus pais, por acreditarem na educação como ferramenta de transformação. E a todos os professores que me inspiraram a ensinar com entusiasmo e compromisso.

AGRADECIMENTOS

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Código de Financiamento 001, por meio do Processo nº 88887.924020/2023−00.

Agradeço ao Programa de Mestrado Profissional em Matemática em Rede Nacional (PROFMAT) e à Universidade Estadual Paulista – UNESP / IBILCE, pela sólida formação acadêmica e pelo suporte oferecido ao longo desta trajetória.

À minha família, pelo apoio incondicional, incentivo e compreensão em cada etapa do percurso.

Ao meu orientador, Prof. Dr. Jéfferson Luiz Rocha Bastos, pela orientação cuidadosa, pelas reflexões compartilhadas e pela confiança no desenvolvimento deste trabalho.

Aos professores do PROFMAT, aos colegas de turma e aos amigos que contribuíram com diálogos, sugestões e companheirismo ao longo do mestrado, minha sincera gratidão.

Aos professores que gentilmente aceitaram compor a banca examinadora, Prof. Dr. Flávia Souza Machado da Silva (UNESP – Instituto de Biociências, Letras e Ciências Exatas – São José do Rio Preto) e Prof. Dr**.** Oyran Silva Raizzaro (UEMS – Universidade Estadual de Mato Grosso do Sul – Nova Andradina), agradeço pela disponibilidade, contribuições e pela atenção dedicada a esta pesquisa.

A todos que, de alguma forma, fizeram parte dessa jornada, o meu muito obrigado.

“A educação matemática precisa formar cidadãos que pensem, que decidam e que tenham autonomia”
(Ubiratan D’Ambrosio).

RESUMO

Esta dissertação investiga o uso da criptografia como recurso didático no ensino de Matemática, mostrando como seus fundamentos podem despertar a curiosidade e o interesse dos estudantes. Baseada em conceitos da Teoria dos Números e da Aritmética Modular, a criptografia favorece o desenvolvimento do raciocínio lógico, do pensamento crítico e de habilidades digitais. O objetivo foi evidenciar a Matemática como base dos algoritmos criptográficos e mostrar como atividades com códigos podem valorizar a disciplina no contexto escolar. A pesquisa, de caráter qualitativo, envolveu revisão bibliográfica, análise histórica da evolução da criptografia e elaboração de propostas didáticas. Do ponto de vista matemático, foram apresentados conteúdos como divisibilidade, algoritmo de Euclides, máximo divisor comum, números primos e resultados clássicos, como o Teorema Fundamental da Aritmética, o Pequeno Teorema de Fermat e o Teorema de Euler, com exemplos e aplicações para dar sustentação teórica. As atividades práticas incluíram cifras clássicas, César, Vigenère e Afim, com o uso do criptodisco e de ferramentas digitais em C++ no Microsoft Visual Studio. Elas foram aplicadas em duas escolas de Votuporanga/SP, com turmas do 9º ano do Ensino Fundamental e do 1º ano do Ensino Médio. Os resultados indicam que a integração da criptografia ao ensino favorece cálculos, aplicação de propriedades matemáticas, raciocínio lógico e trabalho em grupo, tornando a aprendizagem mais significativa. Concluiu−se que a criptografia pode ser uma estratégia pedagógica acessível e inovadora, capaz de enriquecer o ensino de Matemática e aproximar os estudantes dos desafios da sociedade digital.

Palavras−chave: criptografia; ensino de matemática; aritmética modular; propostas didáticas.

ABSTRACT

This dissertation investigates the use of cryptography as a didactic resource for teaching Mathematics, showing how its foundations can stimulate students' curiosity and interest. Supported by concepts from Number Theory and Modular Arithmetic, cryptography helps develop logical reasoning, critical thinking, and digital skills. The objective was to show Mathematics as the basis of cryptographic algorithms and demonstrate how activities involving codes can enhance the subject in the school context. This qualitative research included a literature review, a historical analysis of the evolution of cryptography, and the design of practical didactic proposals. From a mathematical perspective, the study presented topics such as divisibility, Euclid's algorithm, greatest common divisor, prime numbers, and classical results such as the Fundamental Theorem of Arithmetic, Fermat's Little Theorem, and Euler's Theorem, with examples and applications to ensure theoretical consistency. The practical activities involved classical ciphers, Caesar, Vigenère, and Affine, using the cryptodisc and digital tools developed in C++ with Microsoft Visual Studio. These activities were applied in two schools in Votuporanga/SP, with 9th-grade students from Elementary School and 1st-year High School students. The results indicate that integrating cryptography into teaching supports calculations, the use of mathematical properties, logical reasoning, and collaborative work, making learning more meaningful. The study concludes that cryptography can be an accessible and innovative pedagogical strategy that enriches Mathematics teaching and helps students connect with the challenges of the digital society.

Keywords: cryptography; mathematics education; modular arithmetic; didactic proposals.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

BNCC	Base Nacional Comum Curricular
C++	Linguagem de Programação C++
IA	Inteligência Artificial
IBGE	Instituto Brasileiro de Geografia e Estatística
LDB	Lei de Diretrizes e Bases da Educação Nacional
MDC	Máximo Divisor Comum
PNG	Portable Network Graphics (formato de imagem)
RSA	Rivest–Shamir–Adleman (algoritmo de criptografia assimétrica)
T.I.	Tecnologia da Informação
UNESP	Universidade Estadual Paulista
VS ou Visual Studio	Ambiente de desenvolvimento Visual Studio

LISTA DE SÍMBOLOS

+	Adição
−	Subtração (sinal negativo ou operação)
·	Multiplicação simbólica ou escalar
×	Multiplicação cruzada
/ ou ÷	Divisão
%	Porcentagem ou Resto da divisão usado em expressões em C++
mod	Operação módulo
≡	Congruência modular
≢	Não congruente
=	Igualdade
≠	Desigualdade
< / >	Menor que / Maior que
≤ / ≥	Menor ou igual / Maior ou igual
⇒	Implicação lógica
→	Mapeamento ou função
∈	Pertence ao conjunto
∣	Divide
∤	Não divide
( )	Parênteses
{ }	Chaves
[ ]	Colchetes
:	Separador ou definição
;	Final de instrução (programação)
∣ ∣	Valor absoluto
∣	Separador lógico
#	Marcador ou comentário (programação)
&	Conector lógico (e comercial)

\	Caractere de escape (programação)
φ	Função totiente de Euler
⋯	Reticência matemática
“ ” / ‘ ’ / ’	Aspas tipográficas
$\bar{\ }$	Traço superior (média, conjugado, etc.)

SUMÁRIO

INTRODUÇÃO

Vivemos em uma era fortemente marcada pela digitalização das relações sociais, econômicas e educacionais. Nesse cenário, a segurança da informação emerge como tema central, abrangendo desde a proteção de dados pessoais até a integridade de transações e comunicações. A criptografia, tradicionalmente associada à proteção de mensagens sigilosas, assume um papel estratégico, sendo amplamente utilizada em sistemas de autenticação, criptomoedas, comunicação segura e armazenamento de dados (Stallings, 2006).

Ao mesmo tempo, a escola, como espaço de formação integral, precisa dialogar com o universo tecnológico e integrá-lo de forma crítica ao currículo. Integrar a criptografia ao ensino da Matemática possibilita não apenas o desenvolvimento de competências lógico-matemáticas, mas também o fortalecimento de habilidades digitais, de resolução de problemas e de pensamento computacional. No contexto da Educação Matemática, essa abordagem potencializa práticas pedagógicas interdisciplinares, alinhadas às competências previstas pela Base Nacional Comum Curricular (Brasil, 2018).

Diante desse panorama, surge a seguinte questão norteadora: como a aplicação de algoritmos criptográficos pode contribuir para o desenvolvimento do pensamento lógico e despertar o interesse dos alunos nas aulas de Matemática do Ensino Fundamental e Médio?

Este trabalho se justifica pela percepção, enquanto professor de Matemática, de uma lacuna significativa entre os conteúdos matemáticos ensinados nas escolas e suas aplicações em contextos contemporâneos. A criptografia, nesse sentido, oferece um campo fértil para a exploração de conceitos como congruência, aritmética modular, funções, algoritmos e teoria dos números, de forma contextualizada e motivadora (Stallings, 2006).

A pesquisa tem como objetivo geral investigar o potencial didático da criptografia como recurso para o ensino da Matemática nos anos finais do Ensino Fundamental e no Ensino Médio. Busca−se, mais especificamente:

(i) apresentar os fundamentos históricos e matemáticos da criptografia;

(ii) desenvolver e aplicar atividades didáticas com cifras clássicas, como as cifras de César, de Vigenère e Afim;

(iii) implementar essas atividades com auxílio de ferramentas tecnológicas, como o Microsoft Visual Studio;
(iv) analisar, a partir da prática, o impacto dessas propostas no engajamento e na aprendizagem dos estudantes.

A investigação foi estruturada como uma pesquisa qualitativa, de cunho exploratório e aplicado, composta por revisão bibliográfica dos principais autores da área e pela elaboração de sequências didáticas. As atividades práticas envolveram codificação e decodificação de mensagens com cifras clássicas, aliando teoria dos números à programação computacional.

A contextualização histórica das técnicas criptográficas também desempenha um papel formativo ao destacar como diferentes civilizações buscaram proteger informações ao longo dos séculos. Ao compreender esse percurso, da scytale espartana à cifra de César, da cifra de Vigenère ao uso da máquina Enigma na Segunda Guerra Mundial, o estudante é convidado a perceber a Matemática como construção histórica e cultural, dinâmica e funcional. Conforme destaca Hefez (2022), essa abordagem favorece o engajamento dos alunos e amplia o significado atribuído aos conceitos matemáticos trabalhados.

Esta dissertação está organizada em seis capítulos, além da introdução e da conclusão. O Capítulo 1 apresenta uma abordagem histórica da criptografia, desde suas origens na Antiguidade até os avanços modernos, incluindo cifras clássicas, análise de frequência e uma discussão sobre a criptografia como recurso pedagógico. O Capítulo 2 trata dos conceitos fundamentais da Aritmética, como divisibilidade, algoritmo de Euclides, números primos e o Teorema Fundamental da Aritmética, estabelecendo a base teórica para os capítulos seguintes. Já o Capítulo 3 discute os fundamentos matemáticos aplicados à criptografia, com destaque para a aritmética modular, os teoremas de Euler e de Fermat e a construção algorítmica das funções criptográficas. O Capítulo 4 apresenta as aplicações tecnológicas da criptografia, abordando algoritmos modernos, como RSA, e a implementação de códigos em C++ utilizando o ambiente Visual Studio.

O Capítulo 5 oferece um guia prático de atividades didáticas voltadas à educação básica, com propostas detalhadas para aplicação em sala de aula, unindo matemática, tecnologia e segurança da informação. Por fim, o Capítulo 6 descreve a aplicação prática dessas atividades em duas instituições de ensino da cidade de

Votuporanga/SP, identificadas como Escola X e Escola Y. Essa etapa buscou observar a receptividade dos alunos, os desafios enfrentados em sala de aula e as possibilidades de integração entre Matemática, tecnologia e temas contemporâneos, como a segurança digital, consolidando a proposta como um recurso inovador para o ensino da disciplina.

CAPÍTULO 1 – HISTÓRICO E EVOLUÇÃO DA CRIPTOGRAFIA

1.1 Conceitos e Origem Histórica

A criptografia, entendida como a ciência e a arte de ocultar mensagens, atravessa milênios de história da humanidade. A própria etimologia do termo, do grego *kryptós* (oculto) e *gráphein* (escrita), já revela sua essência. Antes mesmo do surgimento da escrita formal, o ser humano desenvolveu maneiras engenhosas de resguardar informações importantes.

> Essa arte de esconder mensagens, a criptografia, é mais antiga do que a própria escrita, sendo encontrada no sistema de escrita egípcio, Hieroglífica, que tinha um propósito de esconder o real significado do texto. (Carneiro, 2017, p. 22)

Desde os tempos mais remotos, reis, imperadores, generais e líderes políticos buscaram formas seguras de comunicação com seus aliados, longe da vigilância de inimigos e espiões. Essa necessidade deu origem a diversos métodos de ocultação da informação, entre eles, técnicas engenhosas de disfarce da própria existência da mensagem. Surge, nesse contexto, a esteganografia, prática que se distingue da criptografia justamente por esconder o fato de que há uma mensagem a ser decifrada.

Entre os métodos mais inusitados, destaca−se o uso de mensagens escritas no couro cabeludo de mensageiros. Heródoto[1], em sua obra História, descreve uma estratégia utilizada por Histieu[2], tirano de Mileto, para incitar Aristágoras[3], que liderou a Revolta Jônica contra o Império Persa no século V a.C. Desconfiando dos meios tradicionais, Histieu raspou a cabeça de um escravo, tatuou sobre a pele a mensagem desejada e aguardou o crescimento dos cabelos. O escravo foi então enviado como portador da mensagem viva, ao chegar ao seu destino, Aristágoras raspou novamente a cabeça do homem e leu as instruções codificadas. (Heródoto, História, Livro V)

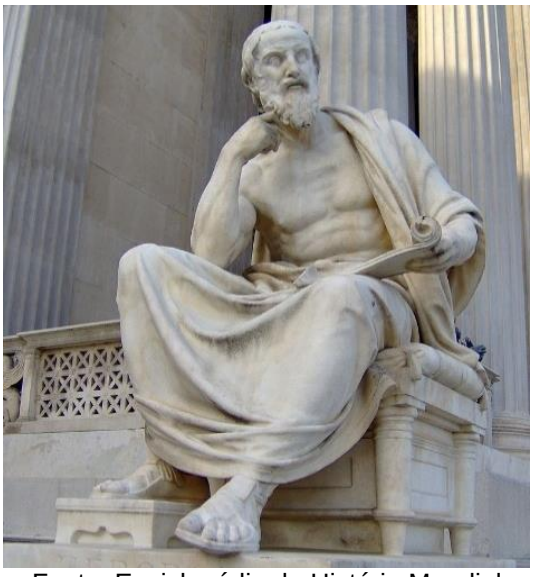
---

1 Heródoto (c. 484 a.C. – c. 425 a.C.)**:** Historiador grego conhecido como o "Pai da História". Escreveu sobre as Guerras Greco-Persas e a Revolta Jônica.

2 Histieu (m. c. 493 a.C.): Governante de Mileto e aliado inicial dos persas. Mais tarde, teria incentivado a Revolta Jônica contra o domínio persa.

3 Aristágoras (m. c. 497 a.C.): Genro de Histieu e líder direto da Revolta Jônica. Iniciou o conflito ao se rebelar contra o rei Dario I da Pérsia.

Figura 1. Escultura de Heródoto, o “pai da história”[4]



Fonte: Enciclopédia da História Mundial

Além das cifras tradicionais, outras técnicas engenhosas foram empregadas ao longo da antiguidade para assegurar a confidencialidade das mensagens. Dentre elas, destacam−se o uso de tintas invisíveis, como o suco de limão, cuja escrita permanecia oculta até ser revelada pelo calor de uma chama, e o emprego de tabuletas de madeira revestidas com uma fina camada de cera, capazes de ocultar o verdadeiro conteúdo da mensagem. Um dos exemplos mais emblemáticos dessas práticas é narrado por Heródoto, que descreve como uma mensagem estratégica crucial sobre os planos do rei persa Xerxes foi secretamente transmitida aos gregos:

> 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 embaixo o que Xerxes pretendia fazer, depois a mensagem foi coberta novamente com cera. Deste modo, as tabuletas pareceriam estar em branco e não causariam problemas com os guardas ao longo da estrada. Quando a mensagem chegou ao seu destino, ninguém foi capaz de perceber o segredo, até que, pelo que entendi, a filha de Cleômenes, Gorgo, que era casada com Leônidas, adivinhou e contou aos outros que se eles raspassem a cera encontrariam alguma coisa escrita na madeira. Isto foi feito, revelando a mensagem, então transmitida para os gregos. (Singh, 2007, p. 20)

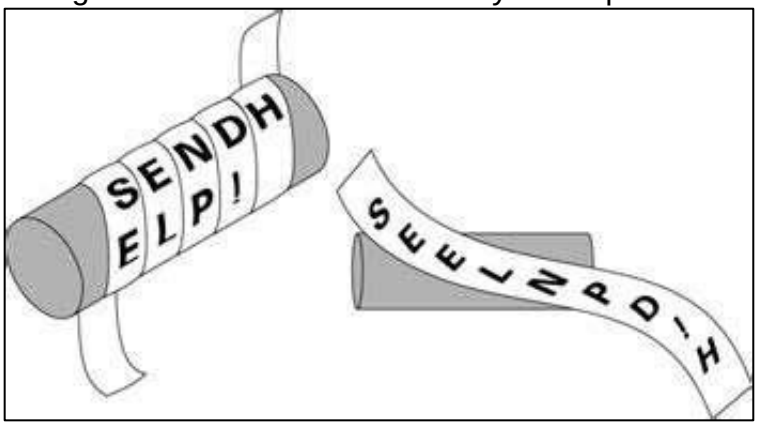
[4] Escultura localizada no Parlamento Austríaco, em Viena. Representa Heródoto, autor da obra "História", uma das principais fontes sobre o uso de mensagens ocultas na Antiguidade.

Essas estratégias evidenciam a sofisticação com que os antigos já lidavam com a necessidade de sigilo.

1.2 Cifras Clássicas: César e Espartana

A utilização de técnicas criptográficas em cenários militares da Antiguidade revela o valor estratégico atribuído à esteganografia como recurso de proteção, sobrevivência e domínio. Outro exemplo notável da Antiguidade é a scytale espartana, um bastão cilíndrico ao redor do qual se enrolava uma tira de couro. A mensagem era escrita de forma contínua ao longo da fita, uma vez desenrolada, os caracteres pareciam desconexos. Somente ao ser enrolada novamente em um bastão de mesmo diâmetro, a chamada “chave”, a mensagem podia ser lida, configurando um dos primeiros mecanismos de criptografia por transposição.

Figura 2. Funcionamento da Scytale Espartana



Fonte: Australian Science.

No campo das cifras de substituição, destaca−se a Cifra de César, atribuída ao general romano Júlio César[5], que, no século I a.C., utilizava essa técnica para proteger comunicações militares sigilosas. Trata−se de uma cifra de substituição monoalfabética, na qual cada letra do texto claro é substituída por outra letra do alfabeto, deslocada um número fixo de posições. O deslocamento tradicional utilizado pelo próprio César, é de três letras à frente, gerando um sistema simples, porém eficiente, para sua época.

[5] Júlio César (100 a.C. – 44 a.C.) foi um destacado general e político romano, cuja liderança militar e influência nas reformas de Roma marcaram profundamente a história da República. Sua trajetória é símbolo de poder estratégico e expansão territorial no mundo antigo.

Figura 3. Busto de Júlio César

Fonte: National Geographic Portugal.

O mais simples desses códigos consiste em substituir uma letra pela seguinte, isto é, transladar o alfabeto uma casa para a frente. Esse princípio básico permite a criação de atividades práticas em sala de aula com excelente aceitação entre os alunos, ao mesmo tempo em que introduz noções fundamentais de padrões, modularidade e simetria. (Coutinho, 2009)

A cifra de César opera por chave simétrica, ou seja, o mesmo valor de deslocamento, denominado chave, é usado tanto para cifrar quanto para decifrar o texto. Por exemplo, se a letra "A" é deslocada três posições para se tornar "D", o processo inverso, decifrar, desloca "D" três posições para trás, retornando à letra original.

> Um sistema simples de substituição monoalfabética, onde cada letra é substituída por outra letra de acordo com a chave utilizada, podendo ser ela qualquer uma das 26 letras do alfabeto, criando assim um texto criptografado, mas mantendo a ordem dos símbolos do texto original. (Tanenbaum e Wetherall, 2011, p. 483)

Ao associar cada letra a um número de 0 a 25 *(*$A=0, B=1, C=2, \ldots, Z=25$*)*, é possível representar o processo de cifragem pela função:

$$c(x) \equiv x + k \pmod{26}$$

Nesse contexto, considera−se:

- *x* como a posição original da letra no alfabeto;

- *k* como a chave responsável pelo deslocamento aplicado na codificação;
- *c(x)* é o valor resultante após a codificação, ou seja, a nova posição da letra cifrada.

A operação de módulo 26 garante que o resultado permaneça dentro dos limites do alfabeto, produzindo assim a letra codificada correspondente. Essa formalização oferece um contexto significativo para introduzir e praticar conceitos como congruência, funções afins e padrões numéricos, promovendo o desenvolvimento do raciocínio lógico e da linguagem algébrica dos estudantes.

- Exemplo 01:

Mensagem original: $AMO\ A\ OBMEP$

Chave utilizada: $k = 1$

Mensagem criptografada: $BNP\ B\ PCNFQ$

- Exemplo 02:

Mensagem original: $A\ LIGEIRA\ RAPOSA\ MARROM\ SALTOU\ SOBRE\ O\ CACHORRO\ CANSADO$

Chave utilizada: $k = 3$

Mensagem criptografada: $D\ OLJHLUD\ UDSRVD\ PDUURP\ VDOWRX\ VREUH\ R\ FDFKRUUR\ FDQVDGR$

De forma análoga à Cifra de César, a Cifra Aditiva Clássica utiliza o mesmo princípio de deslocamento, porém com um alfabeto expandido, abrangendo letras, números e alguns sinais de pontuação, como por exemplo:

ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789.,:!?

A formulação matemática permanece a mesma, sendo expressa por

$$c(x) \equiv x + k \pmod{n}$$

Nesse contexto, considera−se n como o tamanho total do alfabeto utilizado.

Embora essa cifra apresente semelhanças conceituais com a de César, sua implementação prática difere, especialmente quando aplicada em ambientes de programação. No Capítulo 5, ao propor sugestões pedagógicas com o uso do Visual Studio, será possível observar que o código da função responsável pela cifragem

exige cuidados específicos na manipulação do alfabeto ampliado e na conversão dos caracteres.

A cifra introduz, de forma intuitiva, conceitos que serão aprofundados posteriormente na teoria dos números, em especial na aritmética modular. Essa simetria possibilita atividades didáticas com codificações e decodificações entre os próprios alunos, promovendo o raciocínio lógico, o trabalho colaborativo e o domínio de operações modulares. Contudo, a simplicidade dessa cifra também é sua fragilidade. Por usar um alfabeto fixo e previsível, ela é extremamente vulnerável a ataques por análise de frequência, técnica que analisa a ocorrência de letras mais comuns em um idioma para quebrar o código, como demonstrado por estudiosos da criptografia ao longo da história.

1.3 A Influência Árabe e a Análise de Frequência

Durante a Idade Média, a criptografia passou a ser fortemente influenciada pelo mundo árabe. A civilização islâmica avançou significativamente na matemática, na linguística e na lógica, e isso repercutiu diretamente na criptografia. O estudioso Al−Kindi[6] foi o primeiro a registrar, por escrito, o método da análise de frequência, uma poderosa técnica de criptoanálise. A criptografia e a criptoanálise são duas áreas complementares dentro do campo da segurança da informação, com finalidades opostas. Enquanto a criptografia se dedica ao desenvolvimento de técnicas para proteger mensagens e dados, transformando informações legíveis em códigos cifrados por meio de algoritmos matemáticos, a criptoanálise tem como objetivo analisar os códigos gerados pela criptografia e, quando possível, descobrir a mensagem original sem conhecer a chave usada. Ela pode ser aplicada de forma investigativa, por exemplo, para testar a segurança de um sistema, ou em situações de tentativa de acesso não autorizado a informações protegidas. Ambas são fundamentais para o avanço e aprimoramento dos sistemas de segurança digital.

Singh (2001) destaca que a criptoanálise só pôde ser desenvolvida depois que a civilização atingiu um nível suficientemente sofisticado de estudo, em várias

6 Al-Kindi (c. 801 – c. 873) foi um filósofo e cientista árabe do califado Abássida, com contribuições em diversas áreas do conhecimento. É considerado o pioneiro da criptanálise por aplicar métodos estatísticos na quebra de cifras, especialmente por meio da análise de frequência, descrita em sua obra sobre Cifras Criptográficas.

disciplinas, incluindo matemática, estatística e linguística. A análise de frequência consiste em observar quais símbolos aparecem com mais frequência em uma mensagem criptografada e comparar com as letras mais comuns da língua original. Em português, por exemplo, a letra “A” aparece com mais frequência, seguida de “E”, “O”, “S” e “R”. Isso é sintetizado na tabela abaixo:

Tabela 1. Frequência das letras na língua portuguesa

Letra	%	Letra	%	Letra	%
A	14,64	G	1,30	N	5,05
T	4,34	B	1,04	H	1,28
O	10,73	U	4,64	C	3,88
I	6,18	P	2,52	V	1,70
D	4,10	J	0,40	Q	1,20
X	0,21	E	12,57	L	2,78
R	6,53	Z	0,47	F	1,02
M	4,75	S	7,81		

Grupo de Teleinformática e Automação (GTA) – UFRJ. Decifrando Textos em Português.

Com base na frequência média das letras na língua portuguesa, como a alta incidência de letras como A (14,64%), E (12,57%) e O (10,73%), é possível aplicar técnicas de criptoanálise para decifrar cifras de substituição simples. Abramo Hefez (2022) ilustra esse processo ao apresentar a seguinte mensagem cifrada:

“SUMZFI GCSGC SVZFC LZLSJ EZQSL HIFUI JDZQS LTSRF SGCSJ UOZSZ...”.

A criptoanálise da mensagem foi conduzida com base na técnica de análise de frequência, que parte do princípio estatístico de que as letras de um idioma natural não ocorrem com igual probabilidade. Ao examinar a distribuição das letras na mensagem cifrada, observou−se a seguinte frequência percentual, considerando um total de 51 caracteres alfabéticos: S (19,61%), Z (13,73%), F (7,84%), C (7,84%), L (7,84%), G (5,88%), J (5,88%), U (5,88%), Q (3,92%), I (3,92%), T (3,92%), H (1,96%), D (1,96%), O (1,96%) e V (1,96%).

Observa−se que a letra S ocorre com quase 20% de frequência, muito acima da média de qualquer letra em português. Essa informação indica que S representa uma letra muito comum na língua original. Embora tanto E quanto A

apresentem frequências médias aproximadas na língua portuguesa, opta−se inicialmente por associar S → E por três motivos:

i. S é significativamente mais frequente na cifra do que qualquer outra letra;

ii. palavras decifradas como “EZQSL” se tornam “ESTAR” se assumirmos $S = E$;

iii. tentativas alternativas como $S = A$ geram reconstruções menos naturais e menos compatíveis com a estrutura do português.

Com essa substituição e outras por análise estrutural de palavras repetidas, como *“GCSGC” → “SOBRE”,* foi possível construir gradualmente um alfabeto de substituição entre o texto cifrado e a língua portuguesa. A mensagem original recuperada foi:

"ESTUDO SOBRE CIFRA CESAR UTILIZADA PARA ENSINO BASICO SOBRE CODIGO SIMPLES..."

Esse processo ilustra como o conhecimento estatístico da linguagem pode ser utilizado como ferramenta de decodificação, revelando os limites da segurança de cifras monoalfabéticas simples. Também demonstra, do ponto de vista pedagógico, o potencial da criptoanálise como instrumento para introduzir alunos à interseção entre matemática, linguística e segurança da informação.

1.4 A Revolução da Cifra de Vigenère

No século XVI, durante o Renascimento, Blaise de Vigenère[7] desenvolveu uma cifra revolucionária que viria a ser considerada indecifrável por mais de dois séculos. Ao contrário da cifra de César, que utiliza um deslocamento fixo para todas as letras da mensagem, a cifra de Vigenère emprega um método polialfabético, no qual cada letra do texto claro é cifrada com base em uma letra diferente da palavra−chave. Isso significa que o deslocamento aplicado a cada caractere varia conforme a letra correspondente da chave, tornando o sistema muito mais resistente à análise de frequência.

[7] Vigenère (1523–1596) foi um criptógrafo francês que propôs, em 1586, um método polialfabético baseado no uso de uma palavra-chave.

Para aplicar esse método, é necessário primeiro atribuir valores numéricos às letras do alfabeto, conforme a tabela abaixo:

Tabela 2. Codificação Alfabética com Valores Inteiros de A = 0 a Z = 25

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

Fonte: Elaborado pelo autor com base em COUTINHO (2009)

A cifra de Vigenère é então aplicada com base na seguinte fórmula matemática:

$$c_i \equiv p_i + k_i \pmod{26}$$

Onde:

- c_i é o valor numérico da i−ésima letra cifrada;
- p_i é o valor numérico da i−ésima letra do texto claro;
- k_i é o valor numérico da i−ésima letra da chave, repetida conforme necessário para igualar o comprimento da mensagem;
- $\mathrm{mod}\ 26$ garante que o resultado esteja dentro dos limites do alfabeto.

Considere o seguinte exemplo, no qual a mensagem a ser cifrada é "CRIPTOGRAFIA" e a palavra-chave utilizada é "CHAVE":

- Texto claro: CRIPTOGRAFIA
- Palavra−chave: CHAVECHAVECH

Observe que o texto-claro e a palavra-chave devem ter o mesmo número de caracteres. Assim, torna-se necessário repetir a chave até que todos os caracteres da mensagem estejam relacionados. Aplicando os valores, conforme a tabela abaixo, temos:

Tabela 3. Etapas da cifragem na cifra de Vigenère (com operações mod 26)

Posição (i)	Letra do Texto	Valor (P_i)	Letra da Chave	Valor (K_i)	Soma = $P_i + K_i$	Módulo 26	Letra Cifrada (C_i)
1	C	2	C	2	4	4	**E**
2	R	17	H	7	24	24	**Y**
3	I	8	A	0	8	8	**I**

4	P	15	V	21	36	10	**K**
5	T	19	E	4	23	23	**X**
6	O	14	C	2	16	16	**Q**
7	G	6	H	7	13	13	**N**
8	R	17	A	0	17	17	**R**
9	A	0	V	21	21	21	**V**
10	F	5	E	4	9	9	**J**
11	I	8	C	2	10	10	**K**
12	A	0	H	7	7	7	**H**

Fonte: Elaborado pelo autor com base em Hefez (2022) e Stallings (2006).

Dessa forma, ao aplicar o método de Vigenère com a palavra-chave “CHAVE” sobre a mensagem “CRIPTOGRAFIA”, obtém-se o texto cifrado final “EYIKXQNRVJKH”**.**

Essa técnica permite ao professor explorar com os alunos o conceito de funções modulares, operações com congruência, e padrões algébricos, além de promover habilidades como organização de tabelas, codificação e decodificação, sendo esta última realizada pela relação:

$$p_i \equiv c_i - k_i \pmod{26}$$

Ao trabalhar com cifras como a de Vigenère, os estudantes praticam matemática de forma contextualizada, criativa e desafiadora, o que estimula tanto o engajamento quanto o pensamento lógico. Esse mecanismo elimina padrões simples e dificulta ataques baseados em frequência. No entanto, no século XIX, Charles Babbage[8] e Friedrich Kasiski[9] desenvolveram técnicas para quebrar a cifra de Vigenère, baseando−se na repetição de padrões e análise combinatória.

Durante o século XX, especialmente no contexto das duas grandes Guerras, a criptografia passou a desempenhar um papel estratégico e decisivo nas operações militares. Um dos exemplos mais emblemáticos desse período foi o uso da máquina Enigma pela Alemanha nazista. A Enigma era uma máquina eletromecânica baseada em sistemas de rotores, capazes de realizar múltiplas substituições de letras, com milhares de combinações possíveis. Seu uso diário exigia que os operadores

[8] Charles Babbage (1791–1871), matemático e inventor britânico, é considerado o pai da computação por idealizar a máquina analítica, precursora dos computadores modernos. Também foi o primeiro a quebrar a cifra de Vigenère, embora não tenha publicado sua descoberta.
[9] Friedrich Kasiski (1805–1881), oficial militar e matemático amador prussiano, foi o primeiro a publicar, em 1863, um método eficaz para quebrar essa cifra, conhecido como teste de Kasiski.

atualizassem chaves e posições dos rotores, o que tornava a decifração extremamente difícil. Como explica Coutinho (2009), alguns dos primeiros computadores foram montados exatamente para auxiliar na decifração dos códigos secretos usados pelos alemães durante a 2ª Guerra Mundial. Um desses esforços foi liderado pelo matemático britânico Alan Turing[10], cujas contribuições para a quebra da Enigma foram fundamentais para o avanço da computação moderna.

Figura 4. Maquina Enigma

Fonte: The National Museum Of Computing. The Enigma Machine.

A Figura acima apresenta uma das versões históricas da máquina Enigma, preservada atualmente no The National Museum of Computing, no Reino Unido. A imagem revela o intrincado sistema mecânico do equipamento, composto por teclado, rotores intercambiáveis, painel de lâmpadas e conexões elétricas internas, que permitiam a geração de milhares de combinações para criptografar mensagens. A decifração da Enigma envolveu não apenas a genialidade matemática de Alan Turing e sua equipe em Bletchley Park[11], mas também a aplicação prática de conceitos matemáticos avançados, como análise combinatória, probabilidade, teoria da informação e lógica computacional, saberes que se consolidariam como pilares da criptografia moderna e da ciência da computação.

[10] Alan Turing (1912–1954), matemático e lógico britânico, é considerado um dos fundadores da ciência da computação moderna.

[11] Bletchley Park foi o centro de inteligência britânico durante a Segunda Guerra Mundial, onde foram conduzidas operações de criptanálise fundamentais para a quebra de códigos inimigos. Atualmente, é um museu dedicado à história da criptografia.

A operação para quebrar os códigos da Enigma mobilizou matemáticos, engenheiros e linguistas, sendo considerada um marco histórico no desenvolvimento dos computadores (Singh, 2007). Nesse mesmo sentido, Coutinho (2009) destaca que alguns dos primeiros computadores foram montados exatamente para auxiliar na decifração dos códigos secretos usados pelos alemães durante a 2ª Guerra Mundial. Ao incorporar essa imagem ao contexto educacional, pode−se incentivar reflexões sobre o papel da matemática na resolução de problemas concretos, na evolução tecnológica e em eventos históricos de impacto global.

1.5 Criptografia Moderna: Simétrica, Assimétrica e RSA

A partir do final da 2ª Guerra Mundial, com o avanço da computação e o início da era digital, a criptografia evoluiu rapidamente. Surgiram dois grandes modelos: a criptografia simétrica e a criptografia assimétrica. Na criptografia simétrica, utiliza−se uma mesma chave para cifrar e decifrar. O maior desafio nesse modelo é o transporte seguro da chave. Já a criptografia assimétrica, proposta por Diffie e Hellman[12] em 1976, utiliza um par de chaves: uma pública, para cifrar, e uma privada para decifrar.

O sistema mais conhecido de criptografia assimétrica é o RSA, criado por Rivest, Shamir e Adleman[13] em 1978. Sua segurança está baseada na dificuldade de fatorar o produto de dois números primos. Como observa Coutinho (2009), em um código de chave pública, o conhecimento do processo de codificação não garante automaticamente a capacidade de decodificação. Esse princípio abriu caminho para o desenvolvimento de sistemas robustos, como o protocolo HTTPS, os certificados digitais, as assinaturas eletrônicas e as criptomoedas.

Atualmente, a criptografia está presente em praticamente todos os aspectos da vida digital. Ela sustenta a segurança de transações bancárias, garante a privacidade em trocas de mensagens em aplicativos como WhatsApp e Signal, protege os dados transmitidos em redes Wi-Fi e fortalece os mecanismos de

[12] Whitfield Diffie e Martin Hellman são criptógrafos norte-americanos que, em 1976, propuseram o conceito de criptografia de chave pública, marco fundamental para a segurança digital moderna.
[13] Ron Rivest, Adi Shamir e Leonard Adleman são os criadores do algoritmo RSA, desenvolvido em 1977, considerado um dos sistemas de criptografia de chave pública mais utilizados no mundo.

blockchain e contratos inteligentes. Assim, sua presença se tornou tão essencial para a vida cotidiana quanto, muitas vezes, invisível aos usuários.

1.6 A Criptografia como Recurso Pedagógico

Historicamente, a criptografia esteve amplamente vinculada à necessidade de proteger informações e garantir o sigilo em contextos militares, diplomáticos e comerciais (Kahn, 1995). Contudo, além dessas aplicações tradicionais, sua relevância expandiu-se significativamente para o campo educacional, configurando−se como uma poderosa ferramenta didática no ensino da Matemática e em áreas correlatas.

A utilização de técnicas criptográficas no contexto escolar pode contribuir substancialmente para o enriquecimento do processo de ensino−aprendizagem. Por meio da codificação e decodificação de mensagens secretas, os professores são capazes de proporcionar experiências lúdicas e investigativas aos estudantes, tornando conceitos abstratos da matemática mais concretos e estimulando habilidades como o raciocínio lógico, a resolução de problemas, a interdisciplinaridade e o pensamento crítico (Stallings, 2006).

Cifras clássicas, tais como a cifra de César ou a cifra de Vigenère, podem ser introduzidas em sala de aula como estratégias pedagógicas para facilitar a compreensão de conteúdos fundamentais como aritmética modular, análise combinatória, padrões numéricos e análise de frequência (Hefez, 2022). Além disso, tais práticas permitem ao aluno perceber a Matemática não como um conjunto isolado de regras, mas como uma disciplina integrada às questões práticas do cotidiano e às necessidades da comunicação segura e eficiente.

Nesse sentido, as diretrizes da Base Nacional Comum Curricular (BNCC) são particularmente alinhadas ao uso da criptografia como instrumento didático, pois enfatizam o desenvolvimento de competências relacionadas à argumentação fundamentada em dados, resolução de problemas, uso crítico e criativo de recursos digitais, e reconhecimento da aplicação prática da Matemática em situações reais. Dessa forma, a criptografia atua como uma ferramenta intermediária que conecta diretamente o conhecimento matemático formal às necessidades contemporâneas de alfabetização digital e segurança da informação.

CAPÍTULO 2 – CONCEITOS FUNDAMENTAIS DE ARITMÉTICA

O estudo da Teoria dos Números começa com a análise cuidadosa das propriedades dos inteiros e, em particular, da noção de divisibilidade. Segundo Oliveira (2007), a aritmética dos inteiros não é apenas a base da matemática escolar, mas também a porta de entrada para construções mais profundas da matemática abstrata, como a própria criptografia. A primeira ideia essencial é a da divisão exata entre inteiros. Dados $a, b \in \mathbb{Z}$, dizemos que a divide b se existe um número inteiro k tal que $b = a \cdot k$. Escrevemos isso como $a \mid b$. Por exemplo, como $3 \cdot 5 = 15$, temos que $3 \mid 15$ e $5 \mid 15$, mas $4 \nmid 5$.

Essa simples definição é o alicerce de toda a estrutura da teoria dos números. Oliveira (2007, p. 5) observa que “a relação de divisibilidade entre inteiros é reflexiva, transitiva e não simétrica”, e explora as consequências dessas propriedades em teoremas subsequentes. Outro ponto fundamental destacado pelo autor é o Algoritmo da Divisão, que estabelece que, dados inteiros a e b, com $b > 0$, existem inteiros q (quociente) e r (resto) tais que:

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

Esse resultado não apenas garante a possibilidade de dividir com resto qualquer número por outro positivo, mas também fornece a base para o Algoritmo de Euclides, método mais eficiente e antigo de calcular o máximo divisor comum (mdc) entre dois números.

De fato, o Algoritmo de Euclides foi descrito por volta de 300 a.C. em “Os Elementos[14]”, e permanece até hoje como uma das ferramentas centrais para lidar com problemas envolvendo divisibilidade. Ele se fundamenta na seguinte propriedade:

$$\text{mdc}\,(a, b) = \text{mdc}\,(b, r)$$

onde r é o resto da divisão de a por b. A repetição sucessiva dessa operação até que

14 "Os Elementos" (Elementa em latim, Stoicheia em grego) é um livro matemático clássico escrito por Euclides de Alexandria por volta do século III a.C. É considerado uma das obras mais influentes da história da matemática e da ciência.

o resto se torne zero leva ao maior divisor comum entre os dois números originais. Por exemplo, para calcular o mdc entre 119 e 34, temos:

$$119 = 34 \cdot 3 + 17,$$
$$34 = 17 \cdot 2 + 0.$$

Logo, mdc$(119, 34) = 17$.

Ao longo deste capítulo, serão apresentados, com base em Oliveira (2007), os principais conceitos relacionados à divisibilidade, números primos, critérios práticos de divisibilidade e o papel da fatoração única. Cada conceito será acompanhado de proposições, demonstrações formais e exemplos que formam o alicerce teórico necessário para a construção de sistemas criptográficos, tema central desta dissertação.

2.1 Divisibilidade

Nesta seção serão abordados os conceitos fundamentais relacionados à divisibilidade no conjunto dos números inteiros $\mathbb{Z}$.

Definição 2.1 Se a e b são inteiros, dizemos que a divide b, denotando por $a \mid b$, se existir um inteiro c tal que $b = ac$. Se a não divide b escrevemos $a \nmid b$.

Proposição 2.1. Se a, b e c são inteiros, tal que $a \mid b$ e $b \mid c$, então $a \mid c$.

Demonstração: *Como $a \mid b$ e $b \mid c$, existem inteiros k_1 e k_2 com $b = k_1a$ e $c = k_2b$. Substituindo o valor de b na equação $c = k_2b$, teremos $c = k_2k_1a$ o que implica $a \mid c$.* □

Por exemplo, como 3 divide 12 e 12 divide 48, segue−se que 3 também divide 48. Essa relação é expressa simbolicamente como: $3 \mid 12, 12 \mid 48 \Rightarrow 3 \mid 48$. Por outro lado, como não existe nenhum número inteiro c tal que $15 = 4 \cdot c$, concluímos que 4 não divide 15. Em notação matemática: $4 \nmid 15$.

Proposição 2.2. Se a, b, c, m e n são inteiros, $c \mid a$ e $c \mid b$ então $c \mid (ma + nb)$, $\forall m,n \in \mathbb{Z}$.

Demonstração: *Se $c \mid a$ e $c \mid b$ então $a = k_1c$ e $b = k_2c$. Multiplicando−se estas duas equações respectivamente por m e n teremos $ma = mk_1c$ e $nb = nk_2c$. Somando−se membro a membro obtemos $ma + nb = (mk_1 + nk_2)c$, o que nos diz que $c \mid (ma + nb)$.* □

Por exemplo, sejam $a = 15$, $b = 42$, $c = 3$, $m = 8$ e $n = 7$. Como 3 divide 15 e 3 divide 42, então 3 também divide a expressão $8 \cdot 15 + 7 \cdot 42$. Assim, em notação matemática temos que $3 \mid (8 \cdot 15 + 7 \cdot 42)$.

Teorema 2.1 A divisibilidade tem as seguintes propriedades, nas quais os elementos a, d e n pertencem a $\mathbb{Z}$:

i. $n \mid n$

ii. $d \mid n \Rightarrow ad \mid an$

iii. $ad \mid na$, e $a \neq 0 \Rightarrow d \mid n$

iv. $1 \mid n \ \forall n$

v. $n \mid 0 \ \forall n$

vi. $d \mid n$ e $n \neq 0 \Rightarrow |d| \leq |n|$

vii. $d \mid n$ e $n \mid d \Rightarrow |d| = |n|$

viii. $d \mid n$ e $d \neq 0 \Rightarrow (n/d) \mid n$.

Demonstração:

Serão apresentadas apenas as demonstrações dos itens (i), (ii) e (vii), escolhidas por destacarem os argumentos centrais do resultado. As demais demonstrações não são exibidas, pois seguem de maneira análoga, utilizando os mesmos princípios e técnicas já ilustrados nos casos selecionados.

(i) Como $n = 1.n$ segue da definição que $n \mid n$, inclusive para $n = 0$.

(ii) Se $d \mid n$ então $n = cd$ para algum inteiro c. Logo $an = cad$, o que conclui a demonstração.

(vii). Se $d \mid n$ então $n = k_1 d$ e portanto $n \mid d$ é um inteiro. Como $(n/d) \cdot d = n$, segue da definição que $(n/d) \mid n$. □

2.2 Algoritmo da Divisão

Antes da seção 2.2, apresenta−se o chamado Teorema de Eudoxius[15]: Dados a e b inteiros com $b \neq 0$ então a é um múltiplo de b ou se encontra entre dois

[15] Este resultado costuma ser erroneamente atribuído a Arquimedes e chamado "Princípio de Arquimedes"

múltiplos consecutivos de b, isto é, correspondendo a cada par de inteiros a e $b \neq 0$ existe um inteiro q tal que, para $b > 0$,

$$qb \leq a < (q + 1)b$$

e para $b < 0$,

$$qb \leq a < (q - 1)b.$$

Por exemplo, se $a = 11$ e $b = 4$, devemos tomar $q = 2$

$$2 \cdot 4 \leq 11 < 3 \cdot 4.$$

Para $a = -11$ e $b = 4$, tomamos $q = -3$

$$-3 \cdot 4 \leq -11 < (-3 + 1) \cdot 4.$$

Se $a = 11$ e $b = -4$, tomamos $q = -2$

$$(-2) \cdot (-4) \leq 11 < (-2 - 1) \cdot (-4).$$

Para $a = -11$ e $b = -4$, tomamos $q = 3$

$$3 \cdot (-4) \leq -11 < (3 - 1) \cdot (-4).$$

Teorema 2.2 Dados dois inteiros a e b, $b > 0$, existe um único par de inteiros q e r, de modo que q é chamado de quociente e r de resto da divisão de a por b, tais que

$$a = qb + r, \quad \text{com } 0 \leq r < b \ (r = 0 \Leftrightarrow b \mid a)$$

Demonstração: *Pelo Teorema de Eudoxius, dados dois números inteiros a e b, com $b \neq 0$ e $b > 0$, existe um inteiro q tal que*

$$qb \leq a < (q + 1)b,$$

o que implica $0 \leq a - qb$ e $a - qb < b$. Desta forma, se definirmos $r = a - qb$, teremos, garantida, a existência de q e r. A fim de mostrarmos a unicidade, vamos supor a existência de outro par q_1 e r_1 verificando:

$$a = q_1 b + r_1 \quad com \quad 0 \leq r_1 < b.$$

Disto temos $(qb + r) - (q_1 b + r_1) = 0 \Rightarrow b\,(q - q_1) = r_1 - r$, *o que implica* $b \mid (r_1 - r)$. *Mas, como* $0 \leq r_1 < b$ *e* $0 \leq r < b$, *temos* $|r_1 - r| < b$ *e, portanto, como* $b \mid (r_1 - r)$ *devemos ter* $r_1 - r = 0$ *o que implica* $r = r_1$. *Logo* $q_1 b = qb \Rightarrow q_1 = q$, *uma vez que* $b \neq 0$. □

Observação: Embora no enunciado do Teorema 2.2 exista a restrição $b > 0$, isto não é necessário e, utilizando−se a equação, $qb \leq a < (q - 1)b$, teríamos encontrado q e r também para $b < 0$. Podemos, pois, enunciar o Algoritmo da Divisão de Euclides da seguinte forma: Dados dois inteiros a e b, $b \neq 0$ existe um único par de inteiros q e r tais que $a = qb + r$ com $0 \leq r < |b|$.

Por exemplo, se $a = 36$ e $b = 6$, temos que $36 = 6 \cdot 6$, ou seja, 6 divide 36. Isso mostra que 36 é um múltiplo exato de 6. Por outro lado, se a não for múltiplo de b, então ele se encontra entre dois múltiplos consecutivos de b. Por exemplo, tomando $a = 20$ e $b = 7$, temos que os múltiplos mais próximos de 7 são $7 \cdot 2 = 14$ e $7 \cdot 3 = 21$. Como $14 < 20 < 21$, concluímos que 20 está entre dois múltiplos consecutivos de 7.

2.3 O Máximo Divisor Comum (MDC)

O máximo divisor comum de dois inteiros a e b (a ou b diferente de zero), denotado por (a, b), é o maior inteiro que divide a e b.

Teorema 2.3 Identidade de Bezout Seja d o máximo divisor comum de a e b, então existem inteiros n_0 e m_0 tais que $d = n_0 a + m_0 b$.

Demonstração: *Seja* B *o conjunto de todas as combinações lineares* $(na + mb)$ *onde* n *e* m *são inteiros. Este conjunto contém, claramente, números negativos, positivos e também o zero. Vamos escolher* n_0 *e* m_0 *tais que* $c = n_0 a + m_0 b$ *seja o menor inteiro positivo pertencente ao conjunto* B. *Vamos provar que* $c \mid a$ *e* $c \mid b$. *Como as demonstrações são similares, mostraremos apenas que* $c \mid a$. *A prova é por contradição. Suponhamos que* $c \nmid a$. *Neste caso, pelo Teorema 2.2, existem* q *e* r *tais que* $a = qc + r$ *com* $0 < r < c$. *Portanto*

$$r = a - qc = a - q\,(n_0 a + m_0 b) = (1 - qn_0)a + (-qm_0)b.$$

Isto mostra que $r \in B$, *pois* $(1 - qn_0)$ *e* $(-qm_0)$ *são inteiros, o que é uma contradição, uma vez que* $0 < r < c$ *e* c *é o menor inteiro positivo de* B. *Logo* $c \mid a$ *e de forma análoga se prova que* $c \mid b$.

Como d *é um divisor comum de* a *e* b, *existem inteiros* k_1 *e* k_2 *tais que*

$$a = k_1 d \text{ e } b = k_2 d$$

e, portanto,

$$c = n_0 a + m_0 b = n_0 k_1 d + m_0 k_2 d = d(n_0 k_1 + m_0 k_2)$$

o que implica $d \mid c$.

Do Teorema 2.1 (vi), temos que $d \leq c$ *(ambos são positivos) e como d < c não é possível, uma vez que* d *é o máximo divisor comum, concluímos que*

$$d = n_0 a + m_0 b.$$ □

Por exemplo, sejam $a = 30$ e $b = 18$. Vamos determinar o máximo divisor comum de a e b e expressá−lo como uma combinação linear de a e b. Aplicando o Algoritmo de Euclides:

$$30 = 18 \cdot 1 + 12$$
$$18 = 12 \cdot 1 + 6$$
$$12 = 6 \cdot 2 + 0 \quad \Rightarrow \quad \text{mdc}\,(30,18) = 6$$

Agora, voltamos os passos para obter 6 como combinação linear de 30 e 18.
Da segunda linha:

$$6 = 18 - 12 \cdot 1$$

Mas da primeira linha:

$$12 = 30 - 18 \cdot 1$$

Substituindo 12 na segunda linha:

$$6 = 18 - (30 - 18)$$

Colocando o 18 em evidências:

$$6 = 2 \cdot 18 - 30$$

Portanto:

$$6 = (-1) \cdot 30 + 2 \cdot 18$$

Logo, o máximo divisor comum de 30 e 18 pode ser expresso como uma combinação linear com $n_0 = -1$ e $m_0 = 2$, denotado por $6 = (-1) \cdot 30 + 2 \cdot 18$

Na demonstração deste teorema mostramos, não apenas que o máximo divisor comum de a e b pode ser expresso como uma combinação linear destes números, mas que este número é o menor valor positivo dentre todas estas combinações lineares. O teorema seguinte nos dá uma outra caracterização para o máximo divisor comum de dois números.

Teorema 2.4 O máximo divisor comum d de a e b é o divisor positivo de a e b o qual é divisível por todo divisor comum.

Demonstração: *Do teorema anterior e pela Proposição 2.1, concluímos que, se, d_1 é divisor comum de a e b, então $d_1 \mid d$. Portanto, não podem existir dois números distintos tendo cada um a propriedade de ser divisível por todo divisor comum. Isto por causa do Teorema 2.1 (vii) que, no caso de números positivos d_1 e d, nos diz que d_1 deve ser igual a d.* □

Proposição 2.3 Para todo inteiro positivo t, tem-se $(ta, tb) = t(a, b)$.

Demonstração: *Pelo Teorema 2.3, (ta, tb) é o menor valor positivo de $mta + ntb$ com (m e n inteiros), que é igual a t vezes o menor valor positivo de $ma + nb = t(a, b)$. Assim,* $(ta, tb) = t(a, b)$. □

Por exemplo, sejam $a = 8$, $b = 14$ e $t = 3$. Sabemos que $\text{mdc}\,(a, b) = 2$ e que $\text{mdc}\,(ta, tb) = \text{mdc}\,(24, 42) = 6$, o que confirma a validade da identidade $(ta, tb) = t(a, b)$ pois $3 \cdot 2 = 6$. Além disso, como $2 = (-5) \cdot 8 + 3 \cdot 14$, multiplicando ambos os lados por $t = 3$, obtemos $6 = (-15) \cdot 8 + 9 \cdot 14$, ou seja, 6 também pode ser expresso como uma combinação linear de a e b, o que demonstra que (ta, tb) é igual a t vezes o menor valor positivo de $ma + nb$, com $m,n \in \mathbb{Z}$.

Proposição 2.4 Se $c > 0$ e a e b são divisíveis por c, então

$$\left(\frac{a}{c}, \frac{b}{c}\right) = \frac{1}{c}\,(a, b)$$

Demonstração: *Como a e b são divisíveis por c, temos que a/c e b/c são inteiros. Basta, então, substituir na Proposição 2.3 "a" por "a/c" e "b" por "b/c" tomando $t = c$.*

□

Corolário: Se $(a, b) = d$, temos que $\left(\frac{a}{d}, \frac{b}{d}\right) = 1$.

Demonstração: *No que acabamos de demonstrar c é um divisor comum de a e b. Se tomarmos c como sendo o máximo divisor comum d, teremos o resultado desejado.* □

Por exemplo, sejam a = 14, b = 35 e c = 7. Como a e b são divisíveis por c, temos que $\frac{a}{c} = 2$ e $\frac{b}{c} = 5$, que são inteiros. Calculando os máximos divisores comuns, obtemos $(a, b) = (14, 35) = 7$, e, portanto, $\frac{1}{c} \cdot (a, b) = \frac{1}{7} \cdot 7 = 1$. Por outro lado, $\left(\frac{a}{c}, \frac{b}{c}\right) = (2, 5) = 1$. Assim, verifica−se que $\left(\frac{a}{c}, \frac{b}{c}\right) = \frac{1}{c}(a, b)$, como afirma a proposição.

Definição 2.2 Os inteiros a e b são relativamente primos quando $(a, b) = 1$.

Teorema 2.5 Para a, b e x inteiros temos $(a, b) = (a, b + ax)$.

Demonstração: *Sejam $d = (a, b)$ e $f = (a, b + ax)$. Pelo Teorema 2.3 existem inteiros n_0 e m_0, tais que $d = n_0a + m_0b$ e esta expressão pode ser escrita como*

$$d = a\,(n_0 - xm_0) + (b + ax)\,m_0,$$

concluímos que o máximo divisor f de a e $b + ax$ é um divisor de d. Tendo mostrado que $f \mid d$. Por outro lado, pela Proposição 2.2, $d \mid (b + ax)$ e pelo Teorema 2.4, todo divisor comum de a e $b + ax$ é um divisor de f. Tendo, assim, provado que $d \mid f$ concluímos, pelo Teorema 2.1 (vii), que $d = f$, uma vez que ambos são positivos. □

Um exemplo que ilustra o teorema acima é dado pelos números $a = 3$ e $b = 15$. Como $\mathrm{mdc}(3, 15) = 3$, temos que esse valor se mantém inalterado mesmo quando adicionamos ou subtraímos múltiplos de 3 ao número 15. Por exemplo:

$$(3, 15) = (3, 15 + 4 \cdot 3) = (3, 27) = 3,$$
$$(3, 15 + 7 \cdot 3) = (3, 36) = 3,$$
$$(3, 15 - 8 \cdot 3) = (3, -9) = 3.$$

Esses exemplos confirmam que o máximo divisor comum entre a e b é o mesmo entre a e $b + ka$, independentemente do valor de k, positivo ou negativo. Essa propriedade é especialmente útil em demonstrações que envolvem o algoritmo de Euclides ou simplificação de expressões de mdc.

Teorema 2.6 Se $a \mid bc$ e $(a, b) = 1$, então $a \mid c$.

Demonstração: *Como $(a, b) = 1$, pelo Teorema 2.3, existem inteiros n e m tais que $na + mb = 1$. Multiplicando−se ambos lados desta igualdade por c, temos, $n(ac) + m(bc) = c$. Como $a \mid ac$ e, por hipótese, $a \mid bc$ então, pela Proposição 2.2, $a \mid c$.* □

Considere os inteiros $a = 4$, $b = 27$ e $c = 20$. Como $\mathrm{mdc}(4, 27) = 1$ e 4 divide $(27 \cdot 20) = 540$, podemos aplicar o Teorema 2.6 para concluir que 4 divide 20. De fato, 540 é divisível por 4 e, como 4 e 27 são primos entre si, a divisibilidade de 4 por $27 \cdot 20$ implica necessariamente que 4 divide 20. Este exemplo evidencia como a condição de coprimalidade entre a e b garante que o fator a se propague diretamente até c, mesmo sendo inicialmente conhecido apenas que $a \mid bc$.

Teorema 2.7 Se a e b são inteiros e $a = qb + r$ onde q e r são inteiros, então $(a, b) = (b, r)$.

Demonstração: *Da relação $a = qb + r$, podemos concluir que todo divisor de b e r é também um divisor de a (Proposição 2.2). Esta mesma relação, escrita como $r = a - qb$, nos diz que todo divisor de a e b é um divisor de r. Logo, o conjunto dos divisores comuns de a e b é igual ao conjunto dos divisores comuns de b e r, o que nos garante o resultado $(a, b) = (b, r)$.* □

Considere o exemplo aplicado ao algoritmo de Euclides para $a = 50$ e $b = 18$. Temos que $50 = 2 \cdot 18 + 14$, ou seja, $q = 2$ e $r = 14$. Logo, $\mathrm{mdc}(50, 18) = \mathrm{mdc}(18, 14)$. Continuando o processo temos

$$18 = 1 \cdot 14 + 4 \quad \rightarrow \mathrm{mdc}(18, 14) = \mathrm{mdc}(14, 4);$$
$$14 = 3 \cdot 4 + 2 \quad \rightarrow \mathrm{mdc}(14,\ 4) = \mathrm{mdc}(4, 2);$$
$$4 = 2 \cdot 2 + 0 \quad \rightarrow \mathrm{mdc}(\ 4,\ 2) = 2.$$

Tendo encontrado, desta forma, o máximo divisor comum de 50 e 18 que é o último resto não−nulo da sequência de igualdades acima.

2.4 O Algoritmo de Euclides

Teorema 2.8 Sejam $r_0 = a$ e $r_1 = b$ inteiros não negativos com $b \neq 0$. Se o Algoritmo da Divisão for aplicado sucessivamente, obtendo-se

$$r_j = q_j + {}_1 r_{j+1} + r_{j+2},\ 0 \leq r_{j+2} < r_{j+1},$$

para $j = 0, 1, 2, \ldots, n-1$ e $r_{n+1} = 0$, então $(a, b) = r_n$, o último resto não nulo.

Demonstração: *Tendo em mente o exemplo anterior fica fácil acompanhar a demonstração deste algoritmo. Vamos, inicialmente, aplicar o Teorema 2.2 para dividir* $r_0 = a$ *por* $r_1 = b$ *obtendo* $r_0 = q_1 r_1 + r_2$, *em seguida dividimos* r_1 *por* r_2 *obtendo* $r_1 = q_2 r_2 + r_3$ *e assim, sucessivamente, até a obtenção do resto* $r_{n+1} = 0$. *Como, a cada passo o resto é sempre menor do que o anterior, e estamos lidando com números inteiros positivos, é claro que após um número finito de aplicações do Teorema 2.2, teremos resto nulo.*

Temos, a seguinte sequência de equações:

$$\begin{aligned}
r_0 &= q_1 r_1 + r_2 && 0 < r_2 < r_1,\\
r_1 &= q_2 r_2 + r_3 && 0 < r_3 < r_2,\\
r_2 &= q_3 r_3 + r_4 && 0 < r_4 < r_3,\\
&\vdots\\
r_{n-2} &= q_{n-1} r_{n-1} + r_n && 0 < r_n < r_{n-1},\\
r_{n-1} &= q_n r_n + 0.
\end{aligned}$$

A última destas equações nos diz, pelo Teorema 2.7, que o máximo divisor comum de r_n *e* r_{n-1} *é* r_n. *A penúltima, que este número é igual a* (r_{n-1}, r_{n-2}) *e, prosseguindo desta maneira teremos, por repetidas aplicações do Teorema 2.7, a sequência:*

$$r_n = (r_{n-1}, r_n) = (r_{n-2}, r_{n-1}) = \ldots = (r_1, r_2) = (r_0, r_1) = (a, b).$$

Portanto, o máximo divisor comum de a *e* b *é o último resto não nulo da sequência de divisões descrita.* □

Aplicando o Algoritmo de Euclides para $a = 1126$ e $b = 522$:

$$\begin{aligned}
1126 &= 2 \cdot 522 + 82\\
522 &= 6 \cdot 82 + 30\\
82 &= 2 \cdot 30 + 22\\
30 &= 1 \cdot 22 + 8\\
22 &= 2 \cdot 8 + 6
\end{aligned}$$

$$8 = 1 \cdot 6 + 2$$

$$6 = 3 \cdot 2 + 0$$

O último resto não nulo é 2. Portanto, $\text{mdc}(1126, 522) = 2$.

2.5 Teorema Fundamental da Aritmética

Teorema 2.9 (Teorema Fundamental da Aritmética) Todo inteiro maior do que 1 pode ser representado de maneira única (a menos da ordem dos fatores) como um produto de números primos.

Demonstração: *Se n é primo não há nada a ser demonstrado. Suponhamos, que, n seja composto. Seja $p_1 (p_1 > 1)$ o menor dos divisores positivos de n. Afirmamos que p_1 é primo. Isto é verdade, pois, caso contrário existiria p, $1 < p < p_1$ com $p \mid n$, contradizendo a escolha de p_1. Logo, $n = p_1 n_1$.*

Se n_1 for primo a prova está completa. Caso contrário, tomamos p_2 como o menor fator de n_1. Pelo argumento anterior, p_2 é primo e temos que $n = p_1 p_2 n_2$.

Repetindo este procedimento, obtemos uma sequência decrescente de inteiros positivos $n_1, n_2, \ldots, n_r$. Como todos eles são inteiros maiores do que 1, este processo deve terminar. Como os primos na sequência $p_1, p_2, \ldots, p_k$ não são, necessariamente, distintos, n terá, em geral, a forma

$$n = p_1^{a_1} p_2^{a_2} \ldots p_k^k.$$

Para mostrarmos a unicidade usamos indução em n. Para $n = 2$ a afirmação é verdadeira. Assumimos, então, que ela se verifica para todos os inteiros maiores do que 1 e menores do que n. Vamos provar que ela também é verdadeira para n. Se n é primo, não há nada a provar. Vamos supor, então, que n seja composto e que admita duas fatorações, isto é,

$$n = p_1 p_2 \ldots p_s = q_1 q_2 \ldots q_r.$$

Vamos provar que $s = r$ e que cada p_i é igual a algum q_j. Como p_1 divide o produto $q_1 q_2 \ldots q_r$, ele deve divide pelo menos um dos fatores q_j. Sem perda de generalidade, podemos supor que $p_1 \mid q_1$. Como são ambos primos, isto implica $p_1 =$

q_1. Logo $n/p_1 = p_2 \dots p_s = q_2 \dots q_r$. Como $1 < n/p_1 < n$, a hipótese de indução nos diz que as duas fatorações são idênticas, isto é, $s = r$ e, a menos da ordem, as fatorações $p_1p_2 \dots p_s$ e $q_1q_2 \dots q_s$ são iguais. □

2.6 Números Primos

Definição 1.3 Um número inteiro $n(n > 1)$ possuindo somente dois divisores positivos n e 1, é chamado primo. Se $n > 1$ não é primo, dizemos que n é composto.

Proposição 2.4. Se $p \mid ab$, p primo, então $p \mid a$ ou $p \mid b$.

Demonstração: *Se $p \nmid a$, então $(a, p) = 1$ o que implica, pelo Teorema 2.6, $p \mid b$.* □

Conforme Stewart (2015), os números primos são considerados “os átomos da matemática”, pois todos os inteiros positivos podem ser expressos como produto de primos. Por exemplo, o número 7 é primo, pois só é divisível por 1 e 7, já o 15 não é, pois admite outros divisores: 3 e 5.

Na área da criptografia, os primos têm papel fundamental. A dificuldade em fatorar grandes números que resultam do produto de primos sustenta a segurança de diversos sistemas criptográficos, como RSA.

Teorema 2.10 (Euclides) A sequência dos números primos é infinita.

Demonstração: *Vamos supor que a sequência dos primos seja finita. Seja, pois, p_1, p_2, p_3, …, p_n a lista de todos os primos. Consideramos o número $R = p_1 \cdot p_2 \cdot \dots \cdot p_n + 1$. É claro que R não é divisível por nenhum dos p_i de nossa lista e que R é maior do que qualquer p_i. Mas, pelo Teorema 2.9, ou R é primo ou possui algum fator primo e isto implica na existência de um primo que não pertence à nossa lista. Portanto a sequência dos números primos não pode ser finita.* □

A infinitude dos números primos foi demonstrada originalmente por Euclides, por meio de um argumento de contradição (Hardy & Wright, 1979). Esse teorema estabelece que não existe um número finito de primos, resultado que fundamenta a construção e a segurança de diversos algoritmos criptográficos modernos, garantindo a existência de primos suficientemente grandes para a geração de chaves seguras. Conforme o Teorema Fundamental da Aritmética, todo número natural maior que 1 é primo ou pode ser decomposto como produto de primos. Assim, o número r considerado na demonstração ou é ele mesmo primo e, portanto, não

pertence à lista finita original, ou possui um fator primo que também não está nela. Em ambos os casos, há uma contradição com a hipótese inicial de que $p_1, p_2, \ldots, p_n$ seriam todos os primos existentes, concluindo-se, portanto, que o conjunto dos números primos é infinito.

2.7 Do Elementar ao Essencial: uma síntese da aritmética

O presente capítulo reuniu os principais conceitos e resultados da Aritmética que fundamentam o estudo da Teoria dos Números. Foram discutidas noções fundamentais de divisibilidade, o Algoritmo da Divisão, o cálculo do máximo divisor comum por meio do Algoritmo de Euclides, além de propriedades essenciais dos números primos. A demonstração da infinitude dos números primos, baseada em argumento clássico de contradição, bem como o Teorema Fundamental da Aritmética, que assegura a fatoração única de inteiros em primos, encerram este capítulo destacando a estrutura elementar, mas profunda, do conjunto dos números inteiros.

Todos esses resultados são essenciais não apenas do ponto de vista teórico, mas também instrumental, pois servirão de base para o desenvolvimento de tópicos mais avançados, como a aritmética modular, as congruências e suas aplicações no campo da criptografia. O próximo capítulo tratará dos sistemas de congruências, estabelecendo conexões explícitas entre os resultados teóricos apresentados até aqui e suas aplicações no campo da criptografia, as quais serão aprofundadas nos capítulos finais.

CAPÍTULO 3 – FUNDAMENTOS MATEMÁTICOS NA CRIPTOGRAFIA

A Matemática, como ciência milenar, sempre evoluiu motivada pela constante busca humana por compreender, organizar e solucionar problemas concretos do cotidiano. Desde a contagem de dias, as medições de terras e os registros comerciais, até os complexos algoritmos criptográficos da atualidade, ela se manifesta em todas as dimensões da vida em sociedade. Dentre suas diversas áreas, destaca−se a Aritmética Modular, um ramo da Teoria dos Números que, apesar de sua aparente simplicidade, revela estruturas profundas e aplicações práticas surpreendentes. Conhecida também como aritmética do relógio, devido ao seu caráter cíclico, a Aritmética Modular está presente em sistemas computacionais, criptografia, controle de ciclos e calendários antigos.

Toda essa construção matemática está fundamentada principalmente nas obras de Coutinho (2009) e Oliveira (2007), cujas contribuições são referências clássicas no estudo da Teoria dos Números e suas aplicações em criptografia.

3.1 Aritmética Modular: Conceito e Propriedades

Ao longo da história, diversas contribuições enriqueceram o conhecimento acerca da Aritmética Modular. Contudo, foi Carl Friedrich Gauss (1777–1855) quem sistematizou o conceito de congruência no ano de 1801, ao publicar sua obra *Disquisitiones Arithmeticae*, com apenas 24 anos de idade (Gauss, 1801; Goldstein et al., 2007).

> Gauss foi um matemático, astrônomo e físico alemão que contribuiu muito em diversas áreas da ciência, dentre elas a Teoria dos Números, Estatística, Análise Matemática, Astronomia e Óptica. Alguns se referem a ele como ‘Príncipe da Matemática’. Ele considerava a Matemática como rainha das ciências. (Barbosa, 2017, p. 25)

O matemático Gauss introduziu o conceito de congruência ao observar que, dado um número natural m diferente de zero, dois números inteiros a e b são ditos congruentes módulo m quando apresentam o mesmo resto na divisão euclidiana por m. A importância histórica e didática da contribuição de Gauss reside no rigor matemático que aplicou à Aritmética Modular, tornando−a acessível e aplicável não apenas no campo abstrato, mas também em contextos práticos como a criptografia

moderna, segurança de dados e algoritmos computacionais. A partir dessa base teórica sólida, é possível apresentar agora as definições formais que sustentam esse campo.

Definição 3.1 Se a e b são inteiros, dizemos que a é congruente a b módulo m $(m > 0)$ se $m \mid (a - b)$. Denotamos isto por $a \equiv b \pmod{m}$. Se $m \nmid (a - b)$, dizemos que a é incongruente a b módulo m e denotamos $a \not\equiv b \pmod{m}$.

Por exemplo, $11 \equiv 3 \pmod{2}$, pois $2 \mid (11-3)$. Em contrapartida, a diferença $17 - 11 = 6$ não é divisível por 5, isto é, $5 \nmid 6$. Portanto, podemos concluir que $17 \not\equiv 11 \pmod{5}$, pois não existe múltiplo inteiro de 5 que seja igual a 6.

Proposição 3.1 Se a e b são inteiros, temos que $a \equiv b \pmod{m}$ se, e somente se, existir um inteiro k tal que $a = b + km$.

Demonstração: *Se $a \equiv b \pmod{m}$, então $m \mid (a - b)$ o que implica na existência de um inteiro k tal que $a - b = km$, isto é, $a = b + km$. A recíproca é trivial, pois da existência de um k satisfazendo $a = b + km$, temos $km = a - b$, ou seja, que $m \mid (a - b)$ isto é, $a \equiv b \pmod{m}$.* □

Por exemplo, considere $a = 23$, $b = 8$ e $m = 5$. Como $23 - 8 = 15$ e 15 é divisível por 5, temos que $23 \equiv 8 \pmod{5}$. Além disso, existe um inteiro $k = 3$ tal que $23 = 8 + 3 \times 5$, confirmando a proposição de que a congruência ocorre se, e somente se, existir um inteiro k tal que $a = b + km$.

A partir da própria definição, conclui-se de forma imediata que a congruência módulo de um inteiro fixado m satisfaz as três propriedades fundamentais de uma relação de equivalência sobre os inteiros. Esse resultado fundamental será enunciado formalmente a seguir.

Proposição 3.2. Se a, b, c e m são inteiros, $m > 0$, as seguintes sentenças são verdadeiras:

i. Reflexiva: $a \equiv a \pmod{m}$;
ii. Simétrica: se $a \equiv b \pmod{m}$, então $b \equiv a \pmod{m}$;
iii. Transitiva: Se $a \equiv b \pmod{m}$ e $b \equiv c \pmod{m}$, então $a \equiv c \pmod{m}$.

Demonstração:

(i) *Como $m \mid 0$, então $m \mid (a - a)$, o que implica $a \equiv a \pmod{m}$.*

(ii) *Se* $a \equiv b \pmod m$, *então* $a = b + k_1 m$ *para algum inteiro* k_1. *Logo* $b = a - k_1 m$, *o que implica, pela Proposição 3.1,* $b \equiv a \pmod m$.

(iii) *Se* $a \equiv b \pmod m$ *e* $b \equiv c \pmod m$, *então existem inteiros* k_1 *e* k_2 *tais que* $a - b = k_1 m$ *e* $b - c = k_2 m$. *Somando-se, membro a membro estas últimas equações, obtemos* $a - c = (k_1 + k_2)m$, *o que implica* $a \equiv c \pmod m$. □

Esta proposição nos diz que a relação de congruência, definida no conjunto dos inteiros, é uma relação de equivalência, pois acabamos de provar que ela é reflexiva, simétrica e transitiva. Por exemplo, seja $a = 12$ e $m = 7$. Tem-se que $12 \equiv 12 \pmod 7$, pois a diferença entre os dois números é zero e $7 \mid 0$. Para a segunda sentença, considere $a = 15$, $b = 1$ e $m = 7$. Observa-se que $15 - 1 = 14$, e como 14 é múltiplo de 7, segue que $15 \equiv 1 \pmod 7$. Pela propriedade simétrica da congruência, conclui-se também que $1 \equiv 15 \pmod 7$. Em seguida, sejam $a = 20$, $b = 6$, $c = 13$ e $m = 7$. Note que $20 - 6 = 14$, e como 14 é múltiplo de 7, temos $20 \equiv 6 \pmod 7$. Da mesma forma, $13 - 6 = 7$, e, sendo 7 múltiplo de 7, obtemos $6 \equiv 13 \pmod 7$.

Teorema 3.1 Se a, b, c e m são inteiros tais que $a \equiv b \pmod m$, então

i. $a + c \equiv b + c \pmod m$
ii. $a - c \equiv b - c \pmod m$
iii. $ac \equiv bc \pmod m$

Demonstração:

(i) *Como* $a \equiv b \pmod m$, *temos que* $a - b = km$ *e, portanto, como* $a - b = (a + c) - (b + c)$ *temos* $a + c \equiv b + c \pmod m$.

(ii) *Como* $(a - c) - (b - c) = a - b$ *e, por hipótese,* $a - b = km$ *temos que* $a - c \equiv b - c \pmod m$.

(iii) *Como* $a - b = km$ *então* $ac - bc = ckm$ *o que implica* $m \mid (ac - bc)$ *e, portanto,* $ac \equiv bc \pmod m$. □

Considerando os valores $a = 14$, $b = 2$, $c = 3$ e $m = 12$, observa-se que $14 \equiv 2 \pmod{12}$, pois ambos deixam resto 2 ao serem divididos por 12. Somando $c = 3$ aos dois lados, tem-se: $14 + 3 = 17$ e $2 + 3 = 5$. De fato, $17 \equiv 5 \pmod{12}$, pois ambos deixam resto 5 na divisão por 12. De forma análoga, ao subtrair, utilizando os mesmos valores, subtraindo $c = 3$, temos que $14 - 3 = 11$ e $2 - 3 = -1$. Tanto 11 quanto -1 deixam resto 11 quando divididos por 12, então $11 \equiv -1 \pmod{12}$. Por fim,

multiplicando ambos os lados por $c = 3$, temos que $14 \times 3 = 42$ e $2 \times 3 = 6$. Ambos deixam resto 6 quando divididos por 12, logo $42 \equiv 6 \pmod{12}$.

O Teorema 3.1 é um caso particular do Teorema 3.2, basta tomar $d = c$.

Teorema 3.2 Se a, b, c, d e m são inteiros tais que $a \equiv b \pmod{m}$ e $c \equiv d \pmod{m}$, então

i. $a + c \equiv b + d \pmod{m}$
ii. $a - c \equiv b - d \pmod{m}$
iii. $ac \equiv bd \pmod{m}$

Demonstração:

(i) *De* $a \equiv b \pmod{m}$ *e* $c \equiv d \pmod{m}$ *temos* $a - b = km$ *e* $c - d = k_1 m$. *Somando-se membro a membro obtemos* $(a + c) - (b + d) = (k + k_1)m$ *e isto implica* $a + c \equiv b + d \pmod{m}$.

(ii) *Basta subtrair membro a membro* $a - b = km$ *e* $c - d = k_1 m$ *obtendo* $(a - b) - (c - d) = (a - c) - (b - d) = (k - k_1)m$ *o que implica* $a - c \equiv b - d \pmod{m}$.

(iii) *Multiplicamos ambos os lados de* $a - b = km$ *por* c *e ambos os lados de* $c - d = k_1 m$ *por* b, *obtendo* $ac - bc = ckm$ *e* $bc - bd = bk_1 m$. *Basta, agora, somarmos membro a membro estas últimas igualdades obtendo* $ac - bc + bc - bd = ac - bd = (ck + bk_1)m$ *o que implica* $ac \equiv bd \pmod{m}$. □

Para ilustrar, apresento alguns exemplos considerando $a = 17$, $b = 5$, $c = 14$, $d = 2$ e $m = 12$. Como $17 \equiv 5 \pmod{12}$ e $14 \equiv 2 \pmod{12}$, pode-se somar os respectivos termos: $17 + 14 = 31$ e $5 + 2 = 7$. Ambos os resultados deixam resto 7 na divisão por 12, logo $31 \equiv 7 \pmod{12}$, confirmando que $a + c \equiv b + d \pmod{m}$. Na subtração, utilizando os mesmos valores, temos $17 - 14 = 3$ e $5 - 2 = 3$. Como ambos os resultados são iguais, verifica-se que $3 \equiv 3 \pmod{12}$, ou seja, $a - c \equiv b - d \pmod{m}$. Por fim, na multiplicação, multiplica-se a por c e b por d de modo que $17 \times 14 = 238$ e $5 \times 2 = 10$. Tanto 238 quanto 10 deixam resto 10 na divisão por 12, portanto $238 \equiv 10 \pmod{12}$, evidenciando que $ac \equiv bd \pmod{m}$.

Teorema 3.3 Se a, b, c e m são inteiros e $ac \equiv bc \pmod{m}$, então $a \equiv b \pmod{m/d}$ onde $d = (c, m)$.

Demonstração: *De $ac \equiv bc \pmod m$ temos $ac - bc = c(a - b) = km$. Se dividirmos os dois membros por d, teremos $(c/d)(a - b) = k(m/d)$. Logo $(m/d) \mid (c/d)(a - b)$ e, como $(m/d, c/d) = 1$, pelo Teorema 2.6, $(m/d) \mid (a - b)$ o que implica $a \equiv b \pmod{m/d}$.* □

Por exemplo, se $a = 5$, $b = 2$, $c = 6$ e $m = 18$. De $ac \equiv bc \pmod m$ tem-se $5 \cdot 6 \equiv 2 \cdot 6 \pmod{18}$, isto é, $30 \equiv 12 \pmod{18}$. Logo, $ac - bc = c(a - b) = 6\ (5 - 2) = 18 = k \cdot m$, o que implica que com $k = 1$. Calcula-se $d = (c, m) = (6, 18) = 6$. Dividindo-se por d, obtém-se $(\frac{c}{d})(a - b) = (\frac{6}{6}) \cdot 3 = 3 = k \cdot (\frac{m}{d})$, com $m/d = 3$. Assim, $(m/d) \mid (a - b)$, isto é, $3 \mid (5 - 2)$, concluindo-se que $a \equiv b \pmod{m/d}$, ou seja, $5 \equiv 2 \pmod 3$.

Definição 3.2 Se h e k são dois inteiros e $h \equiv k \pmod m$, dizemos que k é um resíduo de h módulo m.

Definição 3.3 O conjunto dos inteiros $\{r_1, r_2, \ldots, r_s\}$ é chamado de sistema completo de resíduos módulo m se:

i. $r_i \not\equiv r_j \pmod m$ para $i \neq j$;
ii. para todo inteiro n, existe um r_i tal que $n \equiv r_i \pmod m$.

Observação. Um sistema completo de resíduos módulo m contém exatamente m elementos.

Como exemplos de sistemas completos de resíduos módulo m, pode-se citar: o conjunto $\{0, 1, 2, 3, \ldots, m - 1\}$ que representa um sistema completo de resíduos módulo m; o conjunto $\{17, 4, -13, 18\}$ que forma um sistema completo de resíduos módulo 4, pois $17 \equiv 1 \pmod 4$, $4 \equiv 0 \pmod 4$, $-13 \equiv 3 \pmod 4$ e $18 \equiv 2 \pmod 4$, respectivamente; e, para m ímpar o conjunto a seguir é um sistema completo de resíduos módulo m, $\left\{ -\frac{m-1}{2}, -\frac{m-3}{2}, \ldots, -1, 0, 1, \ldots, \frac{m-3}{2}, \frac{m-1}{2} \right\}$.

Teorema 3.4 Se k inteiros $r_1, r_2, \ldots, r_k$ formam um sistema completo de resíduos módulo m, então $k = m$.

Demonstração: *Primeiramente, demonstramos que os inteiros $t_0, t_1, \ldots, t_{m-1}$, com $t_i = i$, formam, de fato, um sistema completo de resíduos módulo m. Pelo Teorema 2.2, sabemos que, para cada n, existe um único par de inteiros q e s, tal que $n = mq + s$, onde $0 \leq s < m$. Logo $n \equiv s \pmod m$, sendo s um dos t_i. Como $|t_i - t_j| \leq m - 1$, temos que $t_i \not\equiv t_j \pmod m$ para $i \neq j$. Portanto, o conjunto $\{t_0, t_1, \ldots, t_{m-i}\}$ é um sistema completo de resíduos módulo m. Disto concluímos que cada r_i é congruente a exatamente um*

dos t_i, o que nos garante $k \leq m$. Como o conjunto $\{r_1, r_2, \ldots, r_k\}$ forma, por hipótese, um sistema completo de resíduos módulo m, cada t_i é congruente a exatamente um dos r_i e, portanto $m \leq k$. Desta forma, obtemos $k = m$. □

Teorema 3.5 Se $r_1, r_2, \ldots, r_m$ é um sistema completo de resíduos módulo m e a e b são inteiros, tais que $(a, m) = 1$, então o conjunto

$$\{ ar_1 + b, ar_2 + b, \ldots, ar_m + b \},$$

também é um sistema completo de resíduos módulo m.

Demonstração: *Considerando−se o resultado do teorema anterior, será suficiente mostrar que quaisquer dois inteiros do conjunto $\{ ar_1 + b, ar_2 + b, \ldots, ar_m + b \}$, são incongruentes módulo m. Para isto vamos supor que $ar_i + b \equiv ar_j + b \pmod m$. Logo, pelo Teorema 3.1, temos $ar_i \equiv ar_j \pmod m$. Mas, como $(a, m) = 1$, o Teorema 3.3 nos diz que $r_i \equiv r_j \pmod m$. O fato de $r_i \equiv r_j \pmod m$ implica $i = j$, uma vez que, $r_1, r_2, \ldots, r_m$ formam um sistema completo de resíduos módulo m, o que completa a demonstração.*

□

Da mesma forma que temos um sistema completo de resíduos, podemos ter um sistema reduzido de resíduos (SRR).

Definição 3.4 Entende−se o conjunto $\{r_1, r_2, r_3, \ldots, r_i, \ldots, r_j\}$ por sistema reduzido de resíduos módulo m, se ele for um sistema completo de resíduos no qual foram retirados todos os valores r_i tais que $(r_i, m) \neq 1$, para $i = 1, 2, 3, \ldots, s$, com $s \in \mathbb{N}$.

Por exemplo, o conjunto $\{0, 1, 2, 3, 4, 5, 6, 7\}$ é um sistema completo de resíduos módulo 8, portanto $\{1, 3, 5, 7\}$ é um sistema reduzido de resíduos módulo 8. A fim de se obter um sistema reduzido de resíduos de um sistema completo módulo m, basta retirar os elementos do sistema completo que não são relativamente primos com m.

Proposição 3.3 Se a, b, k e m são inteiros, com $k > 0$ e $a \equiv b \pmod m$, então $a^k \equiv b^k \pmod m$.

Demonstração: *Isto segue, imediatamente, da identidade:*

$$a^k - b^k = (a - b)(a^{k-1} + a^{k-2}b + a^{k-3}b^2 + \ldots + ab^{k-2} + b^{k-1}).$$ □

Teorema 3.6 Se $a \equiv b(\mathrm{mod}\ m_1)$, $a \equiv b(\mathrm{mod}\ m_2)$,..., $a \equiv b(\mathrm{mod}\ m_k)$ onde a, b, m_1, m_2, ... , m_k são inteiros, com m_i positivos para todo $i = 1, 2, \dots, k$, então

$$a \equiv b\ (\mathrm{mod}\ [m_1, m_2, \dots, m_k])$$

onde $[m_1, m2, \dots, m_k]$ é o mínimo múltiplo comum de $m_1, m_2, \dots, m_k$.

Demonstração: *Seja p_n o maior primo que aparece nas fatorações de $m_1, m_2, \dots, m_k$. Cada m_i pode ser expresso como*

$$m_i = p_1^{\alpha_{1i}} . p_2^{\alpha_{2i}} \dots p_{1n}^{\alpha_{ni}},$$

para cada $i = 1, 2, \dots, k$ *e alguns α_{ji} podem ser nulos.*

Como $m_i \mid (a - b), i = 1, 2, \dots, k$ temos que $p_j^{\alpha_{ji}} \mid (a - b)$, $i = 1, 2, \dots, k$, $j = 1, 2, \dots, n$. Logo, se tomarmos $\propto_j = \max_{1 \le i \le k} \{\propto_{ji}\}$ teremos que

$$p_1^{\alpha_1} . p_2^{\alpha_2} \dots p_n^{\alpha_n} \mid (a - b).$$

Mas,

$$p_1^{\alpha_1} . p_2^{\alpha_2} \dots p_n^{\alpha_n} = [m_1, m_2, \dots, m_k]$$

o que implica $a \equiv b\ (mod\ [m_1, m_2, \dots, m_k])$. □

Definição 3.5 Chamamos de congruência linear em uma variável toda congruência da forma $ax \equiv b\ (\mathrm{mod}\ m)$, onde x é a incógnita.

É fácil de se verificar que, se x_0 é uma solução, ou seja, $ax_0 \equiv b\ (\mathrm{mod}\ m)$ e se $x_1 \equiv x_0\ (\mathrm{mod}\ m)$, então x_1 também é solução. É óbvio, pois se $x_1 \equiv x_0\ (\mathrm{mod}\ m)$, então $ax_1 \equiv ax_0 \equiv b\ (\mathrm{mod}\ m)$.

Uma equação da forma $ax + by = c$, em que a, b e c são inteiros, recebe o nome de equação diofantina linear, em referência ao matemático grego Diofanto.

Teorema 3.7 Sejam a, b, c inteiros e $d = (a, b)$. Se $d \nmid c$, então a equação $ax + by = c$ não possui nenhuma solução inteira. Se $d \mid c$, ela possui infinitas soluções, e se $x = x_0$ e $y = y_0$ é uma solução particular, então todas as outras soluções são dadas por

$$x = x_0 + (b/d)k$$
$$y = y_0 - (a/d)k,$$

onde k é um inteiro.

Demonstração: *Se $d \nmid c$, então a equação $ax + by = c$, não possui solução pois, como $d \mid a$ e $d \mid b$, d deveria dividir c, o qual é uma combinação linear de a e b. Suponhamos, que $d \mid c$, pelo Teorema 2.3, existem inteiros n_0 e m_0, tais que*

$$an_0 + bm_0 = d.$$

Como $d \mid c$, existe um inteiro k tal que $c = kd$. Se multiplicarmos, ambos os membros da equação, $an_0 + bm_0 = d$, por k, teremos $a(n_0k) + b(m_0k) = kd = c$. Isto nos diz que o par (x_0, y_0) com $x_0 = n_0k$ e $y_0 = m_0k$ é uma solução de $ax + by = c$. É fácil a verificação de que os pares da forma

$$x = x_0 + (b/d)k$$
$$y = y_0 - (a/d)k$$

são soluções, uma vez que

$$\begin{aligned} ax + by &= a(x_0 + (b/d)k) + b(y_0 - (a/d)k) \\ &= ax_0 + \frac{ab}{d}k + by_0 - \frac{ab}{d}k \\ &= ax_0 + by_0 = c. \end{aligned}$$

O que acabamos de mostrar é que, conhecida uma solução particular (x_0, y_0) podemos, a partir dela, gerar infinitas soluções. Precisamos, agora, mostrar que toda solução da equação $ax + by = c$ é da forma $x = x_0 + (b/d)k$ e $y = y_0 - (a/d)k$. Vamos supor que (x, y) seja uma solução, ou seja, $ax + by = c$. Mas, como $ax_0 + by_0 = c$, obtemos, subtraindo membro a membro, temos

$$ax + by - ax_0 - by_0 = a(x - x_0) + b(y - y_0) = 0,$$

o que implica $a(x - x_0) = b(y_0 - y)$. Como $d = (a, b)$ temos, pelo corolário da Proposição 2.4,

$$\left(\frac{a}{d}, \frac{b}{d}\right) = 1.$$

Portanto, dividindo−se os dois membros da última igualdade por d, teremos

$$\frac{a}{d}(x - x_0) = \frac{b}{d}(y_0 - y).$$

Pelo Teorema 2.6, $(b/d) \mid (x - x_0)$, logo existe um inteiro k tal que $x - x_0 = k\,(b/d)$, ou seja $x = x_0 + (b/d)k$. Substituindo−se este valor de x na equação, $\frac{a}{d}(x - x_0) = \frac{b}{d}(y_0 - y)$, temos $y = y_0 - (a/d)k$, o que conclui a demonstração. □

Através deste teorema, podemos dizer quantas são as soluções incongruentes (caso exista alguma) que a congruência linear $ax \equiv b \pmod{m}$ possui.

Teorema 3.8 Sejam a, b e m inteiros tais que $m > 0$ e $(a, m) = d$. No caso em que $d \nmid b$ a congruência $ax \equiv b \pmod{m}$ não possui nenhuma solução e quando $d \mid b$, possui exatamente d soluções incongruentes módulo m.

Demonstração: *Pela Proposição 3.1 sabemos que o inteiro x é solução de $ax \equiv b \pmod{m}$ se, e somente se, existe um inteiro y tal que $ax = b + my$, ou, o que é equivalente, $ax - my = b$. Do teorema anterior sabemos que esta equação não possui nenhuma solução caso $d \nmid b$, e que se $d \mid b$ ela possui infinitas soluções dadas por $x = x_0 - (m/d)k$ e $y = y_0 - (a/d)k$ onde (x_0, y_0) é uma solução particular de $ax - my = b$. Logo a congruência $ax \equiv b \pmod{m}$ possui infinitas soluções dadas por $x = x_0 - \left(\frac{m}{d}\right)k$. Como estamos interessados em saber o número de soluções incongruentes, vamos tentar descobrir sob que condições $x_1 = x_0 - (m/d)k_1$ e $x_2 = x_0 - (m/d)k_2$ são congruentes módulo m. Se x_1 e x_2 são congruentes então $x_0 - (m/d)k1 \equiv x_0 - (m/d)k_2 \pmod{m}$. Isto implica $(m/d)k_1 \equiv (m/d)k_2 \pmod{m}$, e como $(m/d) \mid m$, temos $(m/d, m) = m/d$, o que nos permite o cancelamento de m/d resultando, pelo Teorema 3.3, $k_1 \equiv k_2 \pmod{d}$. Observe que m foi substituído por $d = m/(m/d)$. Isto nos mostra que soluções incongruentes serão obtidas ao tomarmos $x = x_0 - (m/d)k$, onde k percorre um sistema completo de resíduos módulo d, o que conclui a demonstração.* □

Definição 3.5 Dizemos que uma solução x_0 de $ax \equiv b \pmod{m}$ é única módulo m, quando qualquer outra solução x_1 for congruente a x_0 módulo m.

Definição 3.6 Uma solução a' de $ax \equiv 1 \pmod{m}$, é chamada de um inverso de a módulo m.

Segue, do Teorema 3.8, que se $(a, m) = 1$, então a possui um único inverso módulo m. A proposição seguinte nos diz quando um inteiro a é o seu próprio inverso módulo p, onde p é um número primo.

Proposição 3.4 Seja p um número primo. Um inteiro positivo a é o seu próprio inverso módulo p se, e somente se, $a \equiv 1 \pmod{p}$ ou $a \equiv -1 \pmod{p}$.

Demonstração: *Se a é o seu próprio inverso, então $a^2 \equiv 1 \pmod{p}$, o que significa que $p \mid (a^2 - 1)$. Mas se $p \mid (a-1)(a+1)$, sendo p primo, $p \mid (a-1)$ ou $p \mid (a+1)$, o que implica $a \equiv 1 \pmod{p}$ ou $a \equiv -1 \pmod{p}$. A recíproca é imediata pois, se $a \equiv 1 \pmod{p}$ ou $a \equiv -1 \pmod{p}$, então $p \mid (a-1)$ ou $p \mid (a+1)$. Portanto $p \mid (a-1)(a+1)$ o que significa $a^2 \equiv 1 \pmod{p}$, o que conclui a demonstração.* □

Definição 3.7 Dado um número inteiro $m > 1$ e um inteiro a, a classe de congruência ou classe de resíduos de a módulo m é definida como:

$$\bar{a} = \{x \in \mathbb{Z} \mid x \equiv a \pmod{m}\}.$$

A relação de congruência módulo m é, por definição, uma relação de equivalência em $\mathbb{Z}$. Por uma propriedade geral de relações de equivalência, o conjunto das classes de equivalência formadas por essa relação define uma partição do conjunto $\mathbb{Z}$. Cada classe de congruência $\bar{a}$ contém todos os inteiros que são congruentes da operação a módulo m, e não há interseção entre classes diferentes.

Isso significa que $\bar{a}$ contém todos os inteiros que deixam o mesmo resto que a na divisão por m. As classes de congruência módulo m particionam o conjunto dos inteiros $\mathbb{Z}$, de modo que cada inteiro pertence exatamente a uma única classe, quaisquer duas classes são mutuamente disjuntas ou coincidem, e a união de todas elas resulta no próprio conjunto $\mathbb{Z}$. Por exemplo, seja $m = 5$. O conjunto $\mathbb{Z}$ é particionado em cinco classes de congruência distintas:

$$\bar{0} = \{\ldots, -10, -5, 0, 5, 10, \ldots\}$$
$$\bar{1} = \{\ldots, -9, -4, 1, 6, 11, \ldots\}$$
$$\bar{2} = \{\ldots, -8, -3, 2, 7, 12, \ldots\}$$
$$\bar{3} = \{\ldots, -7, -2, 3, 8, 13, \ldots\}$$
$$\bar{4} = \{\ldots, -6, -1, 4, 9, 14, \ldots\}.$$

As classes de congruência módulo m são essenciais para definir o conjunto quociente $\frac{\mathbb{Z}}{m\mathbb{Z}}$ também denotado por $\mathbb{Z}_m$. Esse conjunto contém exatamente m classes distintas, representadas por $\{\bar{0}, \bar{1}, \ldots, \overline{m-1}\}$. Esse sistema é a base da aritmética modular e aparece com frequência em contextos criptográficos, como no cálculo de chaves públicas e privadas no algoritmo RSA.

3.2 Teoremas Fundamentais e Suas Implicações

A aritmética modular é fundamentada em algumas propriedades importantes que permitem manipular congruências de forma eficiente. Para compreender melhor essas propriedades, vamos analisá−las através de exemplos práticos e simples, ideais para estudantes do Ensino Fundamental. O desenvolvimento da teoria das congruências proporcionou importantes teoremas, que têm grande aplicação na criptografia moderna e segurança digital.

3.2.1 Pequeno Teorema de Fermat

Desde a Antiguidade, já se conhecia o fato de que, se p é primo, então $p \mid (2^p - 2)$. A generalização desse resultado foi apresentada por Pierre de Fermat (1601–1665) por meio de um teorema breve, porém elaborado. Nascido na França, Fermat realizou importantes descobertas na Matemática, embora não fosse matemático de profissão, tendo atuado como advogado e magistrado. Somente a partir da vida adulta passou a dedicar−se ao estudo matemático em seu tempo livre. Conhecido como “o príncipe dos amadores”, contribuiu para o desenvolvimento da Geometria Analítica, das bases do Cálculo Diferencial e Integral e da Probabilidade, sendo, entretanto, na Teoria dos Números que concentrou seu maior interesse. Entre suas contribuições mais relevantes, destaca−se o Pequeno Teorema de Fermat**.**

Teorema 3.9 Sejam $a, p \in \mathbb{N}$ tal que p é primo. Se $p \nmid a$, então, $a^{p-1} \equiv 1 \pmod{p}$.

Demonstração: *Sabemos que o conjunto formado pelos p números $\{0,1, 2, \ldots, p-1\}$ constitui um sistema completo de resíduos módulo p. Isto significa que qualquer conjunto contendo no máximo p elementos incongruentes módulo p pode ser colocado em correspondência biunívoca com um subconjunto de $\{0,1, 2,\ldots, p-1\}$. Vamos, agora,*

considerar os números $a, 2a, 3a, \dots, (p-1)a$. Como $(a, p) = 1$, nenhum destes números ia, com $1 \leq i \leq p-1$ é divisível por p, ou seja, nenhum é congruente a zero módulo p.

Quaisquer dois deles são incongruentes módulo p, pois se $aj \equiv ak (mod\ p)$ implica $j \equiv k\ (mod\ p)$ e isto só é possível se $j = k$, uma vez que ambos j e k são positivos e menores do que p. Temos, portanto, um conjunto de $p-1$ elementos incongruentes módulo p e não divisíveis por p. Logo, cada um deles é congruente a exatamente um dos elementos do conjunto $\{1, 2, 3, \dots, p-1\}$. Se multiplicarmos estas congruências, membro a membro, obtemos:

$$a\ (2a)\ (3a) \dots (p-1)a \equiv 1.\ 2.\ 3.. (p-1)\ (mod\ p),$$

ou seja, $a^{p-1}(p-1)! \equiv (p-1)!(mod\ p)$. Mas, como $((p-1)!, p) = 1$, podemos cancelar o fator $(p-1)!$ em ambos os lados, obtendo

$$a^{p-1} \equiv 1(mod\ p),$$

o que conclui a demonstração. □

Por exemplo, seja $p = 7$, um número primo, e $a = 3$, tal que $\text{mdc}\ (3, 7) = 1$. Aplicando o Pequeno Teorema de Fermat, temos $3^{7-1} = 3^6 = 729 \equiv 1\ (\text{mod}\ 7)$. De fato, 729 deixa resto 1 quando dividido por 7.

Corolário 3.1 Se p é um primo e a é um inteiro positivo, então $a^p \equiv a\ (\text{mod}\ p)$.

Demonstração: *Temos que analisar dois casos: se $p \mid a$ e se $p \nmid a$.*

(i) Se $p \mid a$, então $p \mid (a(a^{p-1} - 1))$ e, portanto $a^p \equiv a\ (mod\ p)$.

(ii) Se $p \nmid a$, pelo Pequeno Teorema de Fermat $p \mid (a^{p-1} - 1)$ e portanto, $p \mid (a^p - a)$.

Logo, em ambos os casos, $a^p \equiv a\ (mod\ p)$. □

Considere os seguintes exemplos. Para $p = 5$ e $a = 10$, observa-se que $5 \mid 10$, de modo que $a^p \equiv a\ (\text{mod}\ p)$, ou seja, $10^5 \equiv 10\ (\text{mod}\ 5)$. De fato, $10^5 \equiv 0\ (\text{mod}\ 5)$ e $10 \equiv 0\ (\text{mod}\ 5)$, logo $10^5 \equiv 10\ (\text{mod}\ 5)$. Já para $p = 5$ e $a = 7$, como $5 \nmid 7$, tem-se $7 \equiv 2\ (\text{mod}\ 5)$. Assim, $7^5 \equiv 2^5 = 32 \equiv 2(\text{mod}\ 5)$, e, como $7 \equiv 2\ (\text{mod}\ 5)$, conclui-se que $7^5 \equiv 7\ (\text{mod}\ 5)$.

Descoberto por Pierre de Fermat em 1640, o Pequeno Teorema de Fermat é amplamente utilizado na criptografia, servindo como ferramenta eficiente para

reduzir potências módulo p e fundamentando testes rápidos de primalidade. Aplica−se em algoritmos como o RSA, auxiliando na redução de expoentes, e constitui a base para o Teorema de Euler, que o generaliza para inteiros não primos.

3.2.2 Teorema de Euler

Proposição 3.6 Sejam $a, m \in \mathbb{N}$, com $m > 1$. A congruência

$$ax \equiv 1 \pmod{m}$$

possui uma solução x_0 se, e somente se, $mdc(a,m) = 1$. Além disso, x é uma solução da congruência se, e somente se,

$$x \equiv x_0 \pmod{m}.$$

Demonstração: *A congruência acima tem uma solução x_0 se, e somente se, $m \mid ax_0 - 1$, o que equivale a dizer que a equação diofantina $aX - mY = 1$ possui solução em números inteiros. Em virtude do Teorema 3.7, isto ocorre se, e somente se, $(a, m) = 1$. Por outro lado, observe que, se x_0 e x são soluções da congruência $aX \equiv 1 \pmod{m}$, então $ax \equiv ax_0 \pmod{m}$, o que implica, em virtude do Teorema 3.3, que $x \equiv x_0 \pmod{m}$. Observe, ainda, que se x_0 é solução da congruência $aX \equiv 1 \pmod{m}$, e $x \equiv x_0 \pmod{m}$, então x é também solução da mesma congruência, pois*

$$ax \equiv ax_0 \equiv 1 \pmod{m}.$$ □

Utilizaremos a notação $\varphi(m)$ para denotar a quantidade de elementos de um sistema reduzido de resíduos módulo m, tal que $m > 1$. Nesse caso, de acordo com a Definição 3.4, será um sistema reduzido de resíduos formado apenas por todos os números naturais entre 0 e $m - 1$ que são primos com m. Se colocarmos $\varphi(1) = 1$, podemos definir a função

$$\varphi : \mathbb{N} \to \mathbb{N},$$

que chamaremos de phi de Euler. Pela definição de sistema reduzido de resíduos, temos que

$$\varphi(m) \leq m - 1, \text{ para todo } n \geq 2.$$

Além disso, se $m \geq 2$, então $\varphi(m) = m - 1$ se, e somente se, m é um número primo. Podemos ter $\varphi(m) = m - 1$ porque se m for um primo, então para todo r que seja um resíduo módulo m teremos que $(m, r) = 1$, ou seja, todos os restos possíveis serão contabilizados no sistema reduzido de resíduos. E podemos ter $\varphi(m) < m-1$, porque se m não for primo, teremos pelo menos um r tal que $(m, r) \neq 1$, o que faz com que o número de elementos do SRR seja menor que $m - 1$.

Por exemplo, para 12, que é um número composto, temos $\varphi(12) = 4$, obtido a partir do sistema reduzido de resíduos $\{1, 5, 7, 11\}$, o que confirma que, nesse caso, $\varphi(m) < m - 1$. Já para o número primo 5, vale $\varphi(5) = 4$, conforme a propriedade $\varphi(m) = m - 1$, pois todos os inteiros positivos menores que 5 são coprimos com ele, formando o conjunto $\{1, 2, 3, 4\}$.

A função phi de Euler é muito importante para a Teoria dos Números, portanto seguem alguns resultados que se utilizam dela.

Proposição 3.7 Se $\{r_1, r_2, \ldots, r_{\varphi(m)}\}$ for um sistema reduzido de resíduos módulo m e $(a, m) = 1$ para todo $a \in Z$, então $\{ar_1, ar_2, \ldots, ar_{\varphi(m)}\}$ também é um sistema reduzido de resíduos.

Demonstração. *Seja $\{r_1, r_2, \ldots, r_{\varphi(m)}\}$ um sistema reduzido de resíduos módulo m e suponha que $\{a_1, a_2, \ldots, a_m\}$ também seja um outro sistema reduzido de resíduos módulo m retirado do sistema anterior apresentado. Como $(a, m) = 1$ e $(r_i, m) = 1$, temos que $(ar_i, m) = 1$. Isto nos diz que se tomarmos $ar_i \equiv ar_j \pmod m$, teremos, pelo Teorema 3.3, $r_i \equiv r_j \pmod m$, portanto $i = j$, o que conclui a nossa demonstração, já que $\{r_1, r_2, \ldots, r_{\varphi(m)}\}$ é um sistema reduzido de resíduos.* □

Teorema 3.10 (Teorema de Euler). Sejam $a, m \in \mathbb{Z}$, com $m > 1$ e $\text{mdc}(a, m) = 1$. Então:

$$a^{\varphi(m)} \equiv 1 \pmod m$$

Demonstração. *Considerando* $\{r_1, r_2, \ldots, r_m\}$ *um sistema reduzido de resíduos módulo* m*, então, pela Proposição 3.7, temos que* $\{ar_1, ar_2, \ldots, ar_{\varphi(m)}\}$ *também é um sistema reduzido de resíduos módulo* m*. Dessa forma,*

$$ar_1 \cdot ar_2 \cdot \ldots \cdot ar_{\varphi(m)} \equiv r_1 \cdot r_2 \cdot \ldots \cdot r_{\varphi(m)} \pmod{m}.$$

Em virtude disso,

$$ar_1 \cdot ar_2 \cdot \ldots \cdot ar_{\varphi(m)} \equiv a^{\varphi(m)} r_1 \cdot r_2 \cdot \ldots \cdot r_{\varphi(m)} \equiv r_1 \cdot r_2 \cdot \ldots \cdot r_{\varphi(m)} \pmod{m}.$$

E como $(r_1 \cdot r_2 \cdot \ldots \cdot r_{\varphi(m)}, m) = 1$*, então,* $a^{\varphi(m)} \equiv 1 \pmod{m}$*.* □

Por exemplo, seja $a = 3$ e $m = 10$. Como $\text{mdc}(3, 10) = 1$, e $\varphi(10) = 4$, então $3^4 = 81 \equiv 1 \pmod{10}$.

Observação. Para relacionar o Teorema de Euler com o Pequeno Teorema de Fermat, basta tomar $\varphi(p) = p - 1$, com p primo, ficando, assim, com $a^{p-1} \equiv 1 \bmod p$.

O Teorema de Euler é uma generalização do Pequeno Teorema de Fermat e desempenha papel central na criptografia moderna, especialmente no algoritmo RSA.

3.3 Introdução a Algoritmos Criptográficos

De acordo com Menezes, Vanstone & Van Oorschot, (1996) a criptografia pode ser classificada em dois grandes grupos: simétrica (mesma chave para codificar e decodificar) e assimétrica (chaves distintas para codificação e decodificação). O algoritmo RSA (Rivest−Shamir−Adleman), criado em 1978 por Rivest, Shamir e Adleman, é um dos algoritmos de criptografia assimétrica mais relevantes, baseado na dificuldade de fatoração de números inteiros grandes formados pelo produto de dois números primos distintos. A seguir, apresentam-se as principais etapas do funcionamento do RSA, cujo exemplo detalhado será desenvolvido no Capítulo 4.

I. Escolher dois números primos grandes p e q.
II. Calcular $n = p \cdot q$ e $\varphi(n) = (p-1)(q-1)$.
III. Escolher uma chave pública e, tal que $\text{mdc}(e, \varphi(n)) = 1$.
IV. Calcular a chave privada d, tal que $e \cdot d \equiv 1 (\text{mod } \varphi(n))$.

V. Para criptografar uma mensagem M, calcular $C \equiv M^e (\mathrm{mod}\ n)$.

VI. Para decodificar, calcular $M \equiv C^d (\mathrm{mod}\ n)$.

A aplicação prática do algoritmo RSA tem início com a conversão da mensagem em dados numéricos, etapa essencial para que o conteúdo textual possa ser tratado de forma matemática. Para isso, cada caractere da mensagem é associado ao seu respectivo valor na Tabela ASCII (*American Standard Code for Information Interchange*), sistema amplamente utilizado na computação para representar e trocar informações entre diferentes dispositivos. O padrão ASCII original define 128 caracteres, numerados de 0 a 127, que incluem letras, números, sinais de pontuação e caracteres de controle, estes últimos, compreendidos entre 0 e 31, correspondem a comandos que indicam ações a serem executadas por dispositivos de saída, como impressoras e monitores. Entre eles, destacam-se o NUL (caractere nulo) e o SOH (início de cabeçalho de transmissão).

Tabela 4. Alguns caracteres na tabela ASCII − Representação Decimal

Caractere	ASCII Decimal	Binário
NUL	00	0000 0000
SOH	01	0000 0001
STX	02	0000 0010
A	65	0100 0001
B	66	0100 0010
C	67	0100 0011
a	97	0110 0001
b	98	0110 0010
c	99	0110 0011
0	48	0011 0000
1	49	0011 0001
2	50	0011 0010
Espaço	32	0010 0000
!	33	0010 0001
@	64	0100 0000
#	35	0010 0011
$	36	0010 0100

Fonte: Adaptado da Tabela ASCII padrão (ANSI X3.4−1986)

Além da versão padrão, existem extensões da tabela ASCII que acrescentam novos códigos, variando de 128 a 255 (ou, em representações assinadas, de −1 a −128), a fim de incluir caracteres acentuados e símbolos utilizados em outros idiomas. Essas variações ampliam a capacidade de codificação e tornam o sistema compatível com diferentes contextos linguísticos e tecnológicos.

Com base nesse sistema de representação, torna-se possível compreender como os algoritmos criptográficos, como o RSA, operam sobre dados textuais transformando caracteres em números e vice-versa. Essa correspondência entre linguagem e número pode ser explorada em sala de aula para introduzir conceitos matemáticos de maneira concreta e envolvente. Atividades que envolvem a codificação e decodificação de mensagens favorecem a compreensão de ideias como modularidade, inverso multiplicativo e exponenciação, além de estimularem o raciocínio lógico e o interesse dos estudantes.

Nesse sentido, Borba e Penteado (2016) destacam que jogos e desafios envolvendo códigos contribuem para o engajamento dos alunos e contextualizam conteúdos matemáticos abstratos em situações práticas e significativas. De forma alinhada, a Base Nacional Comum Curricular (Brasil, 2018) reforça que práticas desse tipo fortalecem competências essenciais, como o raciocínio lógico, a argumentação matemática e a resolução de problemas.

3.4 Síntese dos Conceitos Matemáticos Aplicados à Criptografia

Os conteúdos abordados no Capítulo 3 constituem a espinha dorsal dos sistemas criptográficos clássicos e modernos. A aritmética modular, as congruências, os teoremas fundamentais e os algoritmos de cálculo de inverso formam o núcleo matemático sobre o qual se estruturam diversos métodos de codificação e decodificação de mensagens. Esses elementos, além de sustentarem as cifras estudadas, são essenciais para compreender os mecanismos que garantem segurança, confidencialidade e integridade da informação.

Ao explorar definições como a de congruência modular e de inteiro inversível, bem como propriedades como o cancelamento em congruências e a compatibilidade das operações, o estudante adquire instrumentos para analisar o funcionamento interno de cifras como a de César, Afim, Vigenère e, mais adiante, dos

algoritmos RSA e AES. A matemática deixa, assim, de ocupar um lugar meramente abstrato, assumindo um papel instrumental na resolução de problemas reais e contemporâneos.

Além disso, a apresentação dos teoremas de Fermat e Euler permite fundamentar a lógica de algoritmos assimétricos e otimizar cálculos modulares em aplicações computacionais. Esses resultados teóricos serão aplicados no Capítulo 4 para explicar algoritmos de segurança digital, e no Capítulo 5 como base matemática das atividades práticas desenvolvidas em linguagem de programação. Dessa forma, a síntese dos conceitos aqui apresentados evidencia que o domínio da teoria dos números, da lógica e da álgebra modular é indispensável para compreender, desenvolver e aplicar técnicas criptográficas de maneira crítica e criativa no ambiente educacional.

CAPÍTULO 4 – TECNOLOGIA DA INFORMAÇÃO E CRIPTOGRAFIA

4.1 Introdução aos Sistemas de Informação e Segurança

Os sistemas de informação desempenham um papel central nas organizações contemporâneas, servindo como suporte às operações rotineiras, ao planejamento estratégico e à tomada de decisões. A crescente digitalização de processos, no entanto, aumenta significativamente a exposição de dados sensíveis a diversos tipos de ameaças e ataques cibernéticos. Nesse contexto, a implementação de modelos de segurança torna−se imprescindível para preservar a integridade dos ativos informacionais.

Um dos modelos mais amplamente adotados para estruturar a segurança da informação é o modelo CIA, que abrange os princípios de Confidencialidade, Integridade e Disponibilidade. De acordo com Stallings (2006, p. 9);

> Confidencialidade refere−se à proteção das informações contra acessos não autorizados; integridade implica garantir a precisão e completude das informações e dos métodos de processamento; já disponibilidade assegura que as informações estejam acessíveis quando necessário.

Esses três pilares formam a base das políticas e práticas de proteção de dados em ambientes computacionais constituem o núcleo conceitual em torno do qual todas as práticas de segurança da informação são desenvolvidas. Sem atender a pelo menos um deles, um sistema pode ser considerado vulnerável.

A relevância do modelo CIA se evidencia diante dos inúmeros ataques cibernéticos registrados nos últimos anos. Um caso notório é o *WannaCry*, que, em 2017, afetou mais de 200 mil computadores em cerca de 150 países, criptografando arquivos dos usuários e exigindo pagamento em criptomoedas para liberação dos dados. O ataque comprometeu a disponibilidade e a confidencialidade das informações, afetando sistemas hospitalares, bancos, empresas públicas e privadas.

Outro tipo recorrente de ameaça é o ataque *Distributed Denial of Service* (DDoS), negação de serviço distribuída, que sobrecarrega servidores com tráfego artificial, tornando sistemas inacessíveis e comprometendo diretamente sua disponibilidade. Conforme Tanenbaum (2011), esse tipo de ofensiva inunda os

servidores com tantas requisições que os serviços legítimos não conseguem ser atendidos.

Além disso, os ataques de *phishing* constituem uma ameaça significativa à confidencialidade da informação, sendo amplamente utilizados em ações de engenharia social. Esse tipo de ataque consiste em induzir o usuário ao erro por meio de mensagens fraudulentas que simulam comunicações legítimas, como as de instituições bancárias, órgãos públicos ou serviços digitais. O objetivo é persuadir a vítima a revelar dados sensíveis, como senhas, informações bancárias, documentos pessoais ou credenciais de acesso, que posteriormente podem ser utilizados em fraudes, invasões ou roubos de identidade.

Diante desses riscos, é fundamental que as organizações adotem estratégias de segurança da informação robustas, estruturadas com base no modelo CIA e orientadas por princípios matemáticos e computacionais sólidos. A aplicação desse modelo contribui não apenas para a mitigação de vulnerabilidades, mas também para o fortalecimento da governança digital e da confiança dos usuários nos sistemas tecnológicos.

4.2 Aplicação Tecnológica dos Métodos Criptográficos

Nesse cenário matematicamente estruturado, destacam−se três categorias principais de criptografia: a criptografia simétrica, a criptografia assimétrica e as funções *hash*, que geram resumos criptográficos únicos e irreversíveis das mensagens, amplamente utilizadas para garantir a integridade dos dados (Stallings, 2006). Neste trabalho, será apresentado o estudo da criptografia assimétrica RSA, dada sua relevância teórica e aplicabilidade pedagógica. Já os algoritmos simétricos, como o AES, e as funções *hash*, como o SHA−256, serão apenas mencionados brevemente na próxima seção, sem detalhamento técnico. Cada uma dessas abordagens criptográficas se fundamenta em princípios matemáticos específicos, oferecendo diferentes níveis de segurança e desempenho, conforme as necessidades e requisitos das aplicações práticas.

4.2.1 Criptografia Assimétrica (RSA)

A criptografia assimétrica é uma abordagem criptográfica avançada que utiliza um par de chaves distintas, uma pública e uma privada, proporcionando maior segurança nas comunicações digitais. O algoritmo RSA, criado por Rivest, Shamir e Adleman, destaca−se como uma das implementações mais robustas dessa técnica, baseado profundamente em problemas matemáticos de difícil solução computacional, especialmente a fatoração de números inteiros grandes e operações de aritmética modular. Segundo Stallings (2006), o algoritmo RSA fundamenta−se matematicamente no Teorema de Euler, que afirma que, se dois números são primos entre si, isto é, se mdc $(a, N) = 1$, então vale a congruência $a^{\varphi(N)} \equiv 1 \pmod{N}$. Em outras palavras, a elevação de a à potência φ(N), conhecida como função totiente de Euler, resulta em 1 no sistema modular de módulo N.

O RSA começa com a escolha de dois números primos grandes, p e q, e calcula−se o produto desses números, representado por $N = p \cdot q$. A segurança do RSA reside na dificuldade extrema de fatorar o número N quando este é suficientemente grande. A função totiente de Euler, $\varphi(N) = (p-1)(q-1)$, é então usada para calcular as chaves criptográficas.

A seguir, apresenta−se um exemplo matemático detalhado das etapas de cifragem e decifragem no RSA, ilustrando o funcionamento do algoritmo de forma didática.

I. Escolha dois números primos pequenos para demonstração: $p = 13$ e $q = 17$.

II. Calcule $N = p \cdot q = 13 \cdot 17 = 221$.

III. Calcule a função totiente $\varphi(N) = (p-1)(q-1) = 12 \cdot 16 = 192$.

IV. Escolha uma chave pública e, tal que $1 < e < \varphi(N)$ e $\text{mdc}\,(e, \varphi(N)) = 1$. Deste modo, tome $e = 5$, pois $1 < 5 < 192$ e $\text{mdc}\,(5, 192) = 1$.

V. Determine a chave privada d, tal que $d \cdot e \equiv 1 \pmod{\varphi(N)}$. Assim, temos que $d = 77$, pois $5 \cdot 77 \equiv 1 \pmod{192}$.

As informações encontram-se detalhadas na tabela a seguir.

Tabela 5. Parâmetros utilizados na geração das chaves RSA

Tipo de chave	**Valor**
Módulo N	221
Totiente $\varphi(N)$	192
Chave pública e	5
Chave privada d	77

Fonte: elaborado pelo autor (2025)

Com o intuito de facilitar a compreensão dos princípios matemáticos que fundamentam o algoritmo RSA, apresenta−se a seguir um exemplo completo, com a demonstração de todas as etapas. A palavra "MARIA" será utilizada como mensagem de referência a ser criptografada e decifrada. Todos os cálculos foram realizados com números pequenos, de modo a manter a clareza da explicação, sem comprometer a lógica envolvida no processo.

A tabela a seguir apresenta a conversão de cada letra da mensagem “MARIA” para sua representação decimal na Tabela ASCII, conforme descrito na Seção 3.3.

Tabela 6. Conversão da mensagem “MARIA” para ASCII

Letra	**Código ASCII**
M	77
A	65
R	82
I	73
A	65

Fonte: Adaptado da Tabela ASCII padrão (ANSI X3.4−1986)

Nesta etapa, utilizou−se a palavra “MARIA” como exemplo de mensagem a ser cifrada por meio do algoritmo RSA. Com a escolha dos primos $p = 13$ e $q = 17$, obteve−se um módulo $N = 221$, valor suficientemente grande para acomodar todos os códigos ASCII das letras da palavra, que variam entre 65 e 82. Assim, tornou−se possível aplicar diretamente a exponenciação modular sobre os valores reais da mensagem, sem a necessidade de adaptações ou simplificações.

No entanto, é importante destacar que em abordagens didáticas com primos menores, como $p = 7$ e $q = 11$, por exemplo, cujo módulo $N = 77$ é inferior a alguns códigos ASCII da palavra, torna−se necessário substituir os blocos originais por valores simulados menores que N. Essa estratégia de adaptação é amplamente utilizada em contextos educacionais para tornar os cálculos viáveis manualmente, sem comprometer a lógica e a estrutura do algoritmo. Um exemplo dessa abordagem está apresentado na Tabela 7:

Tabela 7. Blocos numéricos simulados (valores menores que 77)

Bloco Simulado
7
76
58
27
76

Fonte: elaborado pelo autor (2025)

Esses números não representam diretamente a palavra "MARIA", mas simulam uma situação real para demonstrar como o RSA funciona com N pequeno. A escolha desses valores obedece a dois critérios fundamentais:

I. Todos são estritamente menores que $N = 77$, garantindo que possam ser utilizados em operações de cifragem RSA com módulo 77;

II. Os números foram selecionados de forma a representar uma diversidade de resultados nas operações de cifragem e decifragem, permitindo a observação do comportamento matemático do algoritmo sob diferentes entradas.

Além disso, vale destacar que, em implementações reais do RSA, as mensagens são geralmente divididas em blocos binários ou agrupadas por meio de codificações padronizadas, como UTF−8 ou hexadecimal, sendo então convertidas para inteiros menores que N, muitas vezes com o uso de preenchimento criptográfico. Portanto, a adaptação aqui realizada é não apenas válida, como também reflete a complexidade das transformações envolvidas em aplicações profissionais do algoritmo.

Retomando o exemplo da mensagem “MARIA”, cifrada por meio do algoritmo RSA, utilizando diretamente os valores reais correspondentes aos códigos ASCII das letras, sem necessidade de adaptações. Com o módulo $N = 221$, obtido a partir dos primos $p = 13$ e $q = 17$, é possível acomodar todos os valores da sequência pois todos são inferiores a N. A etapa seguinte de cifragem no RSA consiste, portanto, em aplicar a exponenciação modular, de acordo com a fórmula:

$$C_i \equiv M_i^e \pmod{N}$$

onde:

- M é o bloco da mensagem,

- e é o expoente público (chave pública),
- N é o módulo,
- i é a posição do bloco da mensagem,
- C_i é o bloco cifrado, ou seja, a mensagem criptografada).

Utilizando os parâmetros $e = 5$ e $N = 221$, aplica−se a fórmula de exponenciação modular a cada bloco (i) da tabela. O resultado desse processo corresponde aos blocos cifrados, conforme apresentados a seguir.

Tabela 8. Cifragem RSA dos blocos simulados

Posição (i)	Bloco Simulado (M)	Cálculo: $M^e \bmod N$	Bloco Cifrado (C)
1	77	$77^5 \bmod 221$	25
2	65	$65^5 \bmod 221$	182
3	82	$82^5 \bmod 221$	114
4	73	$73^5 \bmod 221$	99
5	65	$65^5 \bmod 221$	182

Fonte: Elaborado pelo autor.

Observe que na tabela acima, por exemplo, na posição $i = 1$, tem−se que $77^5 \bmod 221$ resulta em 25. Pode−se observar também que cada valor de M foi elevado ao expoente $e = 5$ e reduzido módulo 221, o que caracteriza a operação de cifragem RSA. A tabela ilustra como cada bloco da mensagem original foi transformado em seu correspondente cifrado, tornando a leitura do conteúdo ilegível sem a chave privada.

Diferente de cifras clássicas, como a cifra de César ou a cifra de Vigenère, que substituem letras por outras letras do alfabeto, a criptografia RSA não transforma a mensagem original em outra palavra ou sequência de caracteres legíveis. Em vez disso, o algoritmo opera diretamente sobre valores numéricos, resultantes da conversão dos caracteres por meio de uma codificação padrão, como a tabela ASCII.

O resultado do processo de cifragem RSA é uma sequência de números inteiros, que representam a versão criptografada da mensagem original:

$$\text{MARIA} \rightarrow [25, 182, 114, 99, 182].$$

A operação inversa da cifragem é a decifragem, que também se baseia na exponenciação modular, agora com uso da chave privada d. A fórmula aplicada é:

$$M \equiv C^d \bmod N,$$

onde, mantêm−se as mesmas nomenclaturas previamente adotadas.

Ao aplicar esta fórmula para cada valor cifrado da Tabela 8, obtêm−se os blocos recuperados, conforme mostrado na Tabela 9.

Tabela 9. Decifragem RSA da mensagem cifrada [25, 182, 114, 99, 182]

Bloco Cifrado (C)	Cálculo: $C^d \bmod N$	ASCII (M)	Letra
25	$25^{77} \bmod 221$	77	M
182	$182^{77} \bmod 221$	65	A
114	$114^{77} \bmod 221$	82	R
99	$99^{77} \bmod 221$	73	I
182	$182^{77} \bmod 221$	65	A

Fonte: Elaborado pelo autor.

A partir da Tabela 9, verifica−se que os valores originais utilizados para a cifragem foram completamente restaurados. Este exemplo evidencia como os fundamentos matemáticos, especialmente a aritmética modular e a função totiente de Euler, sustentam o funcionamento do algoritmo RSA.

A operação de cifragem com a chave pública e decifragem com a chave privada formam um par matemático inverso no conjunto $\mathbb{Z}_N$. Em aplicações reais, os valores de p, q, e, d e N são consideravelmente maiores, geralmente com centenas ou milhares de bits, o que torna a fatoração de N um problema computacionalmente inviável. Essa característica garante a segurança do RSA frente a ataques por força bruta, conferindo−lhe ampla utilização em contextos de comunicação segura, como protocolos HTTPS, autenticação digital e sistemas de blockchain.

4.2.2 Criptografia Simétrica (AES) e Funções Hash (SHA−256)

A criptografia simétrica, representada pelo algoritmo *Advanced Encryption Standard* (AES), é amplamente reconhecida por sua segurança e eficiência na proteção de informações digitais. Sua base está em fundamentos matemáticos rigorosos, especialmente nas operações definidas sobre o corpo finito $GF(2^8)$, também conhecido como corpo de Galois. Segundo Stallings (2006), o AES realiza transformações como substituições de bytes (*SubBytes*), permutação de linhas (*ShiftRows*), mistura de colunas (*MixColumns*) e adição da chave de rodada

(*AddRoundKey*), utilizando um conjunto de 256 elementos representados por 8 bits, organizados por álgebra modular com base no polinômio irredutível

$$x^8 + x^4 + x^3 + x + 1.$$

Essas operações matemáticas envolvem somas e multiplicações no corpo finito, o que contribui para a segurança do algoritmo ao dificultar a reversão do processo sem o conhecimento da chave original. Essa robustez se deve principalmente à complexidade das substituições não lineares e à eficante dispersão dos bits entre as rodadas de transformação. No entanto, por não ser o foco central deste trabalho, o funcionamento detalhado do AES não será aprofundado.

De forma complementar, as funções *hash* ocupam papel central na garantia da integridade e autenticidade de dados. Diferentemente dos métodos de cifragem, essas funções não são reversíveis, transformando entradas de tamanho arbitrário em saídas de comprimento fixo. O algoritmo SHA−256 (*Secure Hash Algorithm* – 256 bits), pertencente à família SHA−2, gera um valor de 256 bits (ou 64 caracteres hexadecimais) a partir de qualquer dado de entrada. Segundo Stallings (2006, p. 432), “uma função *hash* criptográfica deve possuir três propriedades essenciais: resistência à pré−imagem, resistência à segunda pré−imagem e resistência à colisão”.

A construção do SHA−256 envolve operações lógicas e aritméticas sobre blocos de 512 bits, com preenchimento inicial (padding), compressões modulares, rotações e somas em base 2^{32}. Durante as rodadas, são aplicadas funções como Ch, Maj, Σ_0 e Σ_1, todas baseadas em lógica binária, para promover confusão e difusão, características desejáveis em sistemas criptográficos modernos. O uso de compressão modular impede que alterações mínimas na entrada resultem em mudanças sutis na saída, o que caracteriza o chamado efeito avalanche (Stallings, 2006).

A resistência à colisão refere−se à dificuldade de encontrar duas entradas distintas que gerem o mesmo valor de *hash*. Para o SHA−256, com 256 bits de saída, essa segurança baseia−se na complexidade de ataques do tipo birthday problem, cujo esforço de força bruta se aproxima de 2^{128} tentativas. Conforme afirma Coutinho (2009), essa resistência está relacionada ao fato de a função *hash* atuar como uma impressão digital dos dados: qualquer modificação mínima altera substancialmente o

valor final. Isso a torna essencial em aplicações como armazenamento seguro de senhas, verificação de integridade de arquivos, validação de mensagens e tecnologias como blockchain.

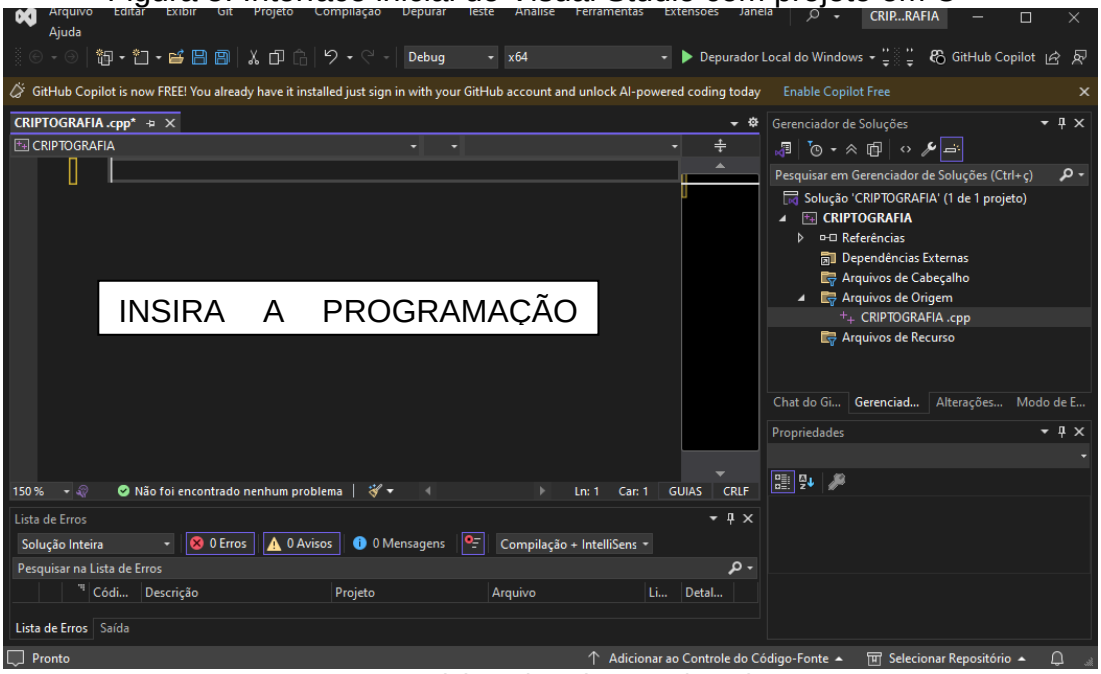
Tanto o AES quanto o SHA−256 representam pilares essenciais da criptografia moderna e são frequentemente utilizados em conjunto com sistemas de chave pública, como o RSA, compondo estruturas híbridas que oferecem segurança, desempenho e integridade nos ambientes digitais contemporâneos.

4.3 Ferramentas Práticas Aplicadas à Criptografia

Para possibilitar a aplicação concreta dos algoritmos criptográficos discutidos nesta dissertação, optou−se pela utilização do Microsoft Visual Studio como ambiente de desenvolvimento. Esta escolha deve−se à segurança da plataforma, ao seu suporte à linguagem C++ e aos recursos integrados que favorecem o aprendizado estruturado da lógica de programação, aspecto essencial para a compreensão dos mecanismos matemáticos subjacentes à criptografia. Além disso, a possibilidade de instalar a plataforma diretamente no computador facilita seu uso em ambientes educacionais e domésticos, proporcionando maior autonomia e acessibilidade ao usuário.

Além de dispor de uma interface gráfica amigável, o Visual Studio permite a criação de projetos modulares, a simulação de entradas e saídas de dados em tempo real, bem como o debug de programas, funcionalidades essas que são especialmente úteis em contextos educacionais. Sua ampla documentação e o apoio de uma grande comunidade de usuários tornam a ferramenta acessível e funcional tanto para docentes quanto para estudantes do Ensino Médio que estejam ingressando nos estudos da programação e da matemática aplicada.

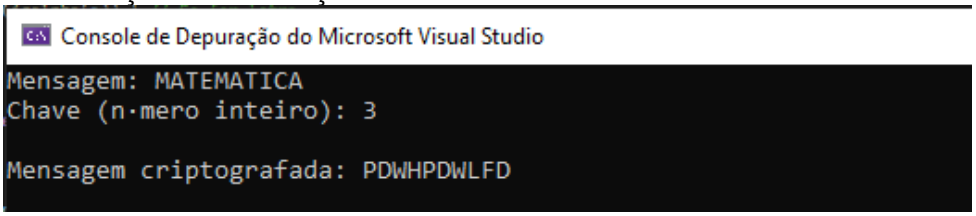
Figura 5. Interface inicial do Visual Studio com projeto em C++



Fonte: elaborado pelo autor (2025)

A escolha do Visual Studio se justifica também pela sua compatibilidade com projetos educacionais, permitindo a criação de aplicativos simples e personalizados**,** nos quais os estudantes podem manipular mensagens criptografadas, explorar cifras clássicas como César, Vigenère e Multiplicativa, e compreender sua lógica algorítmica por meio da implementação prática.

Figura 6. Simulação da execução de cifra de César com entrada e saída via console

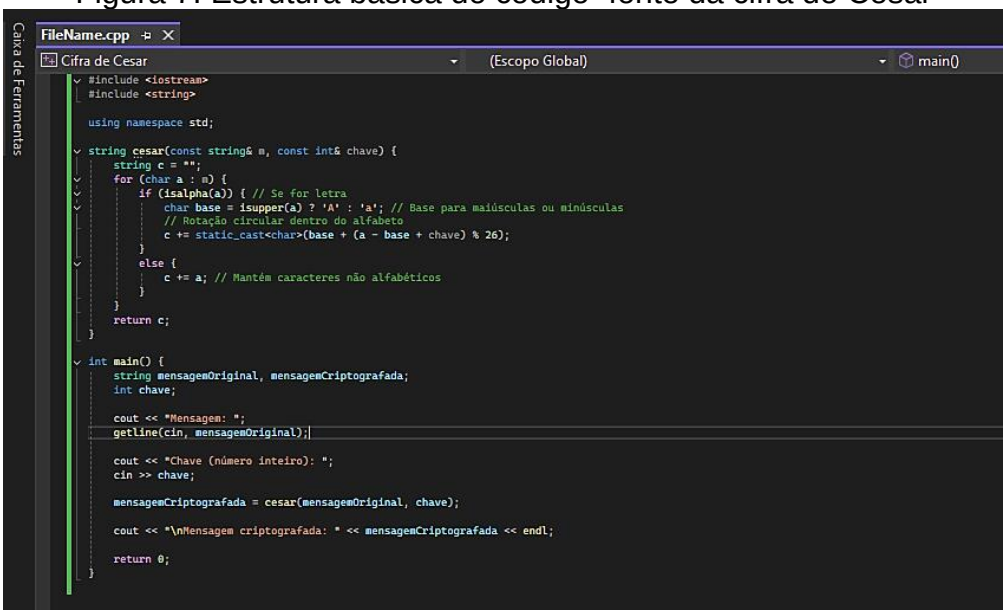


Fonte: elaborado pelo autor (2025)

No contexto desta dissertação, os algoritmos foram codificados em C++**,** linguagem amplamente adotada tanto no meio acadêmico quanto no mercado de tecnologia. O uso dessa linguagem permite não apenas a introdução aos fundamentos da programação estruturada, mas também o desenvolvimento de competências computacionais e matemáticas por parte dos alunos. No Capítulo 5, tais implementações serão descritas de forma detalhada, acompanhadas de explicações

passo a passo, contextualização teórica e propostas de atividades práticas para aplicação em sala de aula.

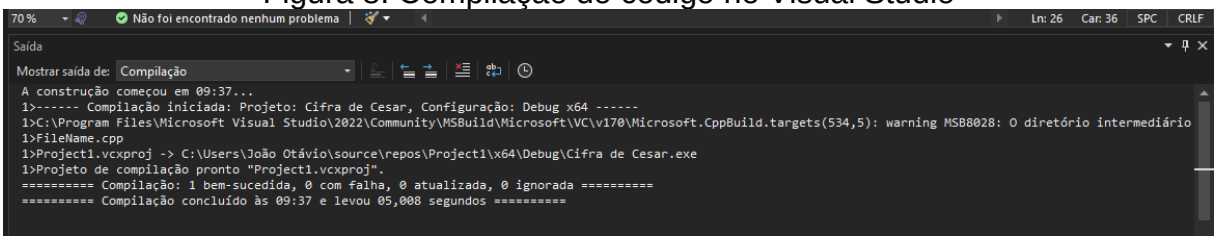
Figura 7. Estrutura básica do código−fonte da cifra de César



Fonte: elaborado pelo autor (2025)

Após a digitação completa do código no ambiente do Visual Studio, é necessário realizar a compilação para verificar se há eventuais de erros na estrutura da programação. Esse processo transforma o código−fonte escrito em linguagem humana, em uma linguagem de máquina executável. A compilação pode ser iniciada pressionando as teclas Ctrl + Alt + B, acionando o mecanismo interno que analisa sintaticamente o programa.

Figura 8. Compilação do código no Visual Studio

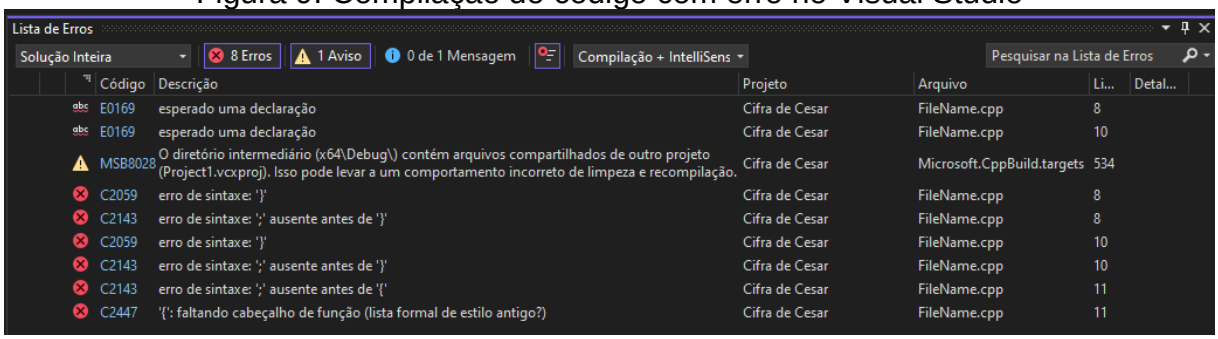


Fonte: elaborado pelo autor (2025)

Caso ocorram erros, o compilador exibirá mensagens específicas na janela de saída, indicando a linha e o tipo de problema encontrado. Nessa etapa, é fundamental que o aluno ou professor identifique e corrija essas falhas antes de

prosseguir com a execução do programa. A presença de erros impedirá a geração do arquivo executável e, consequentemente, a realização do teste da cifra.

Figura 9. Compilação do código com erro no Visual Studio



Fonte: elaborado pelo autor (2025)

Essa associação entre tecnologia e matemática amplia o alcance pedagógico da proposta, tornando a aprendizagem da criptografia mais envolvente, significativa e alinhada com as práticas contemporâneas de ensino. A partir do uso do Visual Studio, é possível transformar conteúdos teóricos em experiências práticas e exploratórias, promovendo um ensino interdisciplinar**,** contextualizado e tecnologicamente engajado**.**

Embora o Visual Studio seja o ambiente central adotado neste trabalho, vale mencionar outras ferramentas que também oferecem potencial didático no ensino da criptografia:

- Scratch: programação em blocos, ideal para introdução a cifras simples como a de César;
- Python com Google Colab: execução de códigos criptográficos em nuvem, com bibliotecas especializadas como cryptography;
- Construct 3: criação de jogos interativos e simulações com criptografia gamificada;
- wxMaxima: ideal para visualizar operações de aritmética modular sem exigir programação textual.

Essas alternativas reforçam a premissa de que o uso pedagógico da tecnologia potencializa o ensino da Matemática, ao tornar os conteúdos mais acessíveis, contextualizados e conectados à cultura digital dos alunos.

4.4 TRADUÇÃO DOS CÓDIGOS−FONTE IMPLEMENTADOS EM C++

Considerando o caráter interdisciplinar desta pesquisa, que integra matemática e programação na abordagem de algoritmos criptográficos, é pertinente apresentar uma síntese dos principais comandos e estruturas empregados na implementação dos códigos em linguagem C++. A familiarização com esses elementos contribui para a reprodução das experiências desenvolvidas, bem como para a adaptação dos códigos a diferentes realidades escolares por parte de professores e estudantes.

Ao longo deste estudo, foram implementadas diversas cifras clássicas, como César, Afim, Multiplicativa, Vigenère, e o algoritmo RSA, utilizando um conjunto comum de instruções, operadores e funções. A seguir, a Tabela 10 apresenta a tradução e explicação didática desses componentes, a fim de ampliar a acessibilidade técnica do conteúdo e favorecer sua utilização em ambientes educacionais.

Tabela 10. Tradução e explicação dos principais comandos C++ utilizados na implementação dos algoritmos de criptografia

Comando ou Função	Tradução / Explicação
#include <iostream>	Importa a biblioteca de entrada e saída (ex.: cin, cout).
#include <string>	Permite uso de strings (textos) no programa.
using namespace std;	Evita repetição de std:: antes de comandos padrão como cout, cin, etc.
int main()	Função principal do programa. Onde tudo começa a ser executado.
getline(cin, mensagem)	Lê uma linha completa de texto digitada pelo usuário.
cin >> chave	Lê um valor (geralmente numérico) digitado pelo usuário.
isalpha(letra)	Verifica se o caractere é uma letra (A–Z ou a–z).
isupper(letra)	Verifica se a letra é maiúscula.
toupper(letra)	Converte letra minúscula para maiúscula.
letra − base	Transforma a letra em um número (ex: A → 0, B → 1...).
base + c	Converte um número de volta em letra.
%	Operador de módulo: retorna o resto da divisão. Usado na aritmética modular.
if, else, while, for	Estruturas de controle condicional ou repetição.
return valor;	Encerra a função e devolve um valor.
string resultado = \"\";	Inicializa uma string vazia onde será armazenado o texto cifrado.
size_t pos = ALFABETO.find(letra)	Busca a posição do caractere na string do alfabeto.

expMod(base, expoente, n)	Função para cálculo de potência modular (usada no RSA).
inversoModular(e, phi)	Função para calcular o inverso de e mod phi usando o algoritmo de Euclides.
mdc(a, b)	Calcula o máximo divisor comum entre dois números.
static_cast<char>(...)	Converte um número inteiro para caractere com segurança.
cout << ...	Imprime textos ou resultados na tela.

Fonte: Elaborado pelo autor (2025)

A sistematização apresentada no quadro não apenas organiza os principais comandos utilizados nos códigos−fonte, como também favorece sua leitura pedagógica, facilitando a compreensão dos processos computacionais envolvidos na implementação dos algoritmos. Essa abordagem contribui diretamente para o letramento digital e para o desenvolvimento do raciocínio matemático, especialmente entre educadores que desejam explorar a interdisciplinaridade entre programação e matemática no contexto escolar. Além disso, a clareza e a apresentação funcional dos elementos da linguagem C++ permitem que as atividades desenvolvidas sejam facilmente interpretadas, adaptadas e replicadas em diferentes realidades educacionais.

CAPÍTULO 5 – ATIVIDADES DIDÁTICAS COM CRIPTOGRAFIA: UM GUIA PARA PROFESSORES

Este capítulo tem como objetivo apresentar um conjunto de atividades didáticas baseadas em diferentes métodos de criptografia, como as cifras de César, Afim, Multiplicativa e Vigenère, com o intuito de tornar o ensino da Matemática mais envolvente e aplicável ao cotidiano dos estudantes. Tais atividades propõem o desenvolvimento de competências matemáticas por meio da resolução de problemas concretos, incentivando o raciocínio lógico, a investigação e a criatividade dos alunos.

As propostas aqui reunidas buscam atender diferentes níveis de ensino, podendo ser adaptadas para turmas do ensino fundamental e médio. Os objetivos didáticos principais incluem:

- Introduzir os conceitos básicos de criptografia e suas aplicações históricas e contemporâneas;
- Desenvolver a compreensão de operações aritméticas modulares e funções matemáticas no contexto da codificação de mensagens;
- Estimular a resolução de problemas, a análise de padrões e a construção de estratégias;
- Promover o trabalho colaborativo e a interdisciplinaridade, articulando matemática com linguagens, história, informática e segurança digital;
- Encorajar o protagonismo dos estudantes em desafios criptográficos contextualizados.

Acredita−se que a aprendizagem da matemática possa se beneficiar significativamente do uso de desafios criptográficos como instrumentos de mediação pedagógica. Ao desvendar códigos e criar mensagens cifradas, os alunos experimentam uma vivência matemática autêntica, ativa e instigante, transformando−se em agentes do próprio conhecimento.

As atividades apresentadas a seguir foram organizadas de forma a proporcionar uma abordagem lúdica e prática da criptografia, combinando experiências no papel, com o uso de rodas de codificação físicas, e aplicações computacionais desenvolvidas na linguagem C++ por meio do ambiente Visual Studio. Cada proposta contempla objetivos pedagógicos bem definidos, instruções detalhadas, trechos de código comentados e exemplos de execução, promovendo o

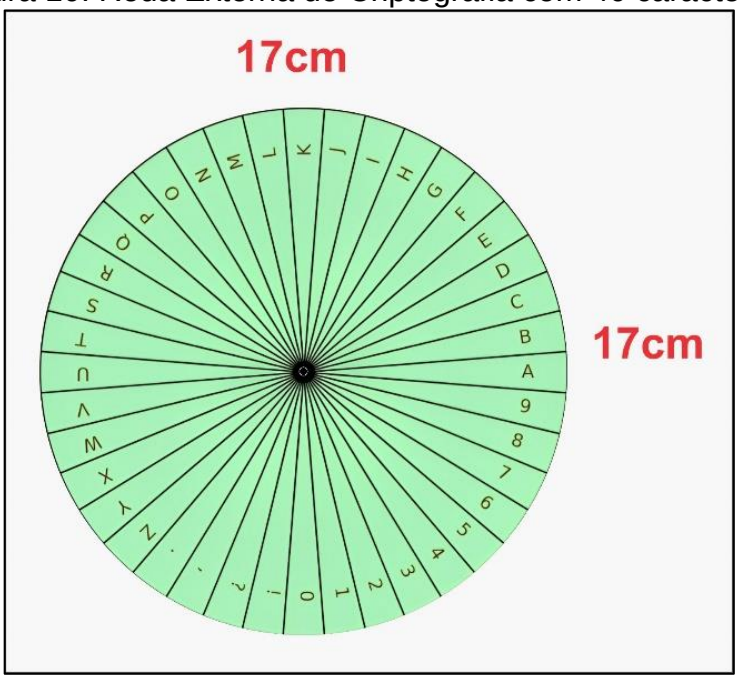
raciocínio lógico, a aprendizagem significativa e o desenvolvimento do pensamento computacional.

Atividade 1: Códigos Secretos com a Roda de Criptografia

Os discentes receberão previamente as rodas de criptografia conforme ilustrado nas figuras 10 e 11 disponibilizadas no anexo desta dissertação. A ferramenta é composta por 40 caracteres que integram um alfabeto expandido, abrangendo as letras do alfabeto latino ($A - Z$), os sinais de pontuação (.,?!), e os algarismos numéricos ($0 - 9$). A estrutura da roda consiste em uma parte externa fixa e uma parte interna móvel, cuja rotação possibilita a definição do deslocamento necessário, denominado chave, para a codificação e decodificação das mensagens.

O objetivo da atividade é proporcionar aos estudantes uma vivência prática do funcionamento da Cifra de César, empregando o criptodisco físico como recurso manipulável para compreender a transformação de mensagens em códigos. A proposta integra conceitos de aritmética modular e lógica criptográfica, favorecendo a compreensão do uso da matemática em situações reais relacionadas à segurança da informação.

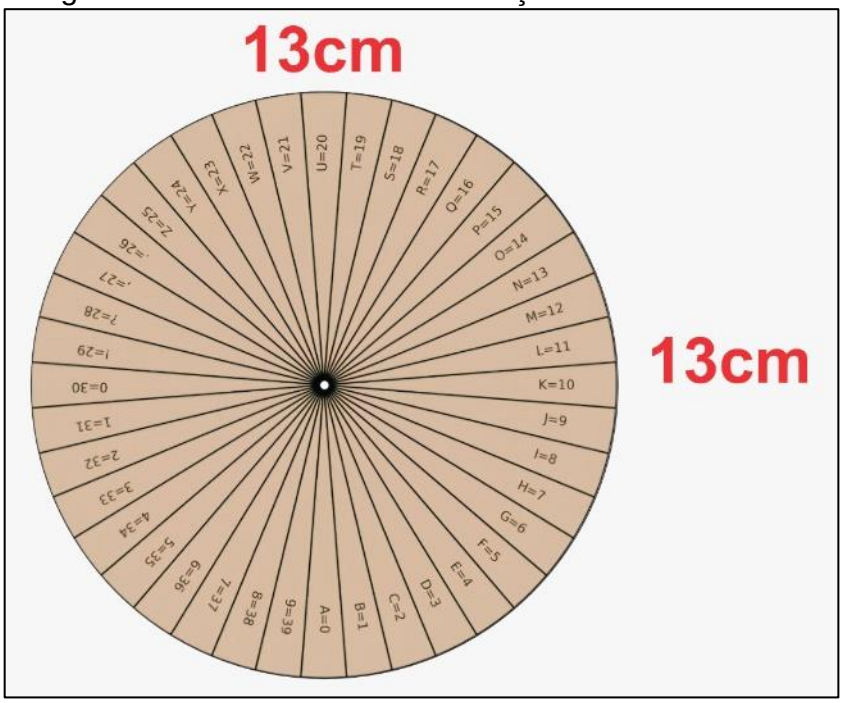
Figura 10. Roda Externa de Criptografia com 40 caracteres.



Fonte: Elaboração do autor com o uso de ferramenta de IA e editoração no Canvas

Para a montagem do criptodisco, recomenda−se que as duas rodas, externa e interna, sejam recortadas cuidadosamente ao longo de seus contornos circulares. Em seguida, a roda interna deve ser posicionada sobre a roda externa de forma que ambas fiquem perfeitamente centralizadas.

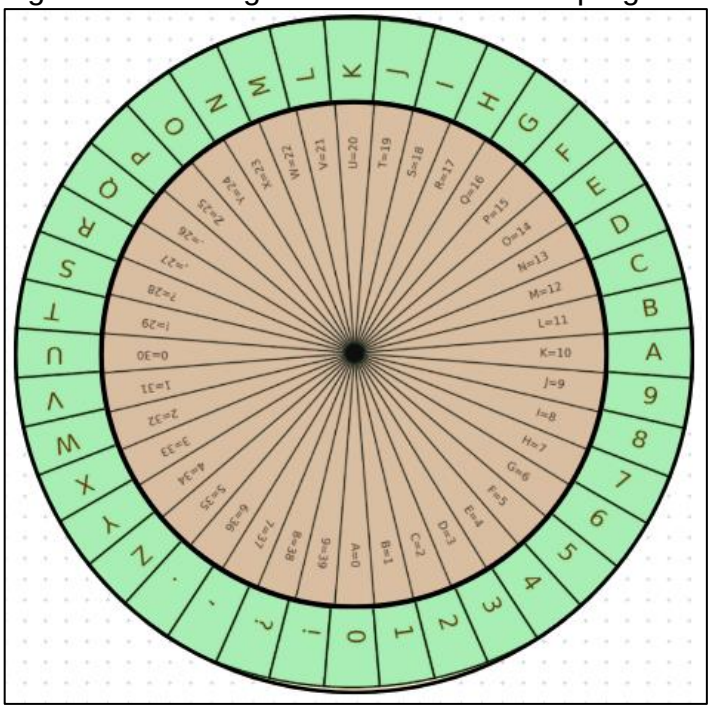
Figura 11. Roda Interna com notação A = 0 até 9 = 39.



Fonte: Elaboração do autor com o uso de ferramenta de IA e editoração no Canvas

Com o auxílio do professor, o aluno poderá fixar as rodas no centro utilizando uma tachinha metálica ou prendedor de papel, garantindo que a roda interna permaneça giratória, possibilitando assim os ajustes de deslocamento necessários à codificação e decodificação de mensagens.

Figura 12. Montagem final da roda de criptografia.



Fonte: Elaboração do autor com o uso de ferramenta de IA e editoração no Canvas

Com a roda de criptografia devidamente montada, aplica−se o princípio da Cifra de César ajustando−se a posição da roda interna de acordo com a chave de codificação escolhida, representada por um valor de deslocamento entre 0 e 39. Para isso, alinha−se o caractere “A” da roda externa ao número correspondente à chave na roda interna. No exemplo ilustrado na Figura 12, a letra “A” da roda externa está posicionada sobre o número “10” da roda interna, o que indica uma chave igual a 10. Com esse alinhamento, cada caractere da mensagem original será substituído pelo caractere situado 10 posições adiante, resultando, por exemplo, em $\mathrm{A} \rightarrow \mathrm{K}$.

Assim sendo, para codificar uma mensagem, localize cada caractere da mensagem na roda externa e substitua pelo caractere correspondente diretamente na roda interna. Considere o exemplo:

- Mensagem original: ESCOLA
- Chave utilizada: 5
- Instruções: Com a roda ajustada, de modo que $\mathrm{A} = 5$, será possível identificar que a letra 'E' da roda externa será substituída por 'J' da roda interna, que a letra ‘S’ da roda externa será substituída por ‘X’ da roda interna, e assim por diante.

Como resultado do processo de cifragem realizado com o criptodisco, obteve-se a mensagem criptografada “JXHTQF”**,** representando a transformação completa do texto original segundo a chave definida.

Para decodificar uma mensagem, alinhe novamente a chave usada e, desta vez, localize o caractere cifrado na roda interna, substituindo pelo correspondente da roda externa. Considere o exemplo:

- Mensagem cifrada: R !R6V!R6ZTR V UZ8V46ZUR
- Chave utilizada: 17
- Instruções: Com a roda ajustada, de modo que $A = 17$, será possível identificar que a o caractere 'R' da roda interna será substituído por 'A' da roda externa, que o caractere ‘!’ será substituído por ‘M’, e assim por diante.

Logo, o processo de decodificação resultou na mensagem “A MATEMATICA É DIVERTIDA”

Sugere-se que o professor apresente brevemente aos alunos a fórmula matemática que representa o processo de codificação, de modo a relacionar a prática com o conceito de aritmética modular. A operação pode ser expressa da seguinte forma:

$$c \equiv m + k \pmod{40}$$

em que c representa o caractere criptografado, m indica a posição do caractere original no alfabeto expandido, k corresponde à chave de deslocamento e n, neste caso igual a 40, define o tamanho total do alfabeto utilizado.

A manipulação da roda de criptografia contribui para o desenvolvimento de diversas competências relevantes no processo de aprendizagem matemática, tais como a compreensão dos conceitos de congruência e aritmética modular, a visualização de deslocamentos e padrões numéricos, a articulação interdisciplinar entre Matemática e Criptografia, além do estímulo ao raciocínio lógico e à construção de estratégias para a codificação de informações.

A proposta permite que os alunos manipulem fisicamente uma ferramenta de criptografia e percebam, de maneira concreta, como um simples deslocamento sistemático pode transformar uma mensagem compreensível em um código ilegível.

Atividade 2. Desvendando Mensagens com a Cifra de César no Visual Studio

Neste experimento didático, os alunos implementam a Cifra de César, uma das formas mais antigas de criptografia. A técnica, atribuída ao imperador romano Júlio César, consiste em deslocar cada letra da mensagem original um número fixo de posições no alfabeto. O objetivo da atividade é explorar a aritmética modular por meio da implementação da Cifra de César, permitindo que os alunos compreendam noções de programação e raciocínio lógico. Matematicamente, a cifra é expressa como:

$$c \equiv m + k \pmod{26},$$

onde:

- c é o valor da letra criptografada;
- m é o valor numérico da letra original (0 a 25);
- k é a chave de deslocamento (inteiro positivo);
- a operação é feita módulo 26 (número de letras do alfabeto latino).

Para realizar o processo inverso, ou seja, decifrar a mensagem codificada, aplica-se a operação contrária ao deslocamento efetuado na cifragem. Dessa forma, a letra original é obtida pela subtração da chave k do valor correspondente à letra cifrada, segundo

$$m \equiv c - k \pmod{26}.$$

Essa operação garante que o texto original seja recuperado corretamente, desde que a mesma chave de deslocamento utilizada na cifragem seja conhecida.

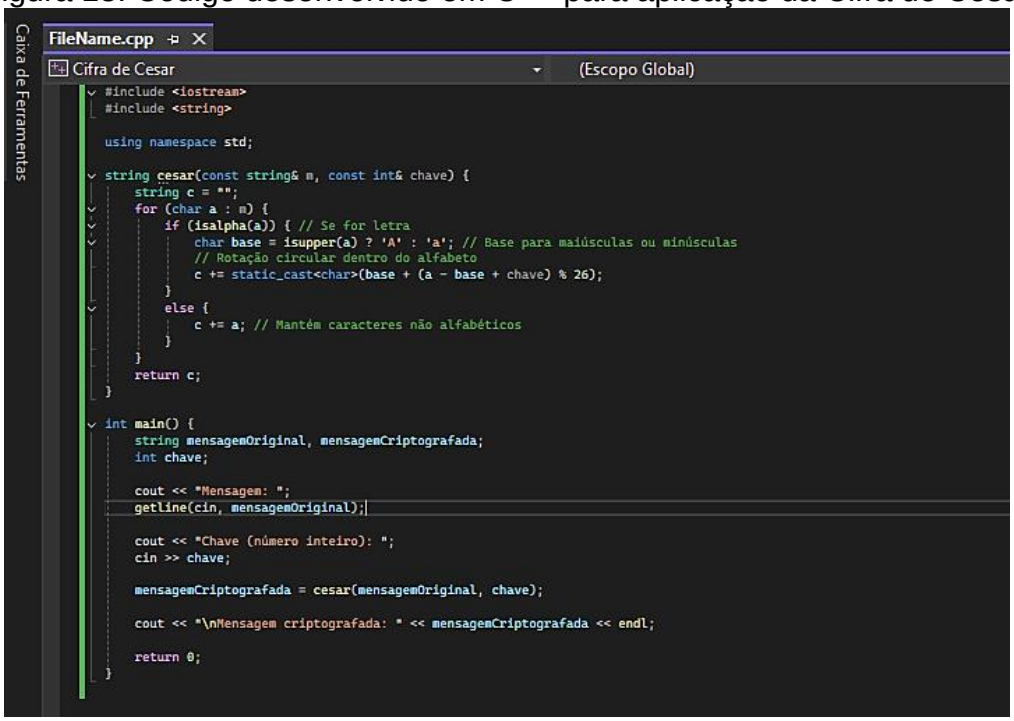
A implementação da Cifra de César foi desenvolvida em linguagem C++**,** utilizando o ambiente de programação Visual Studio. No código, apenas os caracteres pertencentes ao alfabeto são submetidos ao processo de cifragem, permanecendo os demais inalterados. A seguir, apresenta-se o trecho da função responsável pela codificação, que deve ser inserido no editor do aplicativo em uso, conforme os parâmetros previamente definidos.

```
#include <iostream>
#include <string>
using namespace std;
string cesar(const string& m, const int& chave) {
```

```
    string c = "";
    for (char a : m) {
        if (isalpha(a)) { // Se for letra
            char base = isupper(a) ? 'A' : 'a'; // Base para maiúsculas ou minúsculas
            // Rotação circular dentro do alfabeto
            c += static_cast<char>(base + (a − base + chave) % 26);
        }
        else {
            c += a; // Mantém caracteres não alfabéticos
        }
    }
    return c;
}
int main() {
    string mensagemOriginal, mensagemCriptografada;
    int chave;
    cout << "Mensagem: ";
    getline(cin, mensagemOriginal);
    cout << "Chave (número inteiro): ";
    cin >> chave;
    mensagemCriptografada = cesar(mensagemOriginal, chave);
    cout << "\nMensagem criptografada: " << mensagemCriptografada << endl;
    return 0;
                }
```

Esse código é ideal para uso didático, pois permite que os alunos visualizem diretamente como a fórmula matemática do deslocamento é aplicada a cada caractere da *string*, reforçando o conceito de aritmética modular. A estrutura do código também estimula a leitura lógica dos comandos e o desenvolvimento de habilidades básicas de programação.

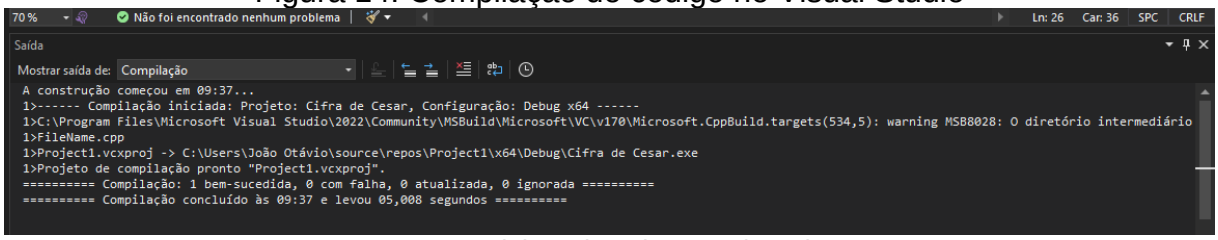
Figura 13. Código desenvolvido em C++ para aplicação da Cifra de César



Fonte: elaborado pelo autor (2025)

Após a digitação completa do código no ambiente do Visual Studio, a compilação pode ser iniciada pressionando as teclas Ctrl + Alt + B, acionando o mecanismo interno que analisa sintaticamente o programa.

Figura 14. Compilação do código no Visual Studio



Fonte: elaborado pelo autor (2025)

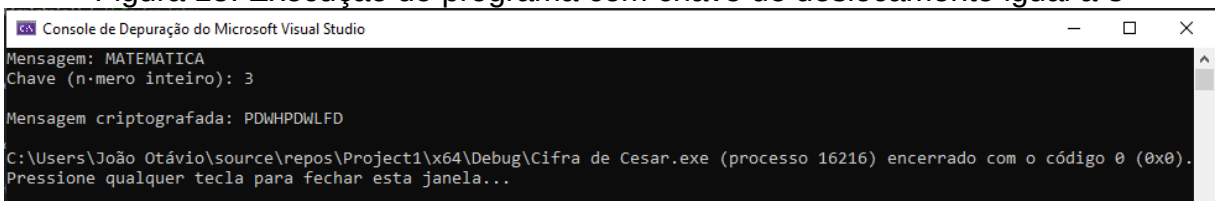
Para ilustrar a atividade, foram realizados dois testes com a mesma palavra, mas chaves distintas:

I. Mensagem original: MATEMATICA

Chave utilizada: $k = 3$

Mensagem criptografada: PDWHPDWLFD

Figura 15. Execução do programa com chave de deslocamento igual a 3



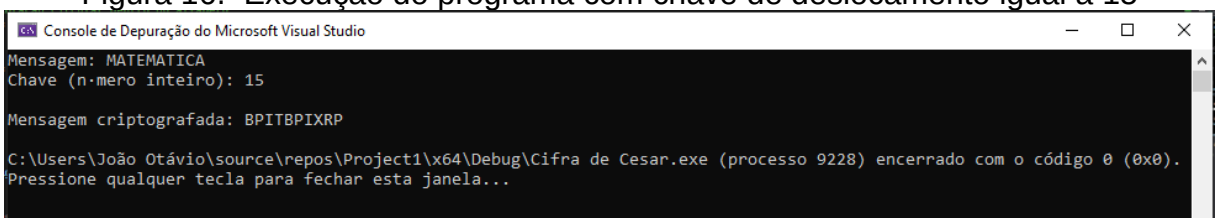
Fonte: elaborado pelo autor (2025)

II. Mensagem original: MATEMATICA

Chave utilizada: $k = 15$

Mensagem criptografada: BPITBPIXRP

Figura 16. Execução do programa com chave de deslocamento igual a 15



Fonte: elaborado pelo autor (2025)

A comparação entre os dois exemplos mostra como pequenas variações na chave podem alterar completamente a mensagem cifrada. Esse é um dos princípios centrais da segurança na criptografia por chave simétrica.

Essa atividade promove não apenas o aprendizado da aritmética modular de forma aplicada, mas também desenvolve competências importantes em lógica computacional e resolução de problemas. A associação entre conceitos matemáticos e sua aplicação em um programa real facilita a compreensão dos estudantes, além de incentivar a interdisciplinaridade entre matemática, história e tecnologia. Adicionalmente, o uso do exemplo clássico de Júlio César (com $k = 3$) fornece uma referência histórica de fácil assimilação pelos alunos, contribuindo para o aspecto lúdico e contextualizado da proposta.

Atividade 3. Expandindo o Código com a Criptografia Aditiva no Visual Studio

Diferente da cifra aditiva clássica que utiliza apenas as 26 letras do alfabeto latino, a criptografia aditiva com alfabeto expandido trabalha com um conjunto mais amplo de caracteres:

ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789 .,:!?

A técnica apresentada é fundamentada na aritmética modular, sendo aplicada sobre um alfabeto de tamanho expandido, o que possibilita trabalhar com um conjunto maior de símbolos em relação às cifras clássicas. O objetivo da atividade visa ampliar o repertório dos estudantes ao trabalhar com alfabetos personalizados, aproximando a matemática de contextos reais de segurança da informação. A fórmula matemática que representa o processo de codificação pode ser expressa como

$$c \equiv m + k \pmod{n},$$

onde:

- c representa o caractere criptografado;
- m é a posição do caractere original no alfabeto expandido;
- k é a chave de deslocamento;
- n é o tamanho total do alfabeto utilizado.

O processo de decifragem, por sua vez, consiste na operação inversa. Para recuperar a mensagem original, é necessário subtrair a chave de deslocamento k do valor numérico associado ao caractere cifrado. Dessa forma, a fórmula de decodificação assume a forma:

$$m \equiv c - k \pmod{n}.$$

Esse mecanismo garante que, mesmo ao ultrapassar os limites do alfabeto, a operação modular assegure o retorno ao intervalo válido de símbolos, mantendo a consistência da transformação e possibilitando a recuperação exata da mensagem inicial.

A atividade foi desenvolvida no ambiente de programação Visual Studio, utilizando a linguagem C++. Para essa aplicação, considerou−se um alfabeto expandido composto por 44 caracteres únicos, abrangendo letras, números e sinais de pontuação. A seguir, apresenta−se o código da função responsável pela implementação da Criptografia Aditiva com base nesse alfabeto expandido.

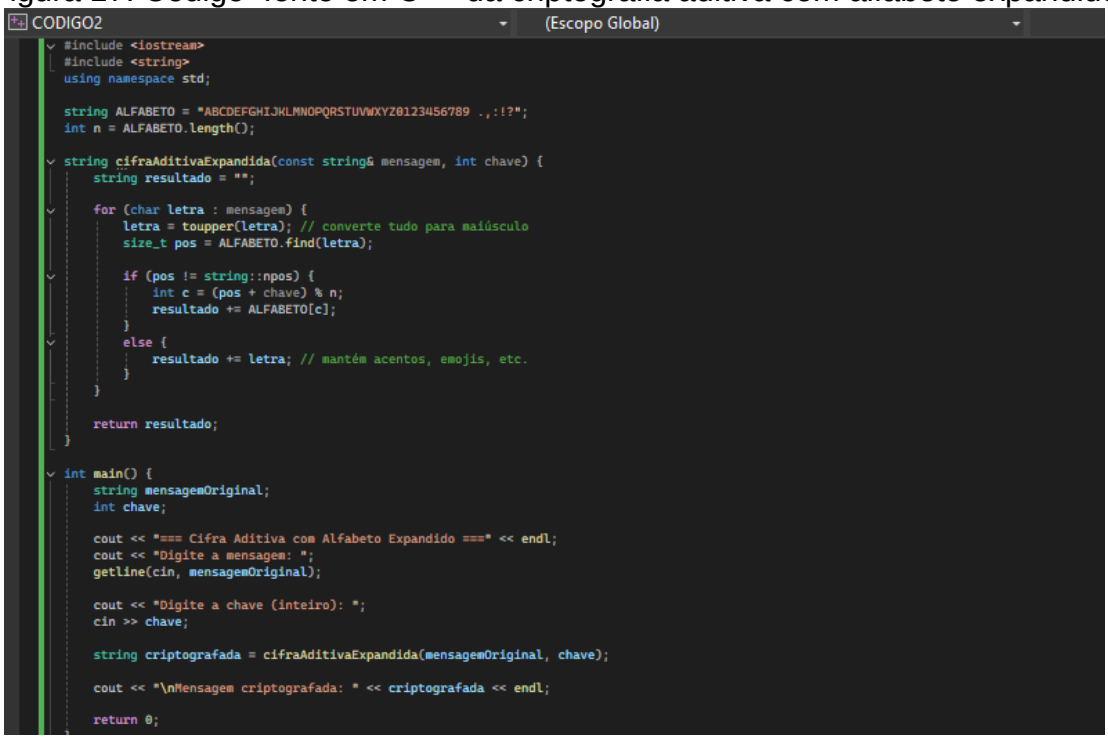
```
#include <iostream>
#include <string>
using namespace std;
string ALFABETO = "ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789 .,:!?";
int n = ALFABETO.length();
```

```
string cifraAditivaExpandida(const string& mensagem, int chave) {
    string resultado = "";
    for (char letra : mensagem) {
        letra = toupper(letra); // converte tudo para maiúsculo
        size_t pos = ALFABETO.find(letra);
        if (pos != string::npos) {
            int c = (pos + chave) % n;
            resultado += ALFABETO[c];
        }
        else {
            resultado += letra; // mantém acentos, emojis, etc.
        }
    }
    return resultado;
}
int main() {
    string mensagemOriginal;
    int chave;
    cout << "=== Cifra Aditiva com Alfabeto Expandido ===" << endl;
    cout << "Digite a mensagem: ";
    getline(cin, mensagemOriginal);
    cout << "Digite a chave (inteiro): ";
    cin >> chave;
    string criptografada = cifraAditivaExpandida(mensagemOriginal, chave);
    cout << "\nMensagem criptografada: " << criptografada << endl;
    return 0;

}
```

O algoritmo percorre a mensagem, identifica a posição de cada caractere dentro do alfabeto expandido, aplica a operação modular e substitui o caractere original por outro deslocado conforme a chave.

Figura 17. Código−fonte em C++ da criptografia aditiva com alfabeto expandido



```
CODIGO2                                                    (Escopo Global)

#include <iostream>
#include <string>
using namespace std;

string ALFABETO = "ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789 .,;!?";
int n = ALFABETO.length();

string cifraAditivaExpandida(const string& mensagem, int chave) {
    string resultado = "";

    for (char letra : mensagem) {
        letra = toupper(letra); // converte tudo para maiúsculo
        size_t pos = ALFABETO.find(letra);

        if (pos != string::npos) {
            int c = (pos + chave) % n;
            resultado += ALFABETO[c];
        }
        else {
            resultado += letra; // mantém acentos, emojis, etc.
        }
    }

    return resultado;
}

int main() {
    string mensagemOriginal;
    int chave;

    cout << "=== Cifra Aditiva com Alfabeto Expandido ===" << endl;
    cout << "Digite a mensagem: ";
    getline(cin, mensagemOriginal);

    cout << "Digite a chave (inteiro): ";
    cin >> chave;

    string criptografada = cifraAditivaExpandida(mensagemOriginal, chave);

    cout << "\nMensagem criptografada: " << criptografada << endl;

    return 0;
```

Fonte: Elaborado pelo autor (2025)

A compilação do código no Visual Studio pode ser realizada pressionando as teclas Ctrl + Alt + B**.** Caso haja erros de sintaxe, o compilador exibirá mensagens indicativas para que o estudante possa identificar e corrigir antes de executar o programa.

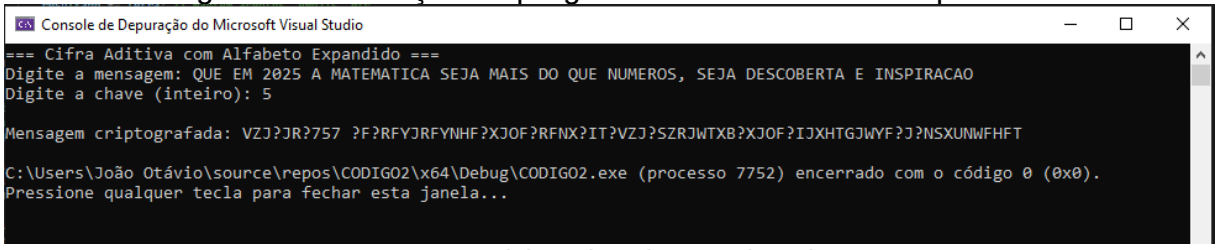
A execução prática da cifra com alfabeto expandido foi realizada com a seguinte configuração:

Mensagem original: QUE EM 2025 A MATEMATICA SEJA MAIS DO QUE NUMEROS, SEJA DESCOBERTA E INSPIRACAO

Chave utilizada: 5

Mensagem criptografada: VZJ?JR?757??F?RFYJRFYNHF?XJOF ?RFNX?IT?VZJ?SZRJWTXB?XJOF?IJXHTGJWYF?J?NSXUNWFHFT

Figura 18. Execução do programa com cifra aditiva expandida



```
Console de Depuração do Microsoft Visual Studio
=== Cifra Aditiva com Alfabeto Expandido ===
Digite a mensagem: QUE EM 2025 A MATEMATICA SEJA MAIS DO QUE NUMEROS, SEJA DESCOBERTA E INSPIRACAO
Digite a chave (inteiro): 5

Mensagem criptografada: VZJ?JR?757 ?F?RFYJRFYNHF?XJOF?RFNX?IT?VZJ?SZRJWTXB?XJOF?IJXHTGJWYF?J?NSXUNWFHFT

C:\Users\João Otávio\source\repos\CODIGO2\x64\Debug\CODIGO2.exe (processo 7752) encerrado com o código 0 (0x0).
Pressione qualquer tecla para fechar esta janela...
```

Fonte: Elaborado pelo autor (2025)

Este exemplo ilustra como o uso de alfabetos expandidos potencializa o ensino da aritmética modular, permitindo aos alunos visualizar a generalização da congruência para diferentes conjuntos de símbolos. Ao incluir letras, números e sinais de pontuação, a atividade aproxima os conceitos matemáticos do cotidiano digital, evidenciando aplicações práticas em áreas como segurança da informação, criptografia e linguagens formais.

Além do aprofundamento conceitual, a proposta promove o desenvolvimento de competências essenciais à formação contemporânea, como raciocínio lógico, abstração e pensamento algorítmico. Ao integrar matemática com elementos de informática, linguagem e cidadania digital, favorece uma aprendizagem ativa e significativa, em sintonia com os desafios educacionais da cultura digital

Atividade 4. Função Afim: Codificando com Critério de MDC

A cifra afim pode ser compreendida como uma generalização da cifra aditiva, pois combina simultaneamente a multiplicação e a adição no contexto da aritmética modular. Diferentemente da cifra de César, que emprega apenas um deslocamento fixo, a cifra afim introduz um fator multiplicativo que amplia significativamente as possibilidades de codificação. O objetivo da atividade é aprofundar a compreensão sobre inversos modulares e sua importância na criptografia, utilizando programação estruturada como recurso pedagógico. O processo de criptografia é representado pela seguinte expressão matemática:

$$c \equiv (a \cdot m + b) \pmod{26},$$

onde:

- c é o valor da letra criptografada;
- m é a posição da letra original no alfabeto (0 a 25);
- a é o fator multiplicativo;
- b é o deslocamento aditivo;
- 26 indica o tamanho do alfabeto.

Para que a cifra seja válida, é necessário que o fator multiplicativo a seja coprimo com o módulo 26, isto é:

$$\mathrm{mdc}(a\,,26)=1.$$

Essa condição garante a existência de um inverso multiplicativo de a no conjunto dos inteiros módulo 26, elemento fundamental para a reversibilidade do processo.

A decifragem da cifra afim consiste, portanto, na aplicação da operação inversa da codificação. A fórmula utilizada para recuperar o valor original é:

$$m \equiv a^{-1} \cdot (c - b) \pmod{26},$$

onde:

- m é a posição da letra original no alfabeto,
- c corresponde ao valor da letra criptografada,
- a^{-1} representa o inverso multiplicativo de a módulo 26,
- b é o deslocamento aditivo.

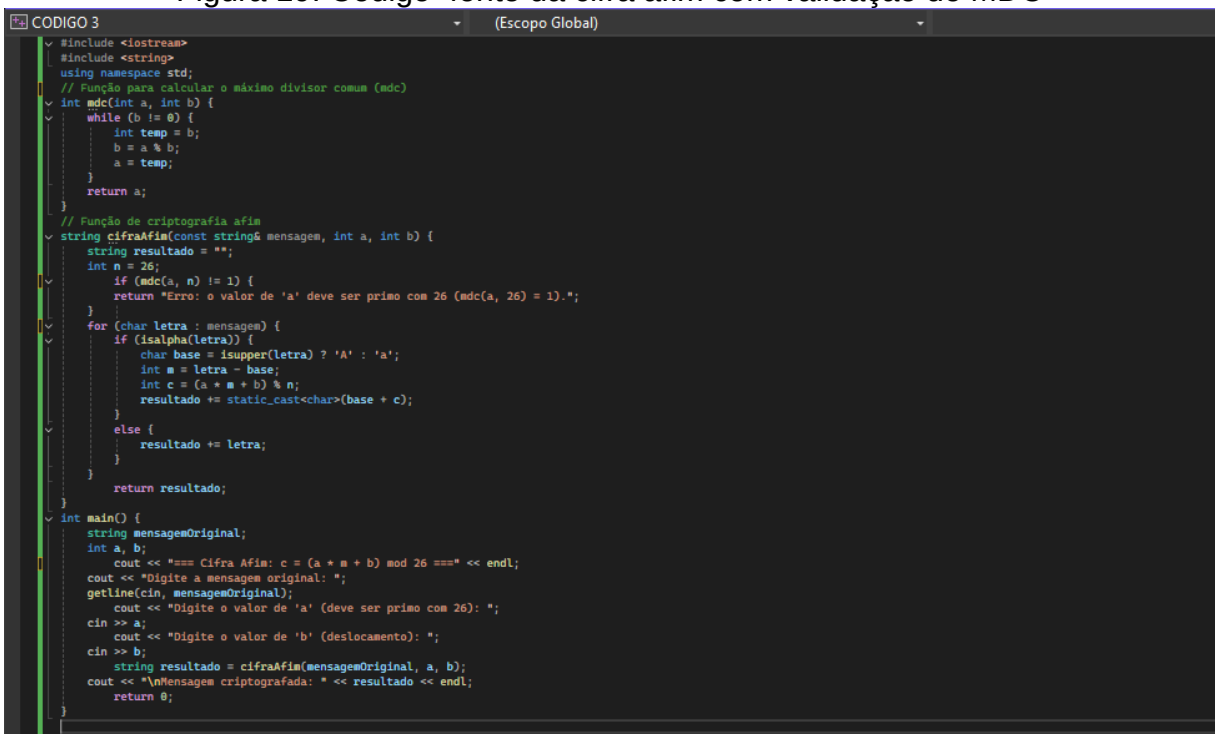
O cálculo de a^{-1} assegura que a multiplicação modular seja devidamente revertida, permitindo a recuperação correta da mensagem inicial. Esse método evidencia como a aritmética modular se mostra essencial para a segurança e a reversibilidade das cifras clássicas, estabelecendo um vínculo direto entre conceitos matemáticos de divisibilidade e aplicações criptográficas.

Na implementação, o código foi desenvolvido em C++ no ambiente Visual Studio, e conta com uma função adicional para validar se o número a é coprimo com 26. Caso contrário, o sistema impede a execução da criptografia, garantindo segurança e robustez ao processo.

```
#include <iostream>
#include <string>
using namespace std;
// Função para calcular o máximo divisor comum (mdc)
int mdc(int a, int b) {
   while (b != 0) {
      int temp = b;
      b = a % b;
      a = temp;
```

```
    }
    return a;
}
// Função de criptografia afim
string cifraAfim(const string& mensagem, int a, int b) {
    string resultado = "";
    int n = 26;
        if (mdc(a, n) != 1) {
        return "Erro: o valor de 'a' deve ser primo com 26 (mdc(a, 26) = 1).";
    }
    for (char letra : mensagem) {
        if (isalpha(letra)) {
            char base = isupper(letra) ? 'A' : 'a';
            int m = letra - base;
            int c = (a * m + b) % n;
            resultado += static_cast<char>(base + c);
        }
        else {
            resultado += letra;
        }
    }
        return resultado;
}
int main() {
    string mensagemOriginal;
    int a, b;
        cout << "=== Cifra Afim: c = (a * m + b) mod 26 ===" << endl;
    cout << "Digite a mensagem original: ";
    getline(cin, mensagemOriginal);
        cout << "Digite o valor de 'a' (deve ser primo com 26): ";
    cin >> a;
        cout << "Digite o valor de 'b' (deslocamento): ";
    cin >> b;
        string resultado = cifraAfim(mensagemOriginal, a, b);
    cout << "\nMensagem criptografada: " << resultado << endl;
        return 0;
}
```

Figura 19. Código−fonte da cifra afim com validação do MDC



Fonte: Elaborado pelo autor (2025)

A compilação pode ser feita com o atalho $\mathrm{Ctrl} + \mathrm{Alt} + \mathrm{B}$, sendo importante revisar e corrigir qualquer erro apontado pelo compilador antes da execução do código.

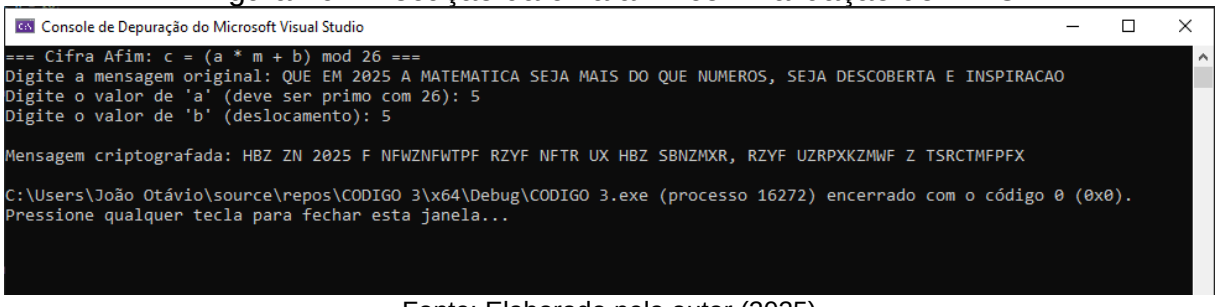
A aplicação prática da cifra foi realizada com os seguintes parâmetros:

Mensagem original: QUE EM 2025 A MATEMATICA SEJA MAIS DO QUE NUMEROS, SEJA DESCOBERTA E INSPIRACAO

Chave utilizada: $a = 5, b = 5$

Mensagem criptografada: HBZ ZN 2025 F NFWZFNFWTPF RZYF NFTR UX HBZ SBNZMXR, RZYF UZRPXKZMWF Z TSRCTMFPFX

Figura 20. Execução da cifra afim com validação do MDC



Fonte: Elaborado pelo autor (2025)

Este exemplo destaca a relevância de verificar previamente as condições matemáticas necessárias à aplicação segura de algoritmos criptográficos. A utilização do máximo divisor comum (MDC) como critério de validação introduz, de maneira aplicada, o conceito de primos relativos e abre espaço para a discussão sobre a existência de inversos modulares e sua importância na codificação.

A atividade também promove uma integração efetiva entre os conhecimentos matemáticos e computacionais, pois exige dos alunos não apenas a implementação do algoritmo, mas também a compreensão dos princípios teóricos que sustentam sua lógica. Essa abordagem estimula o desenvolvimento de competências como a resolução de problemas, o pensamento computacional e a análise crítica, sendo especialmente indicada em propostas pedagógicas voltadas à alfabetização científica e ao uso significativo da tecnologia.

Atividade 5. Codificando com Multiplicação no Sistema

A cifra multiplicativa é uma técnica criptográfica clássica baseada na multiplicação modular, aplicada sobre um alfabeto de tamanho n. Ao ser estendida para um conjunto expandido de símbolos, incluindo letras maiúsculas, números e sinais de pontuação, essa cifra possibilita a codificação de mensagens de maneira mais ampla e realista. A atividade tem por objetivo ampliar a compreensão dos estudantes sobre as exigências matemáticas para a reversibilidade da criptografia e demonstrar a aplicação prática desses conceitos na segurança da informação. O processo de criptografia é descrito pela relação:

$$c \equiv a \cdot m \pmod{n},$$

onde:

- c representa o caractere cifrado;
- m é a posição do caractere original no alfabeto expandido;
- a é a chave multiplicativa;
- n é o tamanho do alfabeto expandido.

A validade dessa cifra depende de que a seja coprimo com n, condição que assegura a existência de um inverso multiplicativo de a módulo n. Esse elemento é indispensável para a etapa de decifragem, que é realizada pela fórmula:

$$m \equiv a^{-1} \cdot c \pmod{n}$$

onde:

- a^{-1} é o inverso multiplicativo de a em relação a n.

Dessa maneira, garante-se a reversibilidade do processo e a recuperação fiel da mensagem original. A cifra multiplicativa, portanto, evidencia de forma simples e elegante a aplicação da aritmética modular no campo da criptografia clássica.

Procedimentos e Implementação: No ambiente Visual Studio, foi desenvolvido um código em C++ que implementa a cifra multiplicativa com alfabeto expandido, incluindo validação do MDC para garantir a chave adequada.

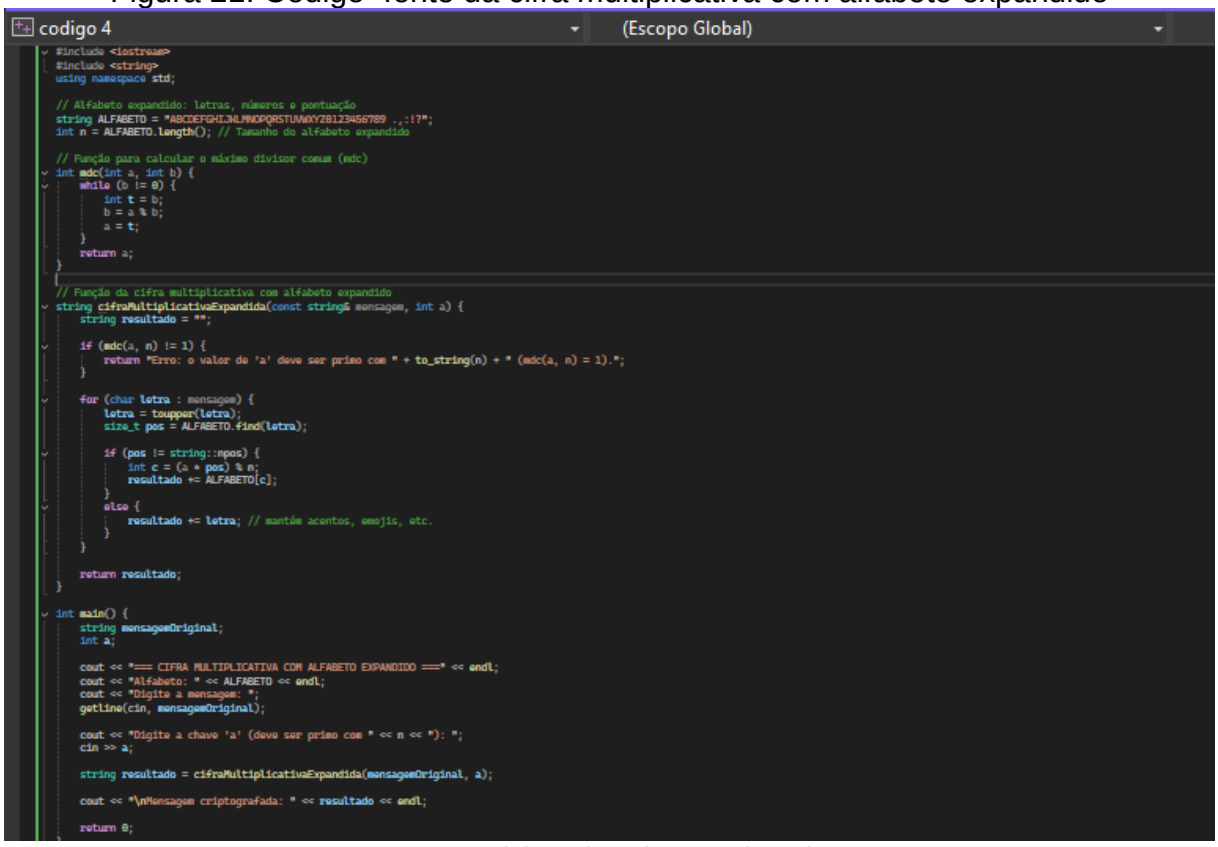
```
#include <iostream>
#include <string>
using namespace std;
// Alfabeto expandido: letras, números e pontuação
string ALFABETO = "ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789 .,:!?";
int n = ALFABETO.length(); // Tamanho do alfabeto expandido
// Função para calcular o máximo divisor comum (mdc)
int mdc(int a, int b) {
    while (b != 0) {
        int t = b;
        b = a % b;
        a = t;
    }
    return a;
}
// Função da cifra multiplicativa com alfabeto expandido
string cifraMultiplicativaExpandida(const string& mensagem, int a) {
    string resultado = "";
    if (mdc(a, n) != 1) {
        return "Erro: o valor de 'a' deve ser primo com " + to_string(n) + " (mdc(a, n) = 1).";
    }
    for (char letra : mensagem) {
        letra = toupper(letra);
        size_t pos = ALFABETO.find(letra);

        if (pos != string::npos) {
            int c = (a * pos) % n;
            resultado += ALFABETO[c];
        }
        else {
            resultado += letra; // mantém acentos, emojis, etc.
```

```
        }
    }
    return resultado;
}
int main() {
    string mensagemOriginal;
    int a;
    cout << "=== CIFRA MULTIPLICATIVA COM ALFABETO EXPANDIDO ===" << endl;
    cout << "Alfabeto: " << ALFABETO << endl;
    cout << "Digite a mensagem: ";
    getline(cin, mensagemOriginal);
    cout << "Digite a chave 'a' (deve ser primo com " << n << "): ";
    cin >> a;
    string resultado = cifraMultiplicativaExpandida(mensagemOriginal, a);
    cout << "\nMensagem criptografada: " << resultado << endl;
    return 0;
                }
```

O alfabeto utilizado é o mesmo das cifras anteriores, contendo 44 caracteres:

Figura 21. Código−fonte da cifra multiplicativa com alfabeto expandido



Fonte: Elaborado pelo autor (2025)

Após digitar o código, o aluno deve realizar a compilação pressionando $\mathrm{Ctrl} + \mathrm{Alt} + \mathrm{B}$, identificando possíveis erros de digitação ou sintaxe. O programa está configurado para validar automaticamente a chave inserida, impedindo a execução em caso de valor inválido de a.

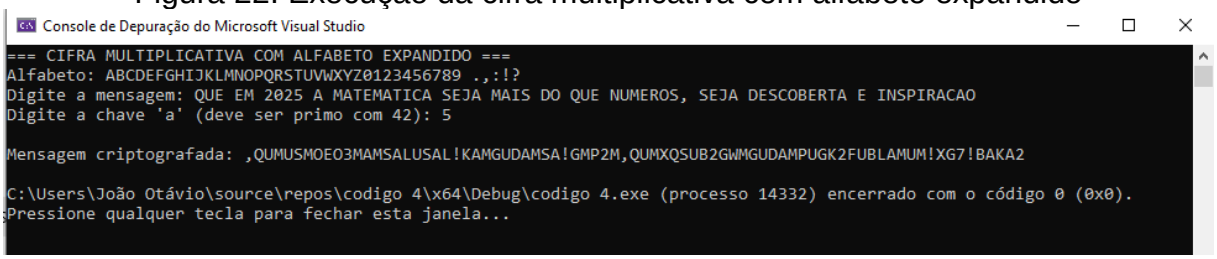
Na execução prática, foi utilizada uma chave válida $\mathrm{a} = 5$, respeitando a condição de coprimalidade com $\mathrm{n} = 44$, resultando na seguinte cifragem:

Mensagem original: QUE EM 2025 A MATEMATICA SEJA MAIS DO QUE NUMEROS, SEJA DESCOBERTA E INSPIRACAO

Chave utilizada: $\mathrm{a} = 5$

Mensagem criptografada: ,QUMUSMOEO3MAMSALUSAL!KAMGUDAMSA!GMP2M,QUMXQSUB2GWMGUDAMPUGK2FUBLAMUM!XG7!BAKA2

Figura 22. Execução da cifra multiplicativa com alfabeto expandido



Fonte: Elaborado pelo autor (2025)

A cifra multiplicativa com alfabeto expandido permite aplicar, de forma prática, conceitos como inverso modular, coprimalidade e congruência, evidenciando a estrutura dos grupos multiplicativos. Ao incluir letras, números e pontuação, aproxima o conteúdo matemático das aplicações reais em segurança digital.

Além disso, a atividade estimula a integração entre matemática e programação, desenvolvendo o raciocínio algorítmico e a análise crítica dos parâmetros. Ao experimentar diferentes valores de chave, os estudantes compreendem o impacto direto dessas variações na decodificação, fortalecendo sua intuição e compreensão dos sistemas criptográficos.

Atividade 6. Palavras como Chave na Cifra de Vigenère

A cifra de Vigenère representa uma evolução em relação às cifras monoalfabéticas, como a de César, por utilizar uma chave composta por letras que se

repete ciclicamente ao longo da mensagem. Cada letra da chave define um deslocamento distinto, o que torna a criptografia mais robusta e menos suscetível à análise de frequência. A atividade permite que os estudantes visualizem a aplicação de chaves alfabéticas para modificar múltiplas letras de uma mensagem com diferentes deslocamentos, promovendo o desenvolvimento de habilidades em matemática, lógica e programação. Sua fórmula matemática é:

$$c_i \equiv m_i + k_i \pmod{26}$$

onde:

- c_i é o caractere cifrado na posição i;
- m_i é o valor da letra original na posição i;
- k_i é o valor da letra correspondente da chave;
- o cálculo é realizado módulo 26, correspondente ao alfabeto.

A decifragem segue a operação inversa, subtraindo o valor da chave em cada posição:

$$m_i \equiv c_i - k_i \pmod{26}$$

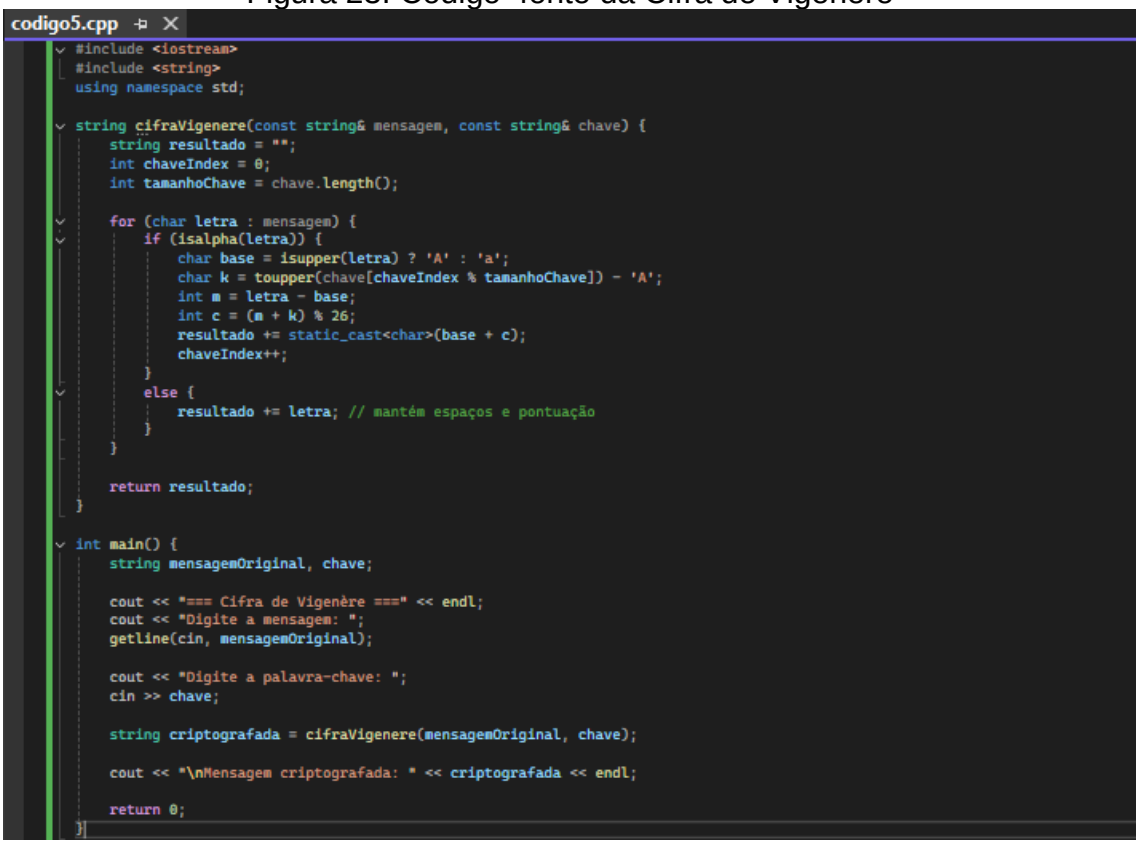
Esse método garante que a mensagem original seja recuperada integralmente. O uso de diferentes deslocamentos em uma mesma palavra confere à cifra de Vigenère maior resistência em comparação às cifras de substituição simples, ilustrando de maneira didática a força da aritmética modular aplicada à criptografia.

A cifra foi implementada em C++ no ambiente Visual Studio, utilizando funções que tratam a repetição da palavra−chave, a conversão das letras em valores numéricos e a aplicação da cifra apenas sobre letras, mantendo caracteres não alfabéticos inalterados.

```
#include <iostream>
#include <string>
using namespace std;
string cifraVigenere(const string& mensagem, const string& chave) {
    string resultado = "";
    int chaveIndex = 0;
    int tamanhoChave = chave.length();
    for (char letra : mensagem) {
        if (isalpha(letra)) {
            char base = isupper(letra) ? 'A' : 'a';
```

```
            char k = toupper(chave[chaveIndex % tamanhoChave]) - 'A';
            int m = letra - base;
            int c = (m + k) % 26;
            resultado += static_cast<char>(base + c);
            chaveIndex++;
        }
        else {
            resultado += letra; // mantém espaços e pontuação
        }
    }
    return resultado;
}
int main() {
    string mensagemOriginal, chave;
    cout << "=== Cifra de Vigenère ===" << endl;
    cout << "Digite a mensagem: ";
    getline(cin, mensagemOriginal);
    cout << "Digite a palavra-chave: ";
    cin >> chave;
    string criptografada = cifraVigenere(mensagemOriginal, chave);
    cout << "\nMensagem criptografada: " << criptografada << endl;
    return 0;
}
```

Figura 23. Código-fonte da Cifra de Vigenére



Fonte: Elaborado pelo autor (2025)

A compilação pode ser realizada com o atalho $\mathrm{Ctrl} + \mathrm{Alt} + \mathrm{B}$**.** É importante orientar os alunos sobre o uso de letras maiúsculas sem acentos tanto na mensagem quanto na chave, para garantir o funcionamento correto da cifra.

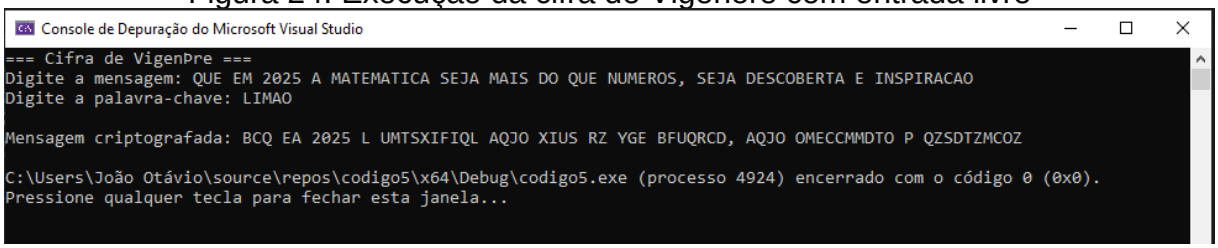
A cifra foi aplicada com a palavra−chave "LIMAO", resultando na seguinte transformação criptográfica:

Mensagem original: QUE EM 2025 A MATEMATICA SEJA MAIS DO QUE NUMEROS, SEJA DESCOBERTA E INSPIRACAO

Palavra−chave utilizada: LIMAO

Mensagem criptografada: *BCQ EA 2025 L UMTSXIFIQL AQJO XIUS RZ YGE BFUQRCD, AQJO OMECCMMDTO P QZSDTZMCOZ*

Figura 24. Execução da cifra de Vigenère com entrada livre



Fonte: Elaborado pelo autor (2025)

A cifra de Vigenère introduz o conceito de múltiplos deslocamentos controlados por uma chave cíclica, ampliando a compreensão dos alunos sobre sistemas criptográficos. Sua aplicação envolve operações modulares e manipulação de *strings*, contribuindo para o desenvolvimento do pensamento computacional. A atividade também favorece a interdisciplinaridade ao dialogar com os conteúdos de Língua Portuguesa, História e Informática. Ao lidar com entradas livres e criptografia polialfabética, os estudantes enfrentam desafios reais da segurança digital, refletindo sobre estratégias de proteção e de exposição de informações.

5.1 Comparação de Métodos de Criptografia

Para ilustrar as diferenças entre diversos métodos de criptografia, foi aplicado quatro técnicas distintas à mesma mensagem original, relacionadas as atividades 3, 4, 5 e 6. Abaixo, apresentamos os resultados e uma breve análise de cada método.

Mensagem Original: QUE EM 2025 A MATEMÁTICA SEJA MAIS DO QUE NÚMEROS, SEJA DESCOBERTA E INSPIRAÇÃO

- Atividade 3. Expandindo o Código com a Criptografia Aditiva
- Chave utilizada: Deslocamento de 5 letras
- Mensagem criptografada: VZJ?JR?757??F?RFYJRFYNHF?XJOF ?RFNX?IT?VZJ?SZRJWTXB?XJOF?IJXHTGJWYF?J?NSXUNWFHFT

- Atividade 4. Função Afim em Ação: Codificando com Critério de MDC
- Chave utilizada: $a = 5, b = 5$
- Mensagem criptografada: HBZ ZN 2025 F NFWZFNFWTPF RZYF NFTR UX HBZ SBNZMXR, RZYF UZRPXKZMWF Z TSRCTMFPFX

- Atividade 5. Codificando com Multiplicação
- Chave utilizada: $a = 5$
- Mensagem criptografada: ,QUMUSMOEO3MAMSALUSAL!KAMGUDAMSA!GM P2M,QUMXQSUB2GWMGUDAMPUGK2FUBLAMUM!XG7!BAKA2

- Atividade 6. Palavras como Chave na Cifra de Vigenère
- Palavra−chave utilizada: LIMÃO
- Mensagem criptografada: BCQ EA 2025 L UMTSXIFIQL AQJO XIUS RZ YGE BFUQRCD, AQJO OMECCMMDTO P QZSDTZMCOZ

A aplicação de diferentes métodos de criptografia à mesma mensagem permite observar como cada técnica transforma o texto original de maneira única, refletindo suas características específicas de segurança e complexidade.

Tabela 11. Mensagens Criptografadas por Método

Atividade	Mensagem Criptografada
3.	VZJ?JR?757??F?RFYJRFYNHF?XJOF?RFNX?IT?VZJ?SZRJWTXB?XJOF?IJXHTGJWYF?J?NSXUNWFHFT
4.	HBZ ZN 2025 F NFWZFNFWTPF RZYF NFTR UX HBZ SBNZMXR, RZYF UZRPXKZMWF Z TSRCTMFPFX

5.	,QUMUSMOEO3MAMSALUSAL!KAMGUDAMSA!GMP2M,QUMXQ SUB2GWMGUDAMPUGK2FUBLAMUM!XG7!BAKA2
6.	BCQ EA 2025 L UMTSXIFIQL AQJO XIUS RZ YGE BFUQRCD, AQJO OMECCMMDTO P QZSDTZMCOZ

Fonte: Elaborado pelo autor (2025)

Com base em autores como Stallings (2006), Coutinho (2009) e Hefez (2022), é possível classificar diferentes métodos de cifragem quanto à sua estrutura algébrica, segurança relativa e complexidade de implementação. Cada cifra apresenta um padrão distinto de codificação, evidenciando as variações na segurança e na complexidade de implementação. Enquanto a Cifra de César oferece uma substituição simples, a Cifra de Vigenère utiliza uma palavra−chave para criar uma substituição polialfabética, aumentando a resistência a ataques de frequência. A Cifra Afim introduz uma função matemática linear, e a cifra multiplicativa aplica transformações específicas, demonstrando como diferentes abordagens podem ser aplicadas para proteger informações sensíveis.

Tabela 12. Comparação de Segurança e Complexidade

Método	Tipo de Cifra	Segurança Relativa	Complexidade de Implementação
Cifra de César	Substituição Monoalfabética	Baixa	Baixa
Cifra Afim	Substituição Monoalfabética	Baixa	Média
Cifra Multiplicativa	Substituição com Função Modular	Baixa	Média
Cifra de Vigenère	Substituição Polialfabética	Média	Média

Fonte: Adaptado de Stallings (2006), Coutinho (2009) e Hefez (2022), com base em análise das características estruturais e pedagógicas das cifras.

A aplicação de diferentes métodos de criptografia à mesma mensagem evidencia as variações em termos de segurança e complexidade. Enquanto cifras simples como a de César e a Afim são fáceis de implementar, oferecem baixa proteção. Técnicas mais avançadas, como a Cifra de Vigenère, neste contexto didático, proporciona maior segurança, mas exigem cuidados adicionais na implementação e gestão de chaves. A escolha do método apropriado deve considerar o nível de segurança desejado e os recursos disponíveis para implementação.

As atividades práticas com criptografia descritas nas seções anteriores oferecem um campo fértil para a elaboração de exercícios e avaliações que promovam não apenas a fixação dos conteúdos, mas também o desenvolvimento de habilidades

como raciocínio lógico, abstração matemática, interpretação algorítmica e aplicação de conceitos teóricos em situações reais.

Cabe ao professor adaptar e diversificar os exercícios de acordo com o perfil da turma, o nível de ensino e os recursos disponíveis, priorizando o protagonismo discente, a colaboração e a contextualização das atividades. A avaliação, mais do que medir acertos, deve ser entendida como processo formativo, capaz de revelar avanços, dificuldades e estratégias utilizadas pelos alunos ao lidarem com desafios reais da matemática.

CAPÍTULO 6 – RELATO DA APLICAÇÃO DIDÁTICA: CRIPTOGRAFIA NA SALA DE AULA

Com o intuito de avaliar a viabilidade e a eficácia das propostas pedagógicas desenvolvidas ao longo desta dissertação, foi realizada a aplicação prática das atividades didáticas de criptografia em duas instituições de ensino na cidade de Votuporanga/SP, identificadas como Escola X e Escola Y. Essa etapa teve como objetivo observar a receptividade dos alunos, os desafios enfrentados em sala de aula e as possibilidades de integração entre a Matemática, a tecnologia e temas contemporâneos, como a segurança digital.

A primeira aplicação ocorreu na Escola X, onde as atividades foram desenvolvidas com estudantes do 9º ano do Ensino Fundamental, durante as aulas de Matemática, envolvendo a participação de 18 alunos. A proposta foi realizada ao longo de três aulas, cada uma com duração de 50 minutos.

Figura 25. Apresentação do professor na Escola X

Fonte: Arquivo pessoal do autor (2025)

A segunda aplicação ocorreu na Escola Y, onde as atividades foram realizadas com estudantes do 1º ano do Ensino Médio, totalizando 38 estudantes participantes. A proposta foi organizada em quatro aulas, cada uma com duração de 50 minutos.

Figura 26. Apresentação do professor na Escola Y

Fonte: Arquivo pessoal do autor (2025)

A metodologia adotada para a aplicação das atividades buscou integrar teoria e prática de maneira dinâmica e contextualizada, utilizando diferentes recursos didáticos. Entre eles, destacaram−se o criptodisco físico, slides expositivos sobre a evolução histórica da criptografia, lousa branca, projetor multimídia e uma folha orientadora com exercícios de codificação e decodificação, elaborada especialmente para a proposta. Na Escola X, as atividades foram desenvolvidas e organizadas em três momentos principais: uma introdução teórica, uma fase de experimentação prática e a resolução de desafios em grupo.

Figura 27. Alunos do 9º ano utilizando o criptodisco em grupo

Fonte: Arquivo pessoal do autor (2025)

A atividade prática, intitulada “Códigos Secretos com a Roda de Criptografia”, teve como objetivo proporcionar aos estudantes uma vivência concreta do funcionamento da Cifra de César, por meio da manipulação do criptodisco.

Organizados em grupos, os alunos escreveram mensagens, escolheram uma chave de deslocamento entre 1 e 39 e ajustaram a roda para realizar a codificação. Em seguida, trocaram suas mensagens com outros grupos, que deveriam decifrá−las com base na chave recebida ou deduzida por tentativa e análise de padrões. A proposta favoreceu o raciocínio lógico, a colaboração entre os pares e a aplicação da aritmética modular em um contexto lúdico e significativo.

Figura 28. Trabalho em grupo durante a codificação das mensagens

Fonte: Arquivo pessoal do autor (2025)

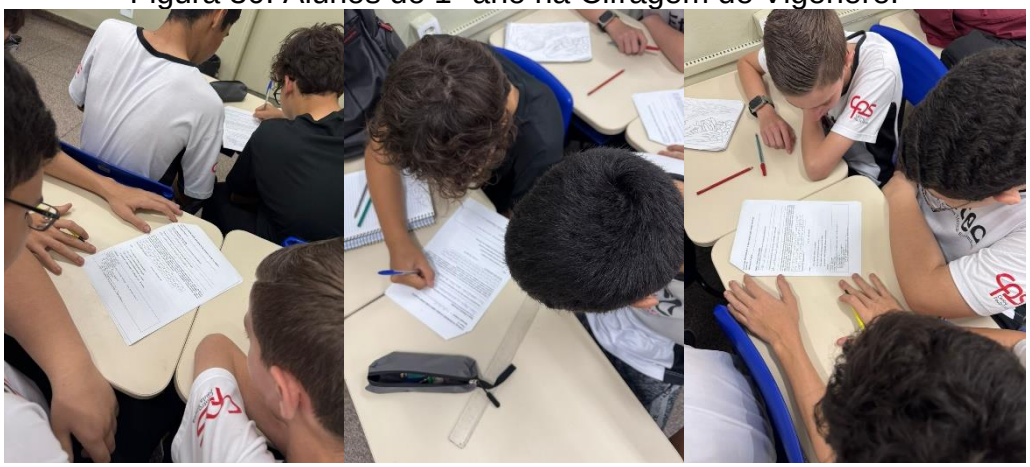
Na Escola Y, a atividade começou com uma apresentação teórica, nos mesmos moldes daquela aplicada na Escola X. Em seguida, os alunos participaram de uma experimentação prática com o criptodisco, organizados em grupos e conduzidos por uma lista de exercícios e desafios propostos.

Figura 29. Alunos do 1º ano utilizando o criptodisco em grupo

Fonte: Arquivo pessoal do autor (2025)

No entanto, com o objetivo de aprofundar os conceitos trabalhados, os alunos do Ensino Médio participaram de uma segunda etapa, voltada à codificação de mensagens com a Cifra de Vigenère, utilizando a tabela alfabética em que A = 0, B = 1, C = 2, e assim por diante até Z = 25. Com o auxílio da folha orientadora de exercícios, os estudantes aplicaram os conceitos de aritmética modular para realizar a criptografia das mensagens, fortalecendo a conexão entre a matemática e os fundamentos lógicos presentes na programação.

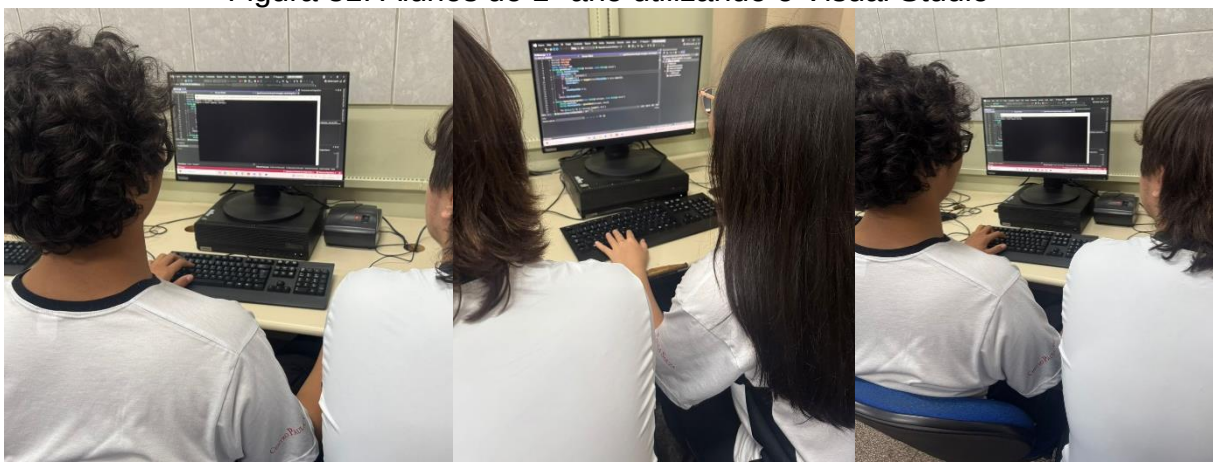
Figura 30. Alunos do 1º ano na Cifragem de Vigenère.



Fonte: Arquivo pessoal do autor (2025)

Na sequência, a aula avançou para a atividade computacional com a Cifra de Vigenère, conforme descrita na Atividade 6 desta dissertação, utilizando o ambiente de desenvolvimento Microsoft Visual Studio. Nessa etapa, os alunos puderam experimentar a implementação prática da cifra por meio da programação, comparando os resultados obtidos manualmente com os gerados pelo código. Essa vivência reforçou a compreensão dos algoritmos e evidenciou a relação entre a matemática modular e a lógica computacional envolvida na criptografia.

Figura 31. Alunos do 1º ano utilizando o Visual Studio

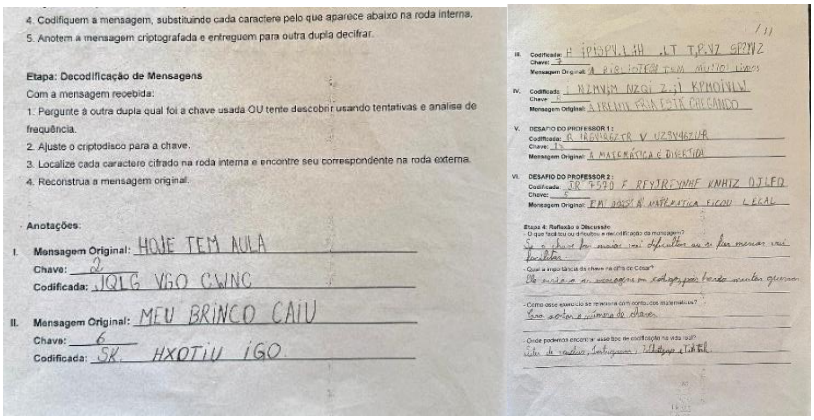


Fonte: Arquivo pessoal do autor (2025)

Ao final das atividades, os alunos foram convidados a participar de uma reflexão coletiva orientada, com o intuito de retomar os conceitos explorados, identificar aprendizagens e promover conexões com a matemática e o cotidiano. Na Escola X, os estudantes responderam a um conjunto de questões relacionadas à atividade com a Cifra de César. Quando questionados sobre o que facilitou ou dificultou a decodificação da mensagem, destacaram elementos como a variação da chave e os padrões visuais na escrita: *“Se a chave for maior vai dificultar, se for menor vai facilitar” e “A lógica e o padrão das letras.”*

Ao refletirem sobre a importância da chave na Cifra de César, as respostas indicaram o reconhecimento do papel da chave como elemento central do processo de codificação: *“Ter uma base para decodificar com facilidade” e “Ele enviava as mensagens em códigos, pois havia muitas guerras.”*

Figura 32. Registro da atividade desenvolvida − Grupo 1

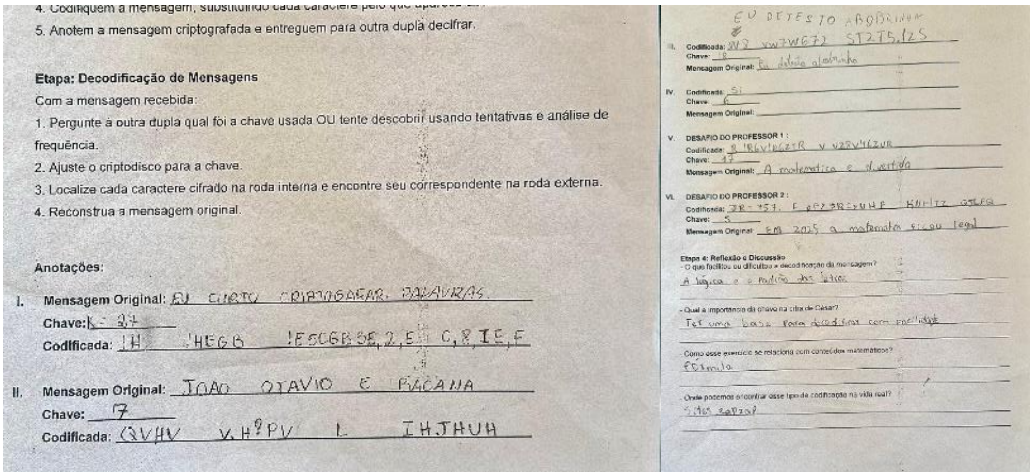


Fonte: Arquivo pessoal do autor (2025)

Sobre a relação da atividade com a matemática, os alunos citaram de forma espontânea elementos estruturais: *“Fórmulas e operação de adição e subtração”* e *“Para contar o número de chaves.”*

Por fim, ao serem convidados a pensar onde esse tipo de codificação poderia ser encontrado na vida real, apontaram exemplos diversos do cotidiano e da cultura digital: *“Sites, WhatsApp, caça ao tesouro, jogos de espiões”* e *“Jogos como Assassin’s Creed, Tomb Raider e The Room, que fazem uso de cifras simples em seus enigmas.”*

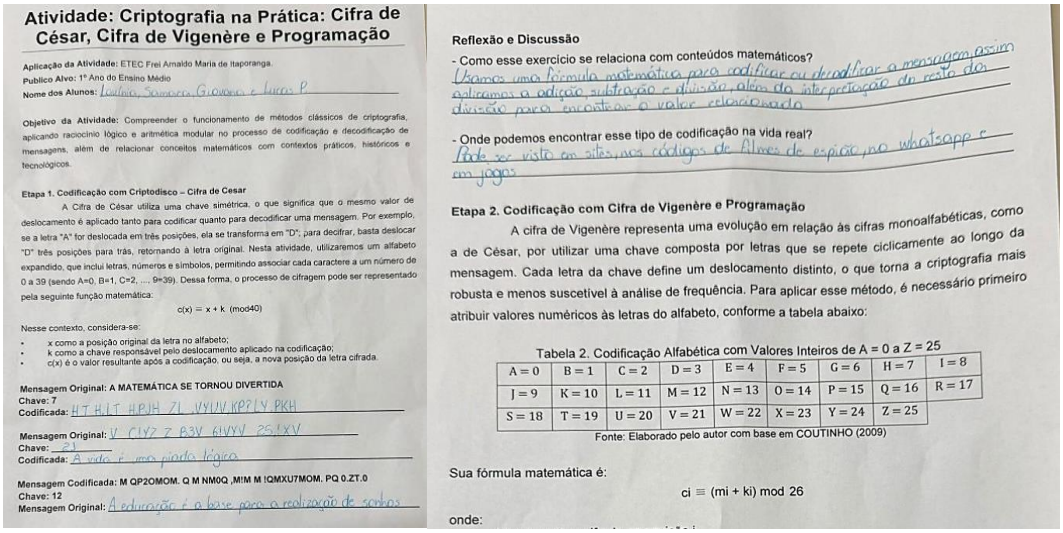
Figura 33. Registro da atividade desenvolvida − Grupo 2



Fonte: Arquivo pessoal do autor (2025)

Já na Escola Y, as reflexões foram direcionadas às atividades com a Cifra de Vigenère e sua implementação no Microsoft Visual Studio. Sobre a relação com conteúdos matemáticos, um dos estudantes respondeu: *“Usamos uma fórmula matemática para codificar ou decodificar a mensagem, assim aplicamos a adição, subtração e divisão, além da interpretação do resto da divisão para encontrar o valor relacionado.”*

Figura 34. Registro da atividade desenvolvida na Escola Y – Parte 1



Atividade: Criptografia na Prática: Cifra de César, Cifra de Vigenère e Programação

Aplicação da Atividade: ETEC Frei Arnaldo Maria de Itaporanga.
Publico Alvo: 1º Ano do Ensino Médio
Nome dos Alunos: Laiana, Sandra, Giovana e Lucas P.

Objetivo da Atividade: Compreender o funcionamento de métodos clássicos de criptografia, aplicando raciocínio lógico e aritmética modular no processo de codificação e decodificação de mensagens, além de relacionar conceitos matemáticos com contextos práticos, históricos e tecnológicos.

Etapa 1. Codificação com Criptodisco – Cifra de Cesar

A Cifra de César utiliza uma chave simétrica, o que significa que o mesmo valor de deslocamento é aplicado tanto para codificar quanto para decodificar uma mensagem. Por exemplo, se a letra "A" for deslocada em três posições, ela se transforma em "D"; para decifrar, basta deslocar "D" três posições para trás, retornando à letra original. Nesta atividade, utilizaremos um alfabeto expandido, que inclui letras, números e símbolos, permitindo associar cada caractere a um número de 0 a 39 (sendo A=0, B=1, C=2, ..., 9=39). Dessa forma, o processo de cifragem pode ser representado pela seguinte função matemática:

$$c(x) = x + k \pmod{40}$$

Nesse contexto, considera-se:

- x como a posição original da letra no alfabeto;
- k como a chave responsável pelo deslocamento aplicado na codificação;
- c(x) é o valor resultante após a codificação, ou seja, a nova posição da letra cifrada.

Mensagem Original: A MATEMÁTICA SE TORNOU DIVERTIDA
Chave: 7
Codificada: H J H L T H P J H ZL .VVUV.KP?LY.PKH

Mensagem Original: V C1Y2 Z B3V 61VYV 25.!XV
Chave: 21
Codificada: A vida é uma piada lógica

Mensagem Codificada: M QP2OMOM. Q M NM0Q ,M!M M !QMXU7MOM. PQ 0.ZT.0
Chave: 12
Mensagem Original: A educação é a base para a realização de sonhos

Reflexão e Discussão

- Como esse exercício se relaciona com conteúdos matemáticos?
Usamos uma fórmula matemática para codificar ou decodificar a mensagem, assim aplicamos a adição, subtração e divisão, além da interpretação do resto da divisão para encontrar o valor relacionado.

- Onde podemos encontrar esse tipo de codificação na vida real?
Pode ser visto em sites, nos códigos de Alves de espião, no whatsapp e em jogos

Etapa 2. Codificação com Cifra de Vigenère e Programação

A cifra de Vigenère representa uma evolução em relação às cifras monoalfabéticas, como a de César, por utilizar uma chave composta por letras que se repete ciclicamente ao longo da mensagem. Cada letra da chave define um deslocamento distinto, o que torna a criptografia mais robusta e menos suscetível à análise de frequência. Para aplicar esse método, é necessário primeiro atribuir valores numéricos às letras do alfabeto, conforme a tabela abaixo:

Tabela 2. Codificação Alfabética com Valores Inteiros de A = 0 a Z = 25

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

Fonte: Elaborado pelo autor com base em COUTINHO (2009)

Sua fórmula matemática é:

$$ci \equiv (mi + ki) \bmod 26$$

onde:

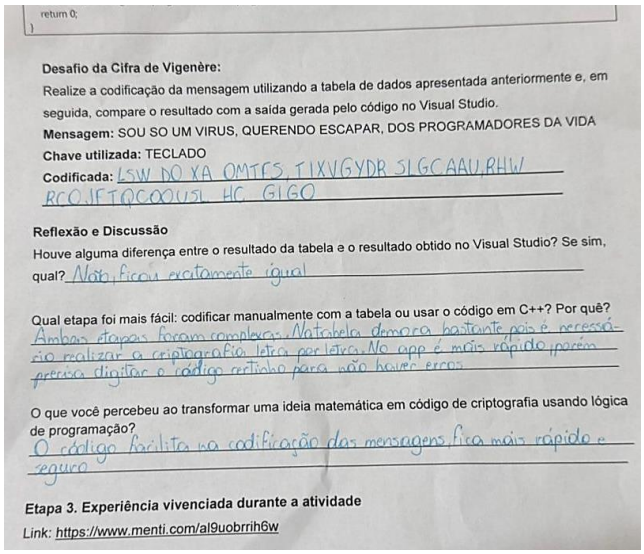
Fonte: Arquivo pessoal do autor (2025)

Ao compararem os resultados obtidos manualmente com a tabela e os gerados pelo código em C++, os alunos relataram: *“Ficou exatamente igual.”*

Sobre qual etapa foi mais fácil, alguns reconheceram a diferença entre as abordagens, destacando as particularidades de cada uma: *“Ambas etapas foram complexas. Na tabela demora bastante, pois é necessário realizar a criptografia letra por letra. No app é mais rápido, porém precisa digitar o código certinho para não haver erros.”*

Por fim, ao refletirem sobre a experiência de transformar uma ideia matemática em código de criptografia utilizando lógica de programação, um grupo sintetizou: *“O código facilita na codificação das mensagens, fica mais rápido e seguro.”*

Figura 35. Registro da atividade desenvolvida na Escola Y – Parte 2



return 0;
}

Desafio da Cifra de Vigenère:
Realize a codificação da mensagem utilizando a tabela de dados apresentada anteriormente e, em seguida, compare o resultado com a saída gerada pelo código no Visual Studio.
Mensagem: SOU SO UM VIRUS, QUERENDO ESCAPAR, DOS PROGRAMADORES DA VIDA
Chave utilizada: TECLADO
Codificada: LSW DO XA OMTFS, TIXVGYDR SLGCAAU,RHW RCOJFTQCOOUSL HC GIGO

Reflexão e Discussão
Houve alguma diferença entre o resultado da tabela e o resultado obtido no Visual Studio? Se sim, qual? Não, ficou exatamente igual

Qual etapa foi mais fácil: codificar manualmente com a tabela ou usar o código em C++? Por quê?
Ambas etapas foram complexas. Na tabela demora bastante pois é necessário realizar a criptografia letra por letra. No app é mais rápido porém precisa digitar o código certinho para não haver erros

O que você percebeu ao transformar uma ideia matemática em código de criptografia usando lógica de programação?
O código facilita na codificação das mensagens, fica mais rápido e seguro

Etapa 3. Experiência vivenciada durante a atividade
Link: https://www.menti.com/al9uobrrih6w

Fonte: Arquivo pessoal do autor (2025)

Durante a execução da atividade, alguns alunos expressaram dúvidas quanto à necessidade do uso de fórmulas matemáticas durante a atividade, verbalizando questionamentos como: *“Qual a necessidade de colocar fórmula?”.* Para contornar esse obstáculo, foi realizada uma explicação mais acessível sobre a importância da linguagem matemática como forma de garantir a confiabilidade dos sistemas criptográficos, acompanhada de exemplos concretos e contextualizados. Esse momento foi marcado por reações espontâneas dos estudantes, como: *“Nunca imaginei que funcionava assim”* e “*Isso desperta muita curiosidade em saber o que está escrito”*. Tais comentários indicam que, apesar de uma resistência inicial, a proposta contribuiu para despertar interesse, surpresa e engajamento cognitivo, promovendo uma aproximação significativa entre os conteúdos matemáticos e o cotidiano dos alunos.

Para encerrar a proposta de maneira interativa e participativa, foi realizada uma atividade digital utilizando a plataforma Mentimeter. Os alunos foram convidados a acessar o link disponibilizado (https://www.menti.com/al9uobrrih6w) e registrar palavras que melhor representassem a experiência vivenciada. O resultado foi a construção coletiva de uma nuvem de palavras, que evidenciou termos como “segurança”, “mensagem secreta”, “matemática” e “criptografia”. Essa etapa final promoveu o engajamento, favoreceu a reflexão coletiva e reforçou os conceitos abordados de forma visual, afetiva e significativa.

Figura 36. Resultado da chuva de palavras

Fonte: https://www.menti.com/al9uobrrih6w

A aplicação prática evidenciou que a criptografia pode ser uma ferramenta poderosa para o ensino de Matemática, ao promover interdisciplinaridade, desenvolvimento do raciocínio lógico e maior envolvimento dos alunos. A atividade com o criptodisco e as cifras clássicas mostrou−se acessível e estimulante para os estudantes do Ensino Fundamental, enquanto a utilização do Visual Studio e da Cifra de Vigenère com os alunos do Ensino Médio possibilitou uma abordagem mais aprofundada, integrando conteúdos de Matemática, tecnologia e pensamento computacional.

Esse relato reforça a proposta desta dissertação de integrar teoria e prática, proporcionando aos alunos experiências significativas de aprendizagem por meio de desafios concretos e relacionados a contextos atuais. Durante a atividade, os alunos demonstraram entusiasmo e curiosidade pelo tema, especialmente nas discussões sobre segurança digital, como o uso de senhas, criptografia em aplicativos de mensagens e a proteção de dados bancários. A participação foi intensa e colaborativa, principalmente na etapa de construção coletiva das mensagens criptografadas, evidenciando o potencial da proposta para aproximar os conteúdos escolares da realidade dos estudantes.

CONCLUSÃO

O presente trabalho buscou investigar o potencial didático da criptografia como recurso de ensino da Matemática nos anos finais do Ensino Fundamental e no Ensino Médio. A partir da questão central sobre de que forma algoritmos criptográficos podem contribuir para o desenvolvimento do raciocínio lógico e para o engajamento dos estudantes, foi possível articular a teoria dos números, a aritmética modular e conceitos de programação em uma proposta pedagógica inovadora e alinhada às demandas atuais da educação.

A revisão histórica e teórica evidenciou que a criptografia sempre esteve ligada ao avanço matemático, desde a simples transposição de letras na Antiguidade até algoritmos robustos como RSA e AES. Essa trajetória mostra que a Matemática, além de ciência abstrata, possui aplicações concretas e indispensáveis à vida em sociedade. O estudo da divisibilidade, do algoritmo de Euclides, dos números primos e dos teoremas de Fermat e Euler revelou−se fundamental para compreender o funcionamento das cifras, confirmando o papel instrumental da Matemática no desenvolvimento da segurança digital.

As atividades aplicadas em sala de aula reforçaram a conexão entre cálculo e prática. O uso do criptodisco, da cifra de César e da cifra de Vigenère permitiu aos alunos exercitar operações aritméticas, aplicar propriedades da aritmética modular e perceber como fórmulas matemáticas sustentam processos de codificação e decodificação. A etapa computacional com o Visual Studio ampliou essa experiência, ao possibilitar que os estudantes transformassem cálculos manuais em algoritmos programados, fortalecendo a compreensão de que cada operação matemática é a base lógica de um processo criptográfico.

Além de despertar curiosidade, a criptografia mostrou−se um recurso pedagógico eficaz para desenvolver competências matemáticas essenciais, como resolução de problemas, análise de padrões e argumentação baseada em cálculos. Ao lidar com cifras, os estudantes foram estimulados a pensar matematicamente, testar hipóteses e verificar resultados, exercitando o raciocínio lógico de maneira significativa. Isso confirma que, quando contextualizada, a Matemática deixa de ser percebida apenas como um conjunto de regras abstratas e passa a ser compreendida como ferramenta útil, dinâmica e interdisciplinar.

Conclui−se que a criptografia pode atuar como estratégia pedagógica inovadora, capaz de enriquecer o ensino de Matemática e de contribuir para a formação de estudantes mais críticos, criativos e preparados para os desafios da sociedade digital, sem perder de vista a Matemática como fundamento essencial de todo o processo.

REFERÊNCIAS

ANSI X3.4−1986. American National Standard Code for Information Interchange. Washington, D.C.: American National Standards Institute, 1986.

BARBOSA, V. C. Fundamentos da Matemática Discreta. 2. ed. Rio de Janeiro: LTC, 2017.

BORBA, M. C.; PENTEADO, M. G. Educação Matemática e Tecnologias Digitais: interfaces com a prática docente. Belo Horizonte: Autêntica, 2016.

BRASIL. Ministério da Educação. Base Nacional Comum Curricular. Brasília: MEC, 2018. Disponível em: <http://basenacionalcomum.mec.gov.br>. Acesso em: [coloque a data de acesso].

CARNEIRO, A. História da criptografia: das cifras antigas à segurança da informação. São Paulo: Érica, 2017.

CLAYBOURNE, Anna. 91 truques matemáticos legais. São Paulo: Pé da Letra, 2021.

COUTINHO, S. C. A matemática e os números secretos: uma introdução à criptografia. 2. ed. Rio de Janeiro: Zahar, 2009.

GAUSS, C. F. Disquisitiones Arithmeticae. Leipzig: Gerh. Fleischer, 1801.

GOLDSTEIN, L.; LAY, D. C.; SCHNEIDER, D. I. Calculus and its applications. Upper Saddle River: Pearson Prentice Hall, 2007.

GTA – Grupo de Teleinformática e Automação (UFRJ). Decifrando Textos em Português. Disponível em: <http://www.gta.ufrj.br/ensino/eel878/frequencia.html>. Acesso em: [coloque a data de acesso].

HARDY, G. H.; WRIGHT, E. M. An Introduction to the Theory of Numbers. 5. ed. Oxford: Oxford University Press, 1979.

HEFEZ, A. Teoria dos Números: uma introdução. Rio de Janeiro: SBM, 2022.
HERÓDOTO. História. Livro V. Tradução de Mário da Gama Kury. Brasília: Editora da UnB, [s.d.].

KAHN, D. The Codebreakers: the comprehensive history of secret communication from ancient times to the internet. New York: Scribner, 1995.

MENEZES, A. J.; VAN OORSCHOT, P. C.; VANSTONE, S. A. Handbook of Applied Cryptography. Boca Raton: CRC Press, 1996.

NIVEN, I.; ZUCKERMAN, H. S.; MONTGOMERY, H. L. An Introduction to the Theory of Numbers. 5. ed. New York: Wiley, 2004.

OLIVEIRA, J. P. Introdução à Teoria dos Números. São Paulo: Livraria da Física, 2007.

PAPERT, S**.** *Mindstorms: children, computers, and powerful ideas.* New York: Basic Books, 1980.

SINGH, S. O livro dos códigos: a história da criptografia, da Antiguidade à era da informática. Tradução de Alyda Faber. Rio de Janeiro: Record, 2007.

STALLINGS, W. Criptografia e segurança de redes: princípios e práticas. 4. ed. São Paulo: Pearson, 2006.

STEWART, I. Os números da natureza: os segredos da matemática que governa o universo. Rio de Janeiro: Zahar, 2015.

TANENBAUM, A. S.; WETHERALL, D. J. Redes de computadores. 5. ed. São Paulo: Pearson, 2011.

GLOSSÁRIO

Aritmética Modular: Ramo da matemática que trabalha com restos de divisões inteiras. É essencial para o funcionamento das cifras utilizadas na criptografia, como as cifras afim, de César e Vigenère.

BNCC: Base Nacional Comum Curricular. Documento normativo que orienta os currículos da educação básica brasileira. Fundamenta a aplicação pedagógica desenvolvida na dissertação.

Cifra Afim: Cifra criptográfica baseada na função matemática do tipo c = (a·m + b) mod n, onde a e n devem ser primos entre si. Permite uma substituição mais segura e parametrizável.

Cifra de César: Técnica criptográfica clássica de substituição monoalfabética, em que cada letra é deslocada um número fixo de posições no alfabeto.

Cifra de Vigenère: Cifra polialfabética que utiliza uma palavra−chave para aplicar diferentes deslocamentos às letras da mensagem, tornando a cifra mais segura que as monoalfabéticas.

Codificação: Processo de conversão de informações em símbolos, sinais ou códigos. Na criptografia, refere−se à transformação de texto claro em texto cifrado.

Competência Matemática: Capacidade de aplicar conhecimentos matemáticos para resolver problemas em contextos diversos, como os desafios criptográficos propostos.

Congruência Modular: Relação entre números inteiros que deixam o mesmo resto ao serem divididos por um mesmo número. Exemplo: 17 ≡ 5 (mod 12).

Criptodisco: Ferramenta física composta por discos concêntricos utilizada para realizar cifras de substituição baseadas em deslocamento.

Criptografia: Ciência que estuda técnicas para codificar mensagens, protegendo informações contra acessos não autorizados.

Função Afim: Função matemática expressa por f(x) = ax + b. É usada para compor cifras como a cifra afim.

MDC: Máximo Divisor Comum. Maior número inteiro que divide dois números simultaneamente. É usado para verificar a existência de inverso multiplicativo.

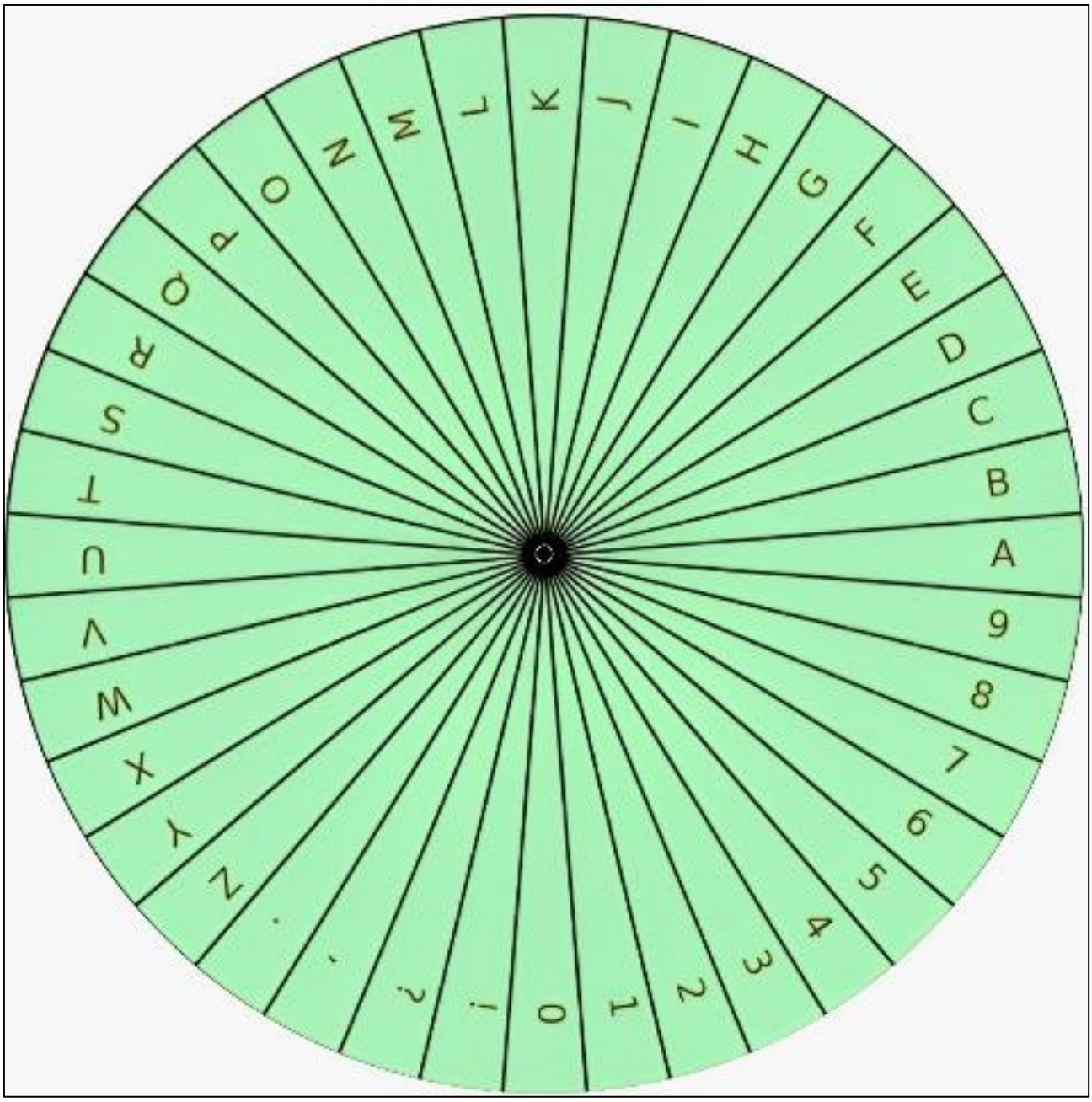
OBMEP: Olimpíada Brasileira de Matemática das Escolas Públicas. Competição que estimula o raciocínio lógico−matemático entre estudantes.

Pensamento Computacional: Habilidade de formular problemas e soluções de forma que possam ser executados por um computador, promovendo lógica, abstração e sistematização.

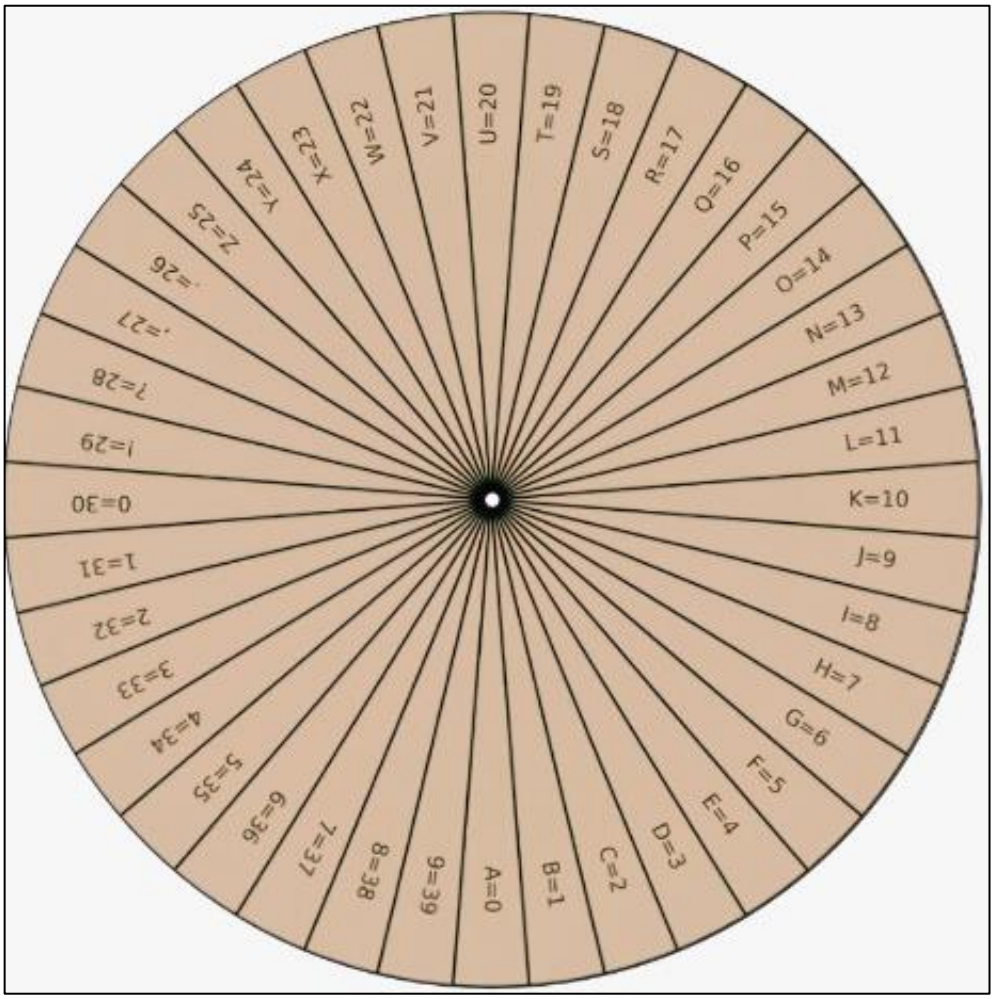
Roda de Criptografia: Dispositivo físico formado por discos sobrepostos que simulam deslocamentos de cifras como a de César.

Visual Studio: Ambiente de desenvolvimento utilizado para implementar os algoritmos criptográficos propostos na dissertação.

APÊNDICE A - Molde para o Criptodisco – Parte Inferior



APÊNDICE B - Molde para o Criptodisco – Parte Superior



APÊNDICE C - Atividade: Criptografia na Prática: Cifra de César, Cifra de Vigenère e Programação

Aplicação da Atividade: Escola Y.
Público Alvo: 1º Ano do Ensino Médio
Nome dos Alunos: _____

Objetivo da Atividade: Compreender o funcionamento de métodos clássicos de criptografia, aplicando raciocínio lógico e aritmética modular no processo de codificação e decodificação de mensagens, além de relacionar conceitos matemáticos com contextos práticos, históricos e tecnológicos.

Etapa 1. Codificação com Criptodisco – Cifra de Cesar

A Cifra de César utiliza uma chave simétrica, o que significa que o mesmo valor de deslocamento é aplicado tanto para codificar quanto para decodificar uma mensagem. Por exemplo, se a letra "A" for deslocada em três posições, ela se transforma em "D"; para decifrar, basta deslocar "D" três posições para trás, retornando à letra original. Nesta atividade, utilizaremos um alfabeto expandido, que inclui letras, números e símbolos, permitindo associar cada caractere a um número de 0 a 39 (sendo A=0, B=1, C=2, ..., 9=39). Dessa forma, o processo de cifragem pode ser representado pela seguinte função matemática:

$$c(x) \equiv x + k \pmod{40}$$

Nesse contexto, considera−se:

- x como a posição original da letra no alfabeto;
- k como a chave responsável pelo deslocamento aplicado na codificação;
- c(x) é o valor resultante após a codificação, ou seja, a nova posição da letra cifrada.

Mensagem Original: A MATEMÁTICA SE TORNOU DIVERTIDA
Chave: 7
Codificada: _____

Mensagem Original: _____
Chave: _____
Codificada: _____

Mensagem Codificada: M QP2OMOM. Q M NM0Q ,M!M M !QMXU7MOM. PQ 0.ZT.0

Chave: 12

Mensagem Original: _____

Reflexão e Discussão

Como esse exercício se relaciona com conteúdos matemáticos?

_____

_____

Onde podemos encontrar esse tipo de codificação na vida real?

_____

_____

Etapa 2. Codificação com Cifra de Vigenère e Programação

A cifra de Vigenère representa uma evolução em relação às cifras monoalfabéticas, como a de César, por utilizar uma chave composta por letras que se repete ciclicamente ao longo da mensagem. Cada letra da chave define um deslocamento distinto, o que torna a criptografia mais robusta e menos suscetível à análise de frequência. Para aplicar esse método, é necessário primeiro atribuir valores numéricos às letras do alfabeto, conforme a tabela abaixo:

Tabela. Codificação Alfabética com Valores Inteiros de A = 0 a Z = 25

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

Fonte: Elaborado pelo autor com base em COUTINHO (2009)

Sua fórmula matemática é:

$$ci \equiv (mi + ki) \bmod 26$$

onde:

- ci é o caractere cifrado na posição i,
- mi é o valor da letra original na posição i,
- ki é o valor da letra correspondente da chave,
- a operação é feita módulo 26 (alfabeto com 26 letras).

Código para inserir no Visual Studio:

```
#include <iostream>
#include <string>
using namespace std;
string cifraVigenere(const string& mensagem, const string& chave) {
    string resultado = "";
    int chaveIndex = 0;
    int tamanhoChave = chave.length();
    for (char letra : mensagem) {
        if (isalpha(letra)) {
            char base = isupper(letra) ? 'A' : 'a';
            char k = toupper(chave[chaveIndex % tamanhoChave]) - 'A';
            int m = letra - base;
            int c = (m + k) % 26;
            resultado += static_cast<char>(base + c);
            chaveIndex++;
        }
        else {
            resultado += letra; // mantém espaços e pontuação
        }
    }
    return resultado;
}
int main() {
    string mensagemOriginal, chave;
    cout << "=== Cifra de Vigenère ===" << endl;
    cout << "Digite a mensagem: ";
    getline(cin, mensagemOriginal);
    cout << "Digite a palavra-chave: ";
    cin >> chave;
    string criptografada = cifraVigenere(mensagemOriginal, chave);
    cout << "\nMensagem criptografada: " << criptografada << endl;
    return 0;
}
```

Desafio da Cifra de Vigenère:

Realize a codificação da mensagem utilizando a tabela de dados apresentada anteriormente e, em seguida, compare o resultado com a saída gerada pelo código no Visual Studio.

Mensagem: SOU SO UM VIRUS, QUERENDO ESCAPAR, DOS PROGRAMADORES DA VIDA

Chave utilizada: TECLADO

Codificada: _____

Reflexão e Discussão

Houve alguma diferença entre o resultado da tabela e o resultado obtido no Visual Studio? Se sim, qual?_____

Qual etapa foi mais fácil: codificar manualmente com a tabela ou usar o código em C++? Por quê? _____

O que você percebeu ao transformar uma ideia matemática em código de criptografia usando lógica de programação?

_____

_____

Etapa 3. Experiência vivenciada durante a atividade

Link: https://www.menti.com/al9uobrrih6w