UNIVERSIDADE FEDERAL DE ALAGOAS
INSTITUTO DE MATEMÁTICA
MESTRADO PROFISSIONAL EM MATEMÁTICA EM REDE NACIONAL

MARCEL CAVALCANTE CERQUEIRA

O Estudo da Criptografia RSA no Ensino Básico com Auxílio de Softwares Computacionais

Maceió
2016

MARCEL CAVALCANTE CERQUEIRA

O Estudo da Criptografia RSA no Ensino Básico com Auxílio de Softwares Computacionais

Dissertação de Mestrado apresentada ao Mestrado Profissional em Matemática em Rede Nacional do Instituto de Matemática da Universidade Federal de Alagoas como requisito parcial para obtenção do grau de Mestre em Matemática.

Orientador: *Prof. Dr. Fernando Pereira Micena*

Maceió
2016

Catalogação na fonte
Universidade Federal de Alagoas
Biblioteca Central
Divisão de Tratamento Técnico
Bibliotecária Responsável: Helena Cristina Pimentel do Vale

C416e Cerqueira, Marcel Cavalcante.
O estudo da criptografia RSA no ensino básico com auxilio de softwares computacionais / Marcel Cavalcante Cerqueira. – 2016.
61f. ; il.

Orientador: Fernando Pereira Micena.
Dissertação (Mestrado Profissional em Matemática) – Universidade Federal de Alagoas. Instituto de Matemática. Programa de Pós Graduação de Mestrado Profissional em Matemática em Rede Nacional, 2016.

Bibliografia: f. 61.

1. Matemática – Estudo ensino. 2. Criptografia RSA. 3. Matemática – Ensino auxiliado por computador. . I. Título.

CDU: 51:371.315

Folha de Aprovação

MARCEL CAVALCANTE CERQUEIRA

O ESTUDO DA CRIPTOGRAFIA RSA NO ENSINO BÁSICO COM O AUXILIO DE SOFTWARES COMPUTACIONAIS

Dissertação submetida ao corpo docente do Programa de Mestrado Profissional em Matemática em Rede Nacional (PROFMAT) do Instituto de Matemática da Universidade Federal de Alagoas e aprovada em 01 de abril de 2016.

Banca Examinadora:

Prof. Dr. Gregório Manoel da Silva Neto - UFAL(Presidente)

Prof. Dr. André Luiz Flores - UFAL

Prof. Dr. Carlos Argolo Pereira Alves - IFAL

Ao meu orientador, prof. Dr. Fernando Micena,
aos meus pais, irmãos, esposa, filha e amigos.

AGRADECIMENTOS

À Deus, por ser o autor principal das páginas da minha vida e presente em todos os momentos de angústia, alegria e evidentemente pela sua divina providência.

À minha mãe, finada durante esta caminhada.

À minha filha, nascida durante esta caminhada.

À minha esposa, pelo amor.

Aos meus familiares a homenagem da mais profunda gratidão pelo apoio e renúncia, imprescindível nos momentos valorosos e especiais. Por terem incentivado a não desistirmos e a prosseguirmos na certeza da realização deste grande sonho.

Aos amigos Profmat, principalmente André Carlos e Aldo, pela contribuição na colobora-ção e socialização do aprendizado.

Aos professores e a Instituição porque foram, são e serão marcos significativos de referência.

À CAPES e ao Profmat.

Eu Acredito, que ás vezes são as pessoas que ninguém espera nada que fazem as coisas que ninguém consegue imaginar.

—ALAN TURING

RESUMO

O objetivo deste trabalho é apresentar uma proposta didática para o ensino da criptografia RSA no ensino básico com o auxílio de softwares computacionais como o GeoGebra, o Python, e software online. Para que a proposta se concretize com sucesso é preciso trabalhar alguns conceitos elementares da aritmética e do uso de tais softwares no ensino deste tema.

Palavras-chave: 1. Matemática - Estudo ensino. 2. Criptografia RSA. 3. Matemática - Ensino auxiliado por computador.

ABSTRACT

The goal of this work is to present a didactic proposal for the teaching of RSA encryption in basic education with the help of computer softwares GeoGebra, Python, and online software. So the proposal is successfully materialize it takes work some basic concepts of arithmetic and software use of the softwares in teaching of this subject.

Keywords: 1. Mathematics - Study teaching . 2. RSA encryption . 3. Mathematics - Teaching computer aided .

SUMÁRIO

1. INTRODUÇÃO

Este trabalho foi motivado em minha prática de ensino da matemática ao perceber que muitos estudantes concluiam o ensino básico sem conseguir resolver algumas propriedades aritméticas básicas. Juntamente com uma necessidade pessoal, de conhecer sobre o tema criptografia, nascida com a perda de arquivos pessoais, pelo roubo de meu notebook . Daí veio a ideia de criar uma proposta didática que propicie a percepção de que os conteúdos de criptografia, aritmética básica (como a divisão euclidiana) e conceitos aritméticos mais sofisticados (como a aritmética modular). E com o uso de software, fazer com que os alunos tenham um aprendizado sobre o tema.

Tais temas são de suma importância para o desenvolvimento de um cidadão, assim como retrata os parâmetros curriculares nacionais na disciplina de matemática: à medida que vamos nos integrando ao que se denomina uma sociedade da informação crescentemente globalizada, é importante que a Educação se volte para o desenvolvimento das capacidades de comunicação, de resolver problemas, de tomar decisões, de fazer inferências, de criar, de aperfeiçoar conhecimentos e valores, de trabalhar cooperativamente.

Como fazer para que essa fragmentação do conhecimento seja superada? Como o uso da tecnologia computacional pode ser útil para essa finalidade?

A estruturação deste trabalho se encontra do seguinte modo. No Capítulo 1, intitulado Conceitos Preliminares, apresentamos uma síntese de alguns conceitos inerentes a aritmética. No Capítulo 2, intitulado Aritmética Modular, apresentamos os conceitos da aritmética modular. No Capítulo 3, Criptografia RSA, introduzimos os conceitos básicos de criptografia e construimos a criptografia RSA. No Capítulo 4, O uso de softwares computacionais no ensino de criptografia, mostramos alguns caminhos do uso dos softwares GeoGebra e Python, sobretudo no ensino da criptografia. No Capitulo 5, Momentos de Aprendizagens, temos uma proposta para implementação em sala de aula que leva ao aluno entender a criptografia RSA com o uso de softwares.

2. CONCEITOS PRELIMINARES

2.1. Introdução

Apresentaremos algumas noções sobre números primos e fatorações, destacamos o algoritmo de Euclides e o teorema da fatoração única. Em seguida, a infinitude dos números primos, e o princípio da indução finita.

2.2. Divisão Euclidiana

Teorema 2.2.1 (Divisão Euclidiana)**.** *Sejam a e b dois números inteiros positivos com $0 < b < a$. Existem dois únicos inteiros positivos q e r tais que*

$$a = bq + r, \ \ com \ \ r < b.$$

Demonstração. Suponha que $b > a$ e considere, enquanto fizer sentido, os números

$$a,\ a-b,\ a-2b,\ ...,\ a-nb,\ ...$$

O conjunto S formado pelos elementos acima tem um menor elemento $r = a - qb$. Vamos provar que $r < b$.

Se $b \mid a$, então $r = 0$ e nada mais temos a provar. Se, por outro lado, $b \nmid a$, então $r \neq b$, e, portanto, basta mostrar que não pode ocorrer $r > b$. De fato, se isto ocorresse, existiria um inteiro positivo $c < r$ tal que $r = c + b$. Consequentemente, sendo $r = c + b = a - qb$, teríamos

$$c = a - (q+1)b \ \in S, \ \ com \ \ c < r,$$

contradição com o fato de r ser o menor elemento de S.

Portanto, temos que $a = bq + r$ com $r < b$, o que prova a existência de q *e* r.

Agora, vamos provar a unicidade. Note que, dados dois elementos distintos de S, a diferença entre o maior e o menor deses elementos, sendo um múltiplo de b, é pelo menos b. Logo, se $r = a - bq$ e $r' = a - bq'$, com $r < r' < b$, teríamos $r' - r \geq b$, o que acarretaria $r' \geq r + b \geq b$, absurdo. Portanto, $r = r'$.

Daí segue-se que $a - bq = a - bq'$, o que implica que $bq = bq'$ e, portanto, $q = q'$.

□

Definição 2.2.1. *Diremos que o número natural d, não nulo, é um máximo divisor comum (mdc) dos números naturais a e b se possuir as seguintes propriedades:*

i) d é um divisor comum de a e de b, e

ii) d é divisível por todo divisor comum de a e b.

O algoritmo de Euclides permite-nos encontrar o mdc, (a,b), de dois inteiros a e b. Os detalhes deste algoritmo serão discutidos na proposição seguinte.

2.2.1 Algoritmo de Euclides

Proposição 2.2.1. *Sejam a e b dois inteiros positivos com $a \geq b$ e seja $\{r_i\}$ a sequencia de inteiros construídos da seguinte maneira: divida a por b; chamamos de q_1 o quociente desta divisão e r_1 seu resto, de forma que*

$$a = bq_1 + r_1, \quad 0 \leq r_1 < b.$$

Da mesma maneira, dividimos b por r_1, levando-nos a

$$b = r_1 q_2 + r_2, \quad 0 \leq r_2 < r_1.$$

Iteramos de forma que

$$r_{i-1} = r_i q_{i+1} + r_{i+1}, \quad 0 \leq r_{i+1} < r_i.$$

A sequência $\{r_i\}$ é estritamente decrescente. Portanto, deve haver um natural n tal que $r_{n+1} = 0$. Segue que $r_n = (a,b)$.

Demonstração. Começamos mostrando que $r_n \mid a$ e $r_n \mid b$. Como $r_{n+1} = 0$, a última equação pode ser escrita como $r_{n-1} = q_{n+1} r_n$. Logo, $r_n \mid r_{n-1}$. A penúltima equação é $r_{n-2} = q_n r_{n-1} + r_n$. Como $r_n \mid r_{n-1}$, segue que $r_n \mid q_n r_{n-1} + r_n$. Logo, $r_n \mid r_{n-2}$. Iteramos pelas equações uma a uma, obtendo que $r_n \mid r_i$ para todo i. Assim, $r_n \mid q_2 r_1 + r_2 = b$. Finalmente, como $r_n \mid b$ e $r_n \mid r_1$, temos que $r_n \mid bq_1 + r_1 = a$. Portanto, $r_n \mid a$ e $r_n \mid b$, o que implica imediatamente que $r_n \mid (a,b)$.

Seja d um divisor de a e b. Devemos mostrar que d divide r_n. Agora, a iteração pelo sistema será para baixo. Como $d \mid a$ e $d \mid b$, então $d \mid r_1 = a - bq_1$. Na segunda equação, temos que $d \mid b$ e $d \mid r_1$; logo, $d \mid r_2 = b - r_1 q_2$. Iterações mostram que $d \mid r_i$ para todo i. Em particular, $d \mid r_n$.

Podemos concluir que $r_n = (a,b)$.

□

Corolário 2.2.1. *Sejam a e b inteiros e seja $c = (a,b)$. Existem $x, y \in \mathbb{Z}$ tais que $c = ax + by$.*

Demonstração. Nossa demonstração faz uso da proposição logo acima. Sabemos que $c = r_n$. Novamente iteramos para cima pelas equações. Como $r_{n-2} = q_n r_{n-1} + r_n$,

$$r_n = r_{n-2} - q_n r_{n-1}. \tag{2.1}$$

Substituindo $r_{n-1} = r_{n-3} - q_{n-1} r_{n-2}$, (2.1) torna-se

$$r_n = r_{n-2}(1 + q_{n-1} q_n)) - q_n r_{n-3}. \tag{2.2}$$

Agora, substituímos $r_{n-2} = r_{n-4} - q_{n-2} r_{n-3}$. Continuando essas substituições iteradas, somos levados a $r_n = r_1 x_1 + r_2 y_1$, com $x_1, y_1 \in \mathbb{Z}$. Substituímos $r_2 = b - r_1 q_2$, levando-nos a

$$r_n = r_1(x_1 - q_2 y_1) + by_1.$$

Finalmente, substituímos $r_1 = a - bq_1$, obtemos o resultado final

$$r_n = a(x_1 - q_2 y_1) + b(-q_1 x_1 + q_1 q_2 y_1 + y_1) = ax + by,$$

onde $x = x_1 - q_2 y_1$ e $y = -q_1 x_1 + q_1 q_2 y_1 + y_1$.

□

2.3. Teorema fundamental da aritmética

Definição 2.3.1 (Número Primo)**.** *Um número inteiro n $(n > 1)$ possuindo somente dois divisores positivos n e 1 é chamado de número primo.*

Se $n > 1$ não é primo será chamado composto, assim, sendo n um número composto, existirá um divisor n_1 de n tal que $n_1 \neq 1$ e $n_1 \neq n$. Portanto, existirá um número inteiro n_2 tal que

$$n = n_1 n_2, \quad com \;\; 1 < n_1 < n \;\; e \;\; 1 < n_2 < n$$

.

Por exemplo, 2, 3, 5, 7, 11 e 13 são primos, enquanto que 4, 6, 8, 9, 10 e 12 são compostos.

O lema abaixo será usado na demonstração de uma propriedade chamada propriedade fundamental dos números primos e para a sua demonstração nos basearemos na divisão euclidiana estendida.

Lema 2.3.1. *Sejam a, b e c inteiros positivos e suponhamos que a e b são primos entre si.*

(1) Se b divide o produto ac então b divide c.

(2) Se a e b dividem c então o produto ab divide c.

Demonstração.

(1) Temos por hipótese que a e b são primos entre si; isto é, que o $mdc(a,b) = 1$. O algoritmo euclidiano estendido nos garante que existem inteiros α e β tais que

$$\alpha.a + \beta.b = 1.$$

multiplicando esta equação por c

$$\alpha.a.c + \beta.b.c = c. \tag{2.3}$$

É evidente que a segunda parcela da soma de 2.3 é divisível por b. Mas a primeira também é divisível por b pela hipótese inicial. Portanto, c é divisível por b, como queríamos mostrar.

(2) A segunda afirmação pode ser provada a partir da primeira. De fato, se a divide c, podemos escrever $c = at$, para algum t. Mas b também divide c. Como a e b são primos entre si, segue da afirmação (1) que b dividir t. Assim teremos que $t = bk$, para algum inteiro k. Portanto

$$c = at = a(bk) = (ab)k$$

é divisível por ab, assim mostramos a afirmação (2).

□

Existem muitas oportunidades de usar as duas partes desse lema, começando pela demonstração de uma propriedade importante dos números primos. Já conhecida pelos gregos antigos, esta propriedade aparece como a Proporção 30 do livro VII dos Elementos de Euclides.

Proposição 2.3.1 (Propriedade Fundamental dos Números Primos)**.** *Seja p um número primo e a e b inteiros positivos. Se p divide o produto ab então p divide a ou p divide b.*

Demonstração. Para provar isto usaremos o Lema 2.3.1. Se p dividir a, está demonstrado. Digamos, então, que p não divide a. Neste ponto usaremos o fato de que p é primo, para concluir que se p não divide a então p e a são primos entre si. Isto ocorre porque qualquer divisor comum a p e a divide p; mas os únicos divisores de p é 1 e p. Portanto, se p não divide a, então $mdc(a,p) = 1$. Daí podemos aplicar o lema: como p e a são primos entre si e como p divide ab temos que p divide b.

□

2.3.1 Teorema da Fatoração Única

Teorema 2.3.1 (Teorema Fundamental da Aritmética)**.** *Dado um inteiro positivo $n \geq 2$ podemos sempre escrevê-lo, de modo único, na forma*

$$n = p_1^{e_1} ... p_k^{e_k},$$

onde $1 < p_1 < p_2 < p_3 < ... < p_k$ são número primos e $e_1, ..., e_k$ são inteiros positivos.

Os expoentes $e_1, ..., e_k$ no teorema são chamados de multiplicidades. Assim a multiplicidade de p_1 na fatoração de n é e_1. Em outras palavras, a multiplicidade de p_1 é o maior expoente e_1 tal que $p_1^{e_1}$ divide n. Observe também que n tem k fatores primos distintos, mas que a quantidade total de fatores primos (distintos ou não) é a soma das multiplicidades $e_1 + ... + e_k$.

Demonstração. Vamos dividir a demonstração em duas partes: a parte (i) com o objetivo de mostrar a existência da fatoração e a parte (ii) para mostrar que esta tal fatoração é única.

Parte (i):

Temos nesta primeira parte que, dado um inteiro $n \geq 2$, podemos escrevê-lo como produto de primos. A melhor maneira de fazer isto é explicando um algoritmo que, tendo por entrada n, determina seus fatores primos e respectivos expoentes.

Começaremos descrevendo um algoritmo mais simples, para determinar apenas um fator de um inteiro dado. Dado n inteiro, tentemos dividir n por cada um dos inteiros de 2 a $n-1$. Se algum destes inteiros dividir n, então achamos um fator de n. E mais, o menor fator que acharmos desta maneira tem que ser primo.

Vejamos porque esta última afirmação é verdadeira. Seja f um inteiro tal que $2 \leq f \leq n-1$. Suponhamos que f é o menor fator de n e que f' é um fator (maior que 1) de f. Pela definição de divisibilidade, existem inteiros a e b tais que

$$n = f.a \quad e \quad f = f'.b.$$

Logo $n = f'.ab$. Portanto f' também é fator de n. Como estamos supondo que f é o menor fator de n, concluímos que $f \leq f'$. Por outro lado, f' é fator de f, o que só pode acontecer se $f' \leq f$. Das duas desigualdades segue que $f = f'$. Sendo assim, o único fator de f maior que 1 é o próprio f. Logo f é primo, que é a afirmação original.

Há ainda uma outra observação que precisamos fazer antes de descrever o algoritmo em detalhes. Pelo que vimos o algoritmo consiste em fazer uma busca, começando de 2, para achar um fator de n. Evidentemente nossa busca não vai ultrapassar de $n-1$ passos: um número inteiro não pode ter um fator maior que ele próprio. Na verdade, não precisamos procurar fatores maiores que $\sqrt{n}$. Lembre-se que o algoritmo que estamos descrevendo determina o menor fator de n maior que 1. Portanto basta verificar que o menor fator de n, maior que 1, é sempre menor ou igual a $\sqrt{n}$. Mas cuidado: há um caso em que isto é falso: Se n é primo, então seu menor fator maior que 1 é o próprio n. Portanto o que temos que verificar é que se n é composto e se $f > 1$ é seu menor fator então $f \leq \sqrt{n}$.

Seja, portanto, n um número composto e $f > 1$ seu menor fator. Então, existe um inteiro positivo a tal que $n = f.a$. Como f é o menor fator, certamente $f \leq a$. Mas $a = n/f$, logo $f \leq n/f$. Disto segue que $f^2 \leq n$, que é equivalente a $f \leq \sqrt{n}$, como queríamos mostrar.

Resumindo o que vimos até agora: o algoritmo deve buscar um número que divida n, começando de 2 e avançando até $\sqrt{n}$. Se n for composto, vamos encontrar o menor fator de n por este método, e este fator é necessariamente primo. Se nenhum dos números pesquisados é fator de n, isto significa que n é primo. Vamos ao algoritmo:

Algoritmo de fatoração
Entrada: inteiro positivo n.
Saída: inteiro positivo f que é o menor fator primo de n ou uma mensagem indicando que n é primo.

Etapa 1: comece fazendo $f = 2$.
Etapa 2: se $\frac{n}{f}$ é inteiro escreva "f é fator de n" e pare; senão vá para Etapa 3.
Etapa 3: incremente f de uma unidade e vá para Etapa 4.
Etapa 4: se $f > \sqrt{n}$ escreva n é primo e pare; senão volte à Etapa 2.

Dado um inteiro $n > 0$, temos assim uma maneira de determinar se n é primo e, se não for, achar um fator de n. Caso n seja primo, já achamos a sua fatoração. Mas se n for composto, queremos achar todos os seus fatores primos e respectivas multiplicidades. Para isto basta aplicar o algoritmo várias vezes.

Digamos que, aplicando o algoritmo a n achamos o fator q_1. Então q_1 é o menor fator primo de n. A seguir aplicando o algoritmo ao co-fator $\frac{n}{q_1}$, determinando assim um segundo fator primo de n, que vamos chamar de q_2. É claro que $q_1 \leq q_2$, mas pode acontecer que sejam iguais, basta que q_1^2 divida n. Continuando, aplicamos o algoritmo ao co-fator $\frac{n}{q_1 q_2}$ e assim por diante. Com isso determinamos uma sequência não decrescente de números primos

$$q_1 \leq q_2 \leq q_3 \leq ... \leq q_s$$

cada uma dos quais é fator de n. A esta sequência corresponde uma outra, formada pelos co-fatores,

$$n > \frac{n}{q_1} > \frac{n}{q_1 q_2} > \frac{n}{q_1 q_2 q_3} > ... > 0.$$

Observe que esta última é uma sequência estritamente decrescente de números inteiros positivos. Como há apenas n inteiros entre n e 0, o algoritmo tem que parar depois de no máximo

n laços. O que pode-se verificar é que o menor elemento da sequência de co-fatores é sempre 1. Para poder escrever a fatoração na forma do enunciado do teorema, basta contar quantos fatores primos iguais ocorrem entre os $q's$. Isto nos permitirá determinar as multiplicidades dos diversos fatores primos.

Parte (ii):

Tendo mostrado a existência da fatoração, é chegada a hora de mostrar que esta fatoração é única. Suponhamos que existe algum inteiro que admite mais de uma fatoração na forma estabelecida no teorema, mostraremos que esta é única. Vamos chamar de n um inteiro positivo entre aqueles que têm duas fatorações distintas. Podemos escrever

$$n = p_1^{e_1}...p_k^{e_k} = q_1^{r_1}...q_s^{r_s} \tag{2.4}$$

onde $p_1 < ... < p_k$ e $q_1 < ... < q_s$ são primos e $e_1,\ e_2,\ ...,e_k,\ r_1,...,r_s$ são inteiros positivos. Mas estamos supondo que estas fatorações são diferentes. Isto pode acontecer por duas razões. Os primos da fatoração à direita podem não ser os mesmos da esquerda; ou, se forem os mesmos, podem ter multiplicidades diferentes.

De acordo com a fatoração da esquerda, p_1 é um primo que divide n. Mas $n = q_1^{r_1}...q_s^{r_s}$, segundo a fatoração a direita. A propriedade fundamental dos primos nos garante então que p_1 deve dividir um dos fatores do produto à direita. Isto significa, em última análise, que p_1 divide um dos primos da fatoração da direita. Mas um primo só pode dividir outro se forem iguais. Logo p_1 tem que ser um dos primos $q_1, q_2, ...\ ou\ q_k$. Suponhamos sem perda de generalidade, que $p_1 = q_1$. Fazendo interativamente tal processo, chegamos a

$$p_1 = q_1,\ p_2 = q_2,\ ...,p_k = q_k\ \ onde\ \ k \leq s$$

Por outro lado, fazendo-se o mesmo, partindo-se da fatoração a direita, obtemos

$$q_1 = p_1,\ q_2 = p_2,\ ...,q_S = p_S\ \ onde\ \ s \leq k$$

Logo temos $k = s$.

Agora, basta mostramos que coincidem as multiplicidades, isto é, $e_i = r_i$, $onde\ i = 1,\ 2,\ ...,k$. Suponhamos sem perda de generalidade, que $p_i = q_i\ onde\ i = 1,\ 2,\ ...,k$. Como $p_1^{e_1}$ divide n e $mdc(p_1^{r^1}, p_2^{r_2}...p_k^{r_k}) = 1$, segue que $p_1^{e_1}$ divide $p_1^{r_1}$, logo e_1 menor ou igual a r_1, analogamente r_1 menor ou igual a e_1, portanto $e_1 = r_1$. Da mesma forma para os demais fatores, $e_2 = r_2, ..., e_k = r_k$ $(e_i = r_i, i = 1, ..., k)$.

□

2.3.2 O crivo de Eratóstenes

Como já dito, é possível verificar se um inteiro N é primo tentando-se simplesmente dividi-lo por todos os números n tais que $n^2 \leq N$.

Eratóstenes, no século III a.E.C., teve a ideia de organizar os cálculos sob a forma do crivo bem conhecido, que leva o seu nome. O crivo serve para determinar todos os números primos e também os fatores primos dos números compostos inferiores a um número N dado arbitrariamente.

Vamos ilustrar o processo tomando como exemplo $N = 101$.

Opera-se da maneira seguinte: escrevem-se todos os inteiros até 101; riscam-se todos os múltiplos de 2, superiores a 2; em cada nova etapa, são riscados todos os múltiplos do menor inteiro p ainda não riscado e que são maiores do que p. Basta chegar-se ao número p tal que p^2 já ultrapassa 101.

Figura 2.1 Números primos

	2	3	~~4~~	5	~~6~~	7	~~8~~	~~9~~	~~10~~
11	~~12~~	13	~~14~~	~~15~~	~~16~~	17	~~18~~	19	~~20~~
~~21~~	~~22~~	23	~~24~~	~~25~~	~~26~~	~~27~~	~~28~~	29	~~30~~
31	~~32~~	~~33~~	~~34~~	~~35~~	~~36~~	37	~~38~~	~~39~~	~~40~~
41	~~42~~	43	~~44~~	~~45~~	~~46~~	47	~~48~~	~~49~~	~~50~~
~~51~~	~~52~~	53	~~54~~	~~55~~	~~56~~	~~57~~	~~58~~	59	~~60~~
61	~~62~~	~~63~~	~~64~~	~~65~~	~~66~~	67	~~68~~	~~69~~	~~70~~
71	~~72~~	73	~~74~~	~~75~~	~~76~~	~~77~~	~~78~~	79	~~80~~
~~81~~	~~82~~	83	~~84~~	~~85~~	~~86~~	~~87~~	~~88~~	89	~~90~~
~~91~~	~~92~~	~~93~~	~~94~~	~~95~~	~~96~~	97	~~98~~	~~99~~	~~100~~
101									

Fonte: Autor (2015)

Assim, todos os múltiplos de $2,3,5,7 < \sqrt{101}$ são cortados. O número 53 é primo porque não foi riscado. Então, os números primos não superiores a 101 são:

$$2,3,5,7,11,13,17,19,23,29,31,37,41,43,47,53,59,61,67,71,73,79,83,89,97 \ \ e \ \ 101.$$

O processo constitui a base da teoria do crivo, que foi desenvolvida com o objetivo de fornecer estimativas da quantidade de números primos satisfazendo dadas condições.

2.4. A infinitude dos números primos

Uma pergunta natural que pode ser feita é quantos números primos existem? Da qual a resposta é: “Existe uma infinidade de números primos".

Daremos três demonstrações para esta afirmação, no entanto, existem muitas outras como podemos ver em [RIBENBOIM, 2012].

2.4.1 A demonstração de Euclides

Demonstração. Suponhamos que a sucessão $p_1 = 2,\ p_2 = 3,\ ...,\ p_r$ dos r números primos seja finita. Façamos $P = p_1.p_2...p_r + 1$ e seja p um número primo que divide P. Esse número p não

pode ser igual a qualquer um dos números $p_1,\ p_2,\ ...p_r$ porque então ele dividiria a diferença $P - p_1.p_2...p_r = 1$, o que é impossível. Assim p é um número primo que não pertence à sucessão e, por consequência, $p_1,\ p_2,\ ...,\ p_r$ não pode formar o conjunto de todos os números primos.

□

2.4.2 A demonstração de Thue

Demonstração. Suponha que existam finitos primos

$$p_1 = 2 < ... < p_r.$$

Sabendo-se que

$$2^n \to \infty \ \ para \ \ n \to \infty \ \ e \ \ \frac{2^n}{(n+1)^r} \to \infty \ \ para \ \ n \to \infty,$$

considere $n \geq 1$ tal que

$$2^n > p_r \quad e \quad (n+1)^r < 2^n.$$

Seja m inteiro positivo, tal que

$$1 \leq m \leq 2^n.$$

Os possíveis valores de m percorrem o conjunto $\{1, ..., 2^n\}$, que possui 2^n elementos.

Pelo (T.F.A.), $m = 2^{e_1}3^{e_2}...p_r^{e_r}$, com $0 \leq e_i \leq n \ \ e \ \ 1 \leq i \leq r$. Fazendo a análise combinatória dos expoentes, segue que existem exatamente $(n+1)^r$ números escritos em tal forma.

Porém a quantidade 2^n de elementos do conjunto do qual m pertence, não pode ser superior a quantidade $(n+1)^r$ dos possíveis m avaliada na combinatória de sua fatoração. Ou seja

$$(n+1)^r \geq 2^n.$$

Absurdo, pois

$$(n+1)^r < 2^n.$$

□

2.4.3 A demonstração de Euler

Esta prova se baseia na famosa identidade de Euler:

$$\sum_{n=1}^{\infty} \frac{1}{n} = \prod_{i=1}^{\infty} \left(\sum_{k=0}^{\infty} \frac{1}{p_i^k} \right),$$

onde p_i é o i-ésimo primo.

Demonstração. Suponha que tenhamos apenas um número finito de primos $2 = p_1 < p_2 < ... < p_r$. Então a identidade de Euler toma a forma

$$\sum_{n=1}^{\infty} \frac{1}{n} = \prod_{i=1}^{r} \left(\sum_{k=0}^{\infty} \frac{1}{p_i^k} \right).$$

Agora a série $\sum \frac{1}{p_i^k}$ converge, pois trata-se de uma série geométrica. A saber, vale que $\sum \frac{1}{p_i^k} = \frac{1}{1-p_i^{-1}}$.
Voltando para a identidade de Euler, temos

$$\sum \frac{1}{p_i^k} = \frac{1}{1-p_i^{-1}} = \prod_{i=1}^{r} \frac{1}{1-p_i^{-1}} < +\infty.$$

Portanto a série harmônica converge, o que é um absurdo. □

2.5. Princípio da indução finita

Há uma propriedade adicional que só os números naturais (inteiros positivos) possuem, que é o Axioma de Indução que passamos a descrever.

2.5.1 Princípio da Indução Matemática

Teorema 2.5.1. *Seja $P(n)$ a proposição que queremos provar. Para que $P(n)$ seja verdadeira para todo n natural, basta que:*

(i) $p(1)$ seja verdadeira;

(ii) e se $P(k)$ for verdadeira para algum número natural k, então $P(k+1)$ também seja verdadeira.

Não vamos demonstrar que o método funciona. Para fazer a demonstração é preciso um valioso axioma e que não vemos a necessidade do mesmo para apenas esta demonstração. Ao invés disto, vamos nos convencer de que o método é razoável. Começamos sabendo que $P(1)$ é verdade. Mas, de acordo com (ii), se $P(1)$ é verdade, então $P(2)$ é verdade. Por (ii) de novo, se $P(2)$ é verdade, então $P(3)$ é verdade; e assim por diante.

É como a queda de dominó. Ponha cada peça do dominó, de pé, do lado de outro, a uma pequena distância. Neste caso, a afirmação $P(n)$ é: a n-ésima peça do dominó cai. De fato, se derrubamos a primeira peça (isto é, se $P(1)$ é verdadeira) e se a queda da k-ésima peça derruba a $k+1$-ésima (isto é, se $P(k)$ verdadeira implica que $P(k+1)$ é verdadeira) então caem todas as peças do dominó.

3. ARITMÉTICA MODULAR

3.1. Introdução

Apresentaremos uma das noções mais fecundas da aritmética, introduzida por Gauss. Trata-se da realização de uma aritmética com os restos da divisão Euclidiana por um número fixado.

3.2. Aritmética dos restos

Definição 3.2.1. *Se a, b e $m > 1$ são inteiros, dizemos que a é congruente a b módulo m se $m \mid (a-b)$. Denotamos isto por*

$$a \equiv b \ (mod\ m).$$

Se $m \nmid (a-b)$ dizemos que a é incongruente a b módulo m e denotamos

$$a \not\equiv b \ (mod\ m).$$

Exemplo 3.2.1.

a) $11 \equiv 3 \ mod\ 2$ pois $2 \mid (11-3)$.

b) como $5 \nmid 6$ e $6 = (17-11)$ temos que $17 \not\equiv 11 \ mod\ 5$.

Proposição 3.2.1. *Dados a, b e $m > 0$ inteiros tem-se:*

a) Se r é o resto da divisão Euclidiana de a por m, então $a \equiv r \ (mod\ m)$. Além disso, para todo inteiro a, existe $r \geq 0$ tal que $a \equiv r \ (mod\ m)$ onde $r \in \{0,1,2,...,m-1\}$ que forma o que chamamos de sistema completo de resíduos módulo m.

b) Se $a \equiv b \ (mod\ m)$, então a e b tem o mesmo resto quando divididos por m.

Demonstração. a) Pela divisão Euclidiana dos inteiros a e m existem q e r tais que $a = mq + r$, onde $r < m$. Temos $mq = a - r$ o que implica $m \mid a - r$ ou seja, $a \equiv r(mod\ m)$. Como $0 \leq r < m$, então qualquer inteiro a quando dividido por m terá o $r \in \{0,1,2,...,m-1\}$, logo $a \equiv r \ (mod\ m)$.

b) Suponha que a e b tenham restos r_1 e r_2, respectivamente, na divisão por m e que $a \equiv b \ (mod\ m)$. Deste modo,

$$\begin{cases} a = mq_1 + r_1 \\ b = mq_2 + r_2 \end{cases}$$

subtraindo as equações, temos

$$a-b=m(q_1-q_2)+(r_1-r_2),$$

mas $a \equiv b\ (mod\ m)$, assim, $a-b=mq_3$. Logo,

$$m(q_1-q_2)+(r_1-r_2)=mq_3 \Leftrightarrow m(q_1+q_3-q_2)=r_2-r_1 \Leftrightarrow$$

$$\Leftrightarrow m \mid |r_2-r_1|<m \Leftrightarrow r_2-r_1=0 \Leftrightarrow r_1=r_2.$$

□

O teorema a seguir mostra que as congruências são uma classe muito importante em matemática e que chamamos de relação de equivalência.

Teorema 3.2.1. *Sejam $a,\ b,\ c$ e $m>0$ inteiros, tem-se:*

i) $a \equiv a\ mod\ m$ *(reflexiva);*

ii) *se* $a \equiv b\ mod\ m$, *então* $b \equiv a\ mod\ m$ *(simétrica);*

iii) *se* $a \equiv b\ mod\ m$ *e* $b \equiv c\ mod\ m$, *então* $a \equiv c\ mod\ m$ *(transitiva).*

Demonstração. i) $m \mid 0 \Rightarrow m \mid (a-a) \Rightarrow a \equiv a\ mod\ m$.

ii) Se $a \equiv b\ mod\ m \Rightarrow m \mid (a-b) \Rightarrow m \mid [-(b-a)] \Rightarrow m \mid (b-a) \Rightarrow b \equiv a\ mod\ m$.

iii) Se $a \equiv b\ mod\ m$ e $b \equiv c\ mod\ m$, então $m \mid (a-b)$ e $m \mid (b-c)$, respectivamente, assim:

$$m \mid [(a-b)+(b-c)] \Rightarrow m \mid (a-c) \Rightarrow a \equiv c\ mod\ m.$$

□

Proposição 3.2.2. *Se $a,\ b,\ c$ e $m>0$ são inteiros tais que $a \equiv b\ mod\ m$, então:*

i) $a+c \equiv b+c\ mod\ m$;

ii) $a-c \equiv b-c\ mod\ m$;

ii) $ac \equiv bc\ mod\ m$.

Demonstração. i) Como $a \equiv b\ mod\ m$, temos que $m \mid (a-b)$, e portanto, $m \mid [(a+c)-(b+c)]=(a-b)$, logo $a+c \equiv b+c\ mod\ m$.

ii) Como $(a-c)-(b-c)=a-b$, e $m \mid (a-b)$, segue que $a-c \equiv b-c\ mod\ m$.

iii) Como $m \mid (a-b)$, então $a-b=mq$, assim $ac-bc=mcq$, logo $m \mid (ac-bc)$, portanto $ac \equiv bc\ mod\ m$.

□

Proposição 3.2.3. *Se $a,\ b,\ c,\ d$ e $m > 0$ são inteiros tais que $a \equiv b\ mod\ m$ e $c \equiv d\ mod\ m$, então,*

i) $a + c \equiv b + d\ mod\ m$;

ii) $a - c \equiv b - d\ mod\ m$;

iii) $ac \equiv bd\ mod\ m$.

As demonstrações seguem diretamente da definição.

Proposição 3.2.4. *Se $a,\ b,\ c$ e $m > 0$ são inteiros e $ac \equiv bc\ mod\ m$, então $a \equiv b\ mod\ \frac{m}{d}$ com $d = mdc(c,m)$.*

Demonstração. De $ac \equiv bc\ mod\ m$ temos $ac - bc = c(a-b) = qm$. Se dividirmos por d, teremos $(\frac{c}{d})(a-b) = q.\frac{m}{d}$; logo $(\frac{m}{d}) \mid [(\frac{c}{d})(a-b)]$ e, como $mdc(\frac{m}{d}, \frac{c}{d}) = 1$, segue que, $(\frac{m}{d}) \mid (a-b)$ o que implica $a \equiv b\ mod\ \frac{m}{d}$.

□

Proposição 3.2.5. *Se $a,\ b,\ k > 0$ e $m > 0$ são inteiros e $a \equiv b\ mod\ m$, então $a^k \equiv b^k\ mod\ m$.*

Demonstração. Segue imediatamente da identidade

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

□

Proposição 3.2.6. *Se $a \equiv b\ mod\ m_1$, $a \equiv b\ mod\ m_2$, ... , $a \equiv b\ mod\ m_k$ onde $a,\ b,\ m_1,\ m_2,\ ...\ m_k$ são inteiros com m_i positivos para $i = 1, 2..., k$, então*

$$a \equiv b\ mod\ [m_1, m_2, ..., m_k]$$

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

Demonstração. Se $a \equiv b\ mod\ m_i$, para todo $i = 1, 2, ..., k$, então, $m_i \mid (a-b)$ para todo $1 \leq i \leq k$. Sendo $a - b$ múltiplo de cada m_i, segue-se que $[m_1, m_2, ..., m_k] \mid (a-b)$ o que nos mostra que $a \equiv b\ mod\ [m_1, m_2, ..., m_k]$.

□

3.3. Congruência Linear

Chamamos de congruência linear em uma variável a uma congruência da forma $ax \equiv b\ mod\ m$ onde x é uma incógnita.

É fácil de se verificar que se x_0 é uma solução, i.e., $ax_0 \equiv b\ mod\ m$ e $x_1 \equiv x_0\ mod\ m$ então x_1 também é solução. Isto é óbvio pois se $x_1 \equiv x_0\ mod\ m$ então $ax_1 \equiv ax_0 \equiv b\ mod\ m$.

Uma equação da forma $ax + by = c$, onde a, b e c são inteiros é chamada equação diofantina linear. (o nome vem do matemático grego Diofanto).

Teorema 3.3.1. *Sejam a e b 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 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, pois, que $d \mid c$. Pelo corolário 2.2.1 existem inteiros n_0 e m_0, tais que

$$an_0 + bm_0 = d \tag{3.1}$$

Como $d \mid c$, existe um inteiro k tal que $c = kd$. Se multiplicarmos ambos os lados de (3.1) 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

$$x = x_0 + \frac{b}{d}k$$

$$y = y_0 - \frac{a}{d}k$$

são soluções, uma vez que

$$ax+by = a(x_0 + \frac{b}{d}k) + b(y_0 - \frac{a}{d}k) =$$

$$= ax_0 + \frac{ab}{d}k + by_0 - \frac{ab}{d}k = ax_0 + by_0 = c$$

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 $x = x_0 + \frac{b}{d}k$, $y = y_0 - \frac{a}{d}k$. Vamos supor que (x,y) seja solução, i.e., $ax+by = c$. Mas, como $ax_0 + by_0 = c$, obtemos, subtraindo membro a membro, que

$$ax+by-a_0-by_0 = a(x-x_0)+b(y-y_0) = 0,$$

o que implica $a(x-x_0) = b(y_0-y)$. Como $d = (a,b)$ temos,

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

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

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

Logo, pela parte 1 do lema 2.3.1, $\left(\frac{b}{d}\right)(x-x_0)$ e portanto existe um inteiro k satisfazendo $x-x_0 = k\left(\frac{b}{d}\right)$, ou seja $x = x_0 + \left(\frac{b}{d}\right)k$. Substituindo-se este valor de x na equação 3.3 temos $y = y_0 - \left(\frac{a}{d}\right)k$, o que conclui a demonstração.

□

Teorema 3.3.2. *Sejam a, b e m inteiros tais que $m > 0$ e $(a,m) = d$. No caso em que $d \nmid b$ a congruência $ax \equiv b \ mod \ m$ não possui nenhuma solução e quando $d \mid b$, possui exatamente d soluções incongruentes módulo m.*

Demonstração. Pela própria definição de congruência dizemos que o inteiro x é solução de $ax \equiv b \ mod \ m$ se, e somente se, existe um inteiro y tal que $ax = b + my$, ou, o que 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 $ax \equiv b \ mod \ m$ possui infinitas soluções dadas por $x = x_0 - (m/d)k$. Como estamos interessados em saber o números de soluções incongruentes, vamos tentar sob que condições $x_1 = x_0 - (m/d)k_1$ e $x_2 = x_0 - (m/d)k_2$ são congruentes módulo m. Isto implica $(m/d)k_1 \equiv (m/d)k_2 \ mod \ m$, e como $(m/d) \mid m$, temos $(m/d, m) = m/d$, o que permite o cancelamento de m/d resultando, pela Proposição 3.2.4, $k_1 \equiv k_2 \ mod \ d$. 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.

□

3.4. Pequeno Teorema de Fermat

Teorema 3.4.1 (Fermat)**.** *Se p é um número primo e a um inteiro qualquer, então*

$$a^p \equiv a \ mod \ p.$$

Além disso, se $p \nmid a$, então

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

Demonstração. Para $p = 2$, temos que $a^2 \equiv a \ mod \ 2$, pois $2 \mid (a^2 - a) = a(a-1)$, números consecutivos.

Agora seja $p > 2$. Faremos uma prova por indução sobre a. Para $a = 1$, tem-se $1^p \equiv 1 \ mod \ p$, pois $p \mid 0$. Agora suponhamos que $a^p \equiv a \ mod \ p$, para $a > 1$ iremos provar que $(a+1)^p \equiv (a+1) \ mod \ p$.

Pelo binômio de Newton,

$$(a+1)^p - (a+1) = a^p - a + \binom{p}{1}a^{p-1} + \ldots + \binom{p}{p-1}a$$

como $p \mid \binom{p}{i}$ para $1 \leq i \leq p-1$ e $p \mid (a^p - a)$ por hipótese de indução, logo $p \mid [(a+1)^p - (a+1)]$, então

$$(a+1)^p \equiv (a+1)\ mod\ p.$$

Portanto a relação é verdadeira pelo princípio de indução finita para todo $a > 1$.
Para $a = 0$ a relação é direta pois $0^p \equiv 0\ mod\ p$.
Agora se $a < 0$, então $-a > 0$ ou seja,

$$(-a)^p \equiv -a\ mod\ p$$

como p é ímpar pois o único primo par é $p = 2$ que já vimos, então

$$-a^p \equiv -a\ mod\ p$$

logo pela propriedade de produto por um inteiro, por exemplo, -1, temos,

$$a^p \equiv a\ mod\ p.$$

Caso $p \nmid a$, como $a^p \equiv a\ mod\ p$ e $p \mid (a^p - a) = a(a-1)$, então $p \mid (a^{p-1} - 1)$, ou seja,

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

□

Exemplo 3.4.1. *Encontrar o resto da divisão de* 237^{28} *por* 13.

Sabemos que $237 \equiv 3\ mod\ 13$, assim $237^4 \equiv 3^4 \equiv 3\ mod\ 13$.
Por Fermat temos que $237^{12} \equiv 1\ mod\ 13$, logo $237^{24} = (237^{12})^2 \equiv (1)^2 \equiv 1\ mod\ 13$.
Multiplicando $237^4 \equiv 3\ mod\ 13$ por $237^{24} \equiv 1\ mod\ 13$ temos:

$$237^{28} \equiv 3\ mod\ 13.$$

3.5. Teorema de Euler

3.5.1 A função ϕ de Euler

Definição 3.5.1. *Seja m um inteiro positivo. Dizemos que $\phi(m)$ corresponde a quantidade de inteiros positivos menores que m e primos com m. Por convenção, $\phi(1) = 1$.*

Exemplo 3.5.1.

$\phi(15) = 8$, pois $(1,15) = (2,15) = (4,15) = (7,15) = (8,15) = (11,15) = (13,15) = (14,15) = 1$
Com a função $\phi : \mathbb{N}^* \longrightarrow \mathbb{N}$ estabelecida algumas relações como veremos abaixo.

Teorema 3.5.1. *Sejam m e m' inteiros positivos com $m > 1$, $m' > 1$ e $(m,m') = 1$. Então,*

$$\phi(m.m') = \phi(m).\phi(m').$$

Observação: a demonstração foge do desígnio do trabalho.

Proposição 3.5.1. *Se p é um número primo e r um inteiro positivo, então tem-se que*

$$\phi(p^r) = p^r - p^{r-1} = p^{r-1}(p-1).$$

Demonstração. De 1 até p^r, temos p^r números naturais que não são primos com p^r, ou seja, todos os múltiplos de p, que são precisamente

$$1.p,\ 2.p,\ \ldots p^{r-1}.p$$

cujo número é p^{r-1}. Portanto,

$$\phi(p^r) = p^r - p^{r-1} = p^{r-1}(p-1).$$

□

Lema 3.5.1. *Se $m = p_1^{\alpha_1}.p_2^{\alpha_2}...p_n^{\alpha_n}$ é a decomposição de m em fatores primos, então*

$$\phi(m) = p_1^{\alpha_1-1}(p_1-1)p_2^{\alpha_2-1}(p_2-1)...p_n^{\alpha_n-1}(p_n-1).$$

A demonstração segue diretamente do Teorema (3.5.1) e da Proposição (3.5.1).

Observação: Se $p\ \ e\ \ q$ são primos e $m = pq$ temos que $\phi(m) = (p-1)(q-1)$.

Exemplo 3.5.2. *Calcule $\phi(34)$.*

Como $34 = 2^1.17^1$

$$\phi(34) = (2-1)(17-1) = 16.$$

Teorema 3.5.2 (Euler)**.** *Se m é um inteiro positivo e a um inteiro com $(a,m) = 1$, então*

$$a^{\phi(m)} \equiv 1\ mod\ m.$$

Demonstração. Vamos considerar p primo sendo um dos fatores primos da decomposição de m e para $k > 0$ vamos fazer a prova por indução sobre k para expressão

$$a^{\phi(p^k)} \equiv 1\ mod\ p^k$$

inicialmente, considerando $k = 1$ temos $a^{\phi(p)} \equiv 1\ mod\ p$, mas $\phi(p) = p-1$ pois p é primo, como $(a.p) = 1$. Temos que $a^{p-1} \equiv 1\ mod\ p$ é o próprio teorema de Fermat e portanto verdadeiro.

Agora, iremos admitir por hipótese de indução que $a^{\phi(p^k)} \equiv 1\ mod\ p^k$ é válido para $k > 1$. Devemos provar que a afirmação é verdadeira para $k+1$.

Por hipótese, $p^k \mid a^{\phi(p^k)} - 1$. Logo,

$$a^{\phi(p^k)} = p^k q + 1 \tag{3.2}$$

para algum q inteiro. Além disso,

$$\phi(p^{k+1}) = p^k(p-1) = p.p^{k-1}(p-1) \quad e \quad \phi(p^k) = p^{k-1}(p-1)$$

$$\phi(p^{k+1}) = p.\phi(p^k) \tag{3.3}$$

assim temos,

$$[a^{\phi(p^{k+1})}] - 1 =^{3.3} a^{p.\phi(p^k)} - 1 = [a^{\phi(p^k)}]^p - 1 =^{3.2} [p^k.q+1]^p - 1 =$$

$$= \binom{p}{1}.p^k.q + \sum_{j=2}^{p} \binom{p}{j} (p^k.q)^j = p^{k+1}.q + \sum_{j=2}^{p} \binom{p}{j} (p^k.q)^j.$$

Como,

$$p \mid \binom{p}{j} \quad e \quad p^k \mid (p^k.q)^j, \;\; 2 \leq j \leq p$$

temos,

$$p^{k+1} \mid \binom{p}{j} (p^k.q)^j$$

sabendo que,

$$p^{k+1} \mid p^{k+1}.q$$

concluimos que,

$$p^{k+1} \mid (a^{\phi(p^{k+1})} - 1) \Rightarrow a^{\phi(p^{k+1})} \equiv 1 \; mod \; p^{k+1}.$$

Assim, pelo princípio de indução finita

$$a^{\phi(p^k)} \equiv 1 \; mod \; p^k.$$

Adotaremos a decomposição de m em fatores primos seja, $m = p_1^{\alpha_1}.p_2^{\alpha_2}...p_n^{\alpha_n}$, denotando $m = p_j^{\alpha_j}.m_j$ onde $m_j = p_1^{\alpha_1}...p_2^{\alpha_2}...\overline{p_j^{\alpha_j}}...p_n^{\alpha_n}$ é a decomposição com a exclusão do fator $p_j^{\alpha_j}$.

Observe que $(p_j^{\alpha_j}, m_j) = 1$, sabemos do caso anterior

$$a^{\phi(p_j^{\alpha_j})} \equiv 1 \; mod \; p_j^{\alpha_j}, \;\; 1 \leq j \leq n \tag{3.4}$$

Elevando a equação (3.4) acima, por $\phi(m_j)$.

$$\left[a^{\phi(p_j^{\alpha_j})}\right]^{\phi(m_j)} \equiv 1 \; mod \; p_j^{\alpha_j}$$

assim,

$$a^{\phi(p_j^{\alpha_j}.\, m_j)} \equiv 1 \ mod \ p_j^{\alpha_j}$$

que equivale,

$$a^{\phi(m)} \equiv 1 \ mod \ p_j^{\alpha_j}, \ \ 1 \leq j \leq n$$

assim,

$$p_j^{\alpha_j} \mid [a^{\phi(m)} - 1], \ \ 1 \leq j \leq n.$$

Deste modo, como $(p_i, p_j) = 1$ para $i \neq j$ obtemos um sistema de congruências

$$\begin{cases} a^{\phi(m)} \equiv 1 \ mod \ p_1^{\alpha_1} \\ a^{\phi(m)} \equiv 1 \ mod \ p_2^{\alpha_2} \\ \qquad \vdots \\ a^{\phi(m)} \equiv 1 \ mod \ p_n^{\alpha_n} \end{cases}$$

Portanto pela Proposição (3.2.6), temos

$$a^{\phi(m)} \equiv 1 \ mod \ [p_1^{\alpha_1}, \ldots, p_n^{\alpha_n}],$$

e como $m = [p_1^{\alpha_1}, \ldots, p_n^{\alpha_n}]$, então

$$a^{\phi(m)} \equiv 1 \ mod \ m.$$

□

O próximo teorema é um teste de primalidade muito impotante, no entanto, não tão eficiente pela grande necessidade de eficiência computacional.

Teorema 3.5.3 (Wilson)**.** *Será p um número primo se, e somente se, $(p-1)! \equiv -1 \ mod \ p$*

Demonstração. .
($\Rightarrow$) Como $(2-1)! \equiv 1 \equiv -1 \ mod \ 2$ o resultado é válido para $p = 2$. Pelo teorema 3.3.2, a congruência $ax \equiv 1 \ mod \ p$ tem uma única solução para todo a no conjunto $\{1, \ 2, \ldots \ , \ p-1\}$ e como, destes elementos, somente 1 e $p-1$ são seus próprios inversos módulo p, podemos agrupar os números $2, \ 3, \ldots, \ p-2$ em $\frac{(p-3)}{2}$ pares cujo produto seja congruente a 1 módulo p. Se multiplicarmos estes congruencias, membro a membro, teremos:

$$2 \times 3 \times \ldots \times (p-2) \equiv \ 1 \ mod \ p$$

e multiplicando ambos os lados desta congruência por $p-1$ obtemos:

$$2 \times 3 \times \ldots \times (p-2) \times (p-1) \equiv \ (p-1) \ mod \ p$$

isto é,

$$(p-1)! \equiv \ -1 \ mod \ p, \ \ pois \ \ (p-1) \equiv \ -1 \ mod \ p.$$

($\Leftarrow$) Suponha que p composto, ou seja, $p = a.b, \ \ 1 < a < p$ e satisfaz a relação $(p-1)! \equiv -1 \ mod \ p$, então $p \mid [(p-1)! + 1]$, mas como, $a \mid p$, logo $a \mid [(p-1)! + 1]$ e, além disso, $a \mid (p-1)!$, pois $1 < a < p$. Concluimos que $a \mid 1$ o que implica que $a = 1$ (absurdo).

□

4. CRIPTOGRAFIA RSA

4.1. Introdução

Neste capítulo falaremos um pouco sobre a criptografia de uma forma geral e construiremos todos os aspectos ligado a criptografia RSA.

4.2. Criptografia

Em grego, cryptos significa secreto, oculto. A criptografia estuda métodos para codificar uma mensagem de modo que só seu destinatário legítimo consiga interpretá-la. É a arte dos “código secretos", que todos já praticamos quando criança. O mais simples destes códigos consiste em substituir uma letra pela seguinte; isto é transladar o alfabeto uma casa para diante.

Na criptografia, o texto normalmente é chamado de mensagem e uma pessoa autorizada a ler a mensagem é chamada destinatário. O autor da mensagem pode ser chamado de remetente e a ação de criptograr uma mensagem é chamada de codificação da mensagem. Utiliza-se a expressão decodificar para o ato que deve ser realizado pelo destinatário para conversão da mensagem criptografada para a mensagem original, enquanto a expressão decifrar é utilizada para a conversão realizada por outra pessoa não autorizada. Para alguns processos de criptografia, se um não destinatário decifra uma mensagem codifica por algum método ele é capaz de decifrar qualquer mensagem codificada pelo tal método. Dizemos então que “o código foi quebrado", código aí sendo utilizado no sentido do método utilizado para a codificação.

Hoje, a comunicação entre computadores pela internet vem criando novos desafios para criptografia. Como é relativamente fácil interceptar mensagens enviadas por linha telefônica, torna-se necessário codificá-las, sempre que contenham informações sensíveis. Isto inclui transações bancárias ou comerciais, ou até mesmo uma compra feita com cartão de crédito.

Podemos imaginar um banco que deseja fazer uma transação com milhões de reais para um outro banco. Ele precisa ter certeza que ninguém vai interceptar esta mensagem. Por outro lado, ele precisa ter a certeza que o outro banco recebeu a quantia com sucesso, para isto a mensagem precisa estar assinada.

Por isso tornou-se necessario a invenção de novos códigos, de tal forma que fossem difíceis de ser decifrados, mesmo que uma pessoa possuisse um computador. Tais códigos foram criados justamente para esses tipos de transações financeiras. Por isso os códigos modernos, chamados de criptografia de chave pública, são formados por duas chaves distintas, a pública e a privada, uma para codificar e outra para decodificar mensagens. Neste método cada pessoa ou entidade mantém duas chaves: uma pública, que pode ser divulgada livremente, e outra privada, que deve ser mantida em segredo. Esta é a ideia introduzida em 1976 por W. Diffie e M.E. Hellman da Universidade de Stanford e, independentemente, por R.C. Merkle da Universidade de Califórnia.

4.3. Criptografia RSA

O método de criptografia RSA foi inventado em 1978 por R.L.Rivest, A. Shamir e L. Adleman, que na época trabalhavam no Massachussets Institute of Techonology (M.I.T). As letras RSA correspodem às iniciais dos inventores do código.

Suponhamos, por simplicidade, que a mensagem original é um texto onde não há números, apenas palavras, e os espaços entre as palavras. Então a primeira coisa a se fazer na criptografia RSA é transformar a mensagem em uma sequência de números.

Utilizaremos 99 para os espaços entre as palavras, e para as letras, a tabela abaixo. Tal escolha é boa, pois estamos sempre associando cada elemento à um único número de exatamente dois dígitos. Assim, quando retornarmos para texto, haverá um único caracter associado.

Figura 4.1 Pré-codificação

a	b	c	d	e	f	g	h	i	j	k	l	m
10	11	12	13	14	15	16	17	18	19	20	21	22

n	o	p	q	r	s	t	u	v	w	x	y	z
23	24	25	26	27	28	29	30	31	32	33	34	35

Fonte: Autor (2015)

Portanto, em última análise a mensagem é constituída pelas letras que formam as palavras e pelos espaços entre palavras. Esta primeira parte é chamada de pré-codificação, para distingui-la do processo de codificação (encriptação) e decodificação (desencriptação) propriamente dito.

Exemplo 4.3.1. *Para ilustração, pré-codificando a frase 'Aprendendo a criptografia RSA', obtemos o número*

$$10252714231314231324991099122718252924162710.$$

Ocorre também que o número encontrado deverá ser separado em blocos, os quais devem seguir os seguintes critérios: não coincidir com alguma unidade linguística (letra, palavra, ou qualquer outra), para evitar a decodificação por contagem de frequencia; não iniciar com zero, para não trazer problemas na decodificação; ser menor do que n, onde este é o produto de dois números primos (p e q) que são escolhidos para a criação de um sistema de criptografia RSA (como veremos mais a frente).

Exemplo 4.3.2. *Escolhidos dois números primos, sejam $p = 11$ e $q = 97$, assim como $n = pq$, então $n = 1067$. Realizando a separação em blocos do número encontrado no exemplo anterior, seguindo-se os devidos critérios, teriamos:*

$$1025, 271, 423, 131, 423, 132, 499, 109, 912, 271, 82, 52, 924, 162, 710.$$

Dentre vários outros possíveis.

A propriedade abaixo é levada em consideração para o processo de codificação e decodificação.

Proposição 4.3.1. *Sejam a e n dois inteiros com $a < n$. Se $(a,n) = 1$, existe um único $x \in \{1,\ 2, ...,\ n-1\}$ tal que*

$$ax \equiv\ 1\ mod\ n$$

A demonstração segue diretamente da Proposição 3.3.2.

4.3.1 A ideia por trás do RSA

Um sistema de criptografia de chave pública é inicialmente estabecido pela pessoa (ou organização), este chamaremos de receptor, que deseja receber mensagens de uma maneira segura. É o receptor que estabelece o sistema e publica como enviar mensagens.

Vamos fazer um exemplo para ilustrar o processo RSA, apresentado em etapas.

Exemplo 4.3.3. *Bob (receptor) deseja receber uma mensagem de Alice (emissor). Eles irão utilizar o processo de criptografia RSA. Para isto, realizam as etapas abaixo.*

1ª etapa:

Bob escolhe dois números primos, sejam $p = 3$ e $q = 5$, e então calcula $n = pq$, obtendo $n = 15$.

2ª etapa:

Bob aplica $n = 15$ na função de Euler $\phi(n = pq) = (p-1)(q-1)$, assim obtém $\phi(15) = (3-1)(5-1) = 2.4 = 8$.

3ª etapa:

Bob escolhe a chave de encriptação $e \in \{1, ..., 14\}$, da qual $(e, \phi(15)) = 1$, como por exemplo $e = 3$. Esta será usada por Alice para a encriptação.

4ª etapa:

Bob calcula a chave de desencriptação $d \in \{1, ..., 14\}$, a qual $3d \equiv\ 1\ mod\ \phi(15)$, assim $d = 11$. Esta permanece secreta em poder de Bob, que a utilizará para desencriptar a mensagem recebida.

5ª etapa:

Alice recebe de Bob a chave pública $n = 15$ e $e = 3$, para então criptografar a mensagem 13, e envia-la para Bob. Assim, obtém $a = 7$ de $a \equiv\ 13^3\ mod\ 15$. Alice envia o número 7 para Bob.

6ª etapa:

Bob recebe a mensagem encriptada $a = 7$ de Alice. Para desencriptografar , usa a sua chave secreta $d = 11$. Então faz $m \equiv\ 7^{11}\ mod\ 15$, e encontra a mensagem $m = 13$.

Daremos o algoritmo de uma forma mais geral e detalhando alguns pontos importantes no processo de encriptação e desencriptação.

Passo 1

O receptor escolhe dois números primos p e q, e calcula $n = pq$ que é a “chave pública". Por questão de vulnerabilidade do sistema, os números escolhidos não deverão ser consecultivos, e terem aproximadamente cem dígitos cada. Assim, n terá aproximadamente 200 dígitos, dificultando que computadores recuperem p e q numa quantidade de tempo razoável.

Passo 2

O receptor calcula $\phi(n)$, onde ϕ é a função de Euler definida como segue: $\phi(n)$ é o número de inteiros em $\{1,\ 2, ...,\ n-1\}$ relativamente primos com n, para $n > 1$. Por convenção, definimos $\phi(1) = 1$. Na observação do Lema 3.5.1 vimos $\phi(n) = (p-1)(q-1)$. Note que esta fórmula requer conhecimento sobre p e q. Logo, calcular $\phi(n)$ sem conhecer a fatoração de n parece tão difícil quanto fatorar n.

Passo 3: escolhendo a chave de encriptação.

O receptor escolhe $e \in \{1,\ ...,n-1\}$ relativamente primo com $\phi(n)$. O número e é a chave de encriptação. Este número é público e é usado pelo emissor para codificar a mensagem seguindo instruções publicadas pelo receptor.

Passo 4: construindo a chave de desencriptação.

Existe $d \in \{1,\ ...,\ n-1\}$ tal que $ed \equiv\ 1\ mod\ \phi(n)$. A existência de d segue da Proposição 4.3.1. Esta chave, a “chave privada", permanece secreta em poder do receptor e permite ao mesmo desencriptar suas mensagens recebidas.

Passo 5: encriptando uma mensagem.

O emissor quer enviar uma mensagem que consiste num inteiro $m \in \{1,\ ...,\ n-1\}$. Para encriptá-la, o emissor calcula o resto a da divisão de m^e por n. Assim, temos $m^e \equiv\ a\ mod\ n$, com $a \in \{1,\ ...,n-1\}$. O inteiro calculado a é a mensagem encriptada. O emissor envia a.

Passo 6: desencriptando uma mensagem.

O receptor recebe uma mensagem encriptada a. Para desencriptá-la, o receptor calcula $m \equiv a^d\ mod\ n$. Mostraremos mais a frente, na Proposição 4.3.2, que isto sempre levará precisamente à mensagem original m.

Para acompanhar os resultados apresentados nos próximos exemplos, recomendo o uso de uma calculadora científica.

Exemplo 4.3.4. *Bob (o receptor) escolhe dois números primos, sejam $p = 641$ e $q = 109$, faz o produto dos mesmos*

$$n = (641).(109) = 69869.$$

Aplica $n = 69869$ na função de Euler

$$\phi(69869) = (641-1)(109-1) = 640.108 = 69120.$$

Bob escolhe a chave de encriptação $e \in \{1,...,69868\}$, *a qual* $(e,\phi(69869)) = (e,69120) = 1$. *Seja então*

$$e = 169.$$

Calcula a chave de decriptação $d \in \{1,...,69868\}$, *da qual* $169d \equiv 1 \ mod\ \phi(69869) = 69120$, *assim temos*

$$d = 409.$$

Alice (a emisora) recebe de Bob, a chave pública $n = 69869$ *e* $e = 169$, *e a utiliza para enviar a mensagem numérica* $m = 4875$ *para Bob. Para encripta-la, calcula* $a \equiv 4875^{169} \ mod\ 69869$, *e obtém*

$$a = 39331.$$

Bob recebe a mensagem encriptada $a = 39331$ *de Alice e deseja decifrá-la. Assim utiliza-se da sua chave secreta* $d = 409$, *para calcular* $m \equiv 39331^{409} \ mod\ 69869$. *E encontra*

$$m = 4875.$$

que é a real mensagem que Alice queria passar para Bob.

Proposição 4.3.2. *Encriptação e desencriptação RSA são inversas uma da outra: se encriptamos uma mensagem* m *como* a, *onde* $m^e \equiv a \ mod\ n$, *a desincriptação sempre leva à mensagem original* m. *Isto é,* $a^d \equiv m \ mod\ n$.

Demonstração. Se $(m,p) = 1$, o pequeno teorema de Fermat nos dá que

$$a^d \equiv m^{ed} = m^{k(p-1).(q-1)+1} = (m^{p-1})^{k(q-1)}.m = 1^{k(q-1)}.m = m \ mod\ p.$$

Se, por outro lado, $(m,p) \neq 1$, temos que $p \mid m$ ou, em outras palavras, $m \equiv \ mod\ p$. Desse modo,

$$a^d \equiv m^{ed} \equiv 0 \equiv m \ mod\ p.$$

Portanto, $a^d \equiv m \ mod\ p$ em qualquer caso, isto é, $a^d - m$ é múltiplo de p. Contas análogas nos são que $a^d \equiv m \ mod\ q$ e, portanto, $a^d - m$ também é múltiplo de q. Como p e q são primos distintos, temos que $n = pq \mid (a^d - m)$, isto é, $a^d \equiv m \ mod\ n$.

□

Exemplo 4.3.5. *Alice ao receber de Bob, a chave pública* $e = 529$ *e* $n = 1067$, *resolve enviar a frase "Aprendendo a criptografia RSA". Sabe-se que para a pré-codificação é utilizada a tabela abaixo, e que espaços em branco é* 99.

Figura 4.2 Pré-codificação

a	b	c	d	e	f	g	h	i	j	k	l	m
10	11	12	13	14	15	16	17	18	19	20	21	22

n	o	p	q	r	s	t	u	v	w	x	y	z
23	24	25	26	27	28	29	30	31	32	33	34	35

Fonte: Autor (2015)

Pré-codificando, ela obtém o número

$$10252714231314231324991099122718252924162710.$$

Que separando em blocos, seguindo os critérios, ela obtém

$$1025,271,423,131,423,132,499,109,912,271,82,52,924,162,710.$$

Encriptando, cada um dos blocos, utilizando-se da relação $a \equiv m^{529} \ mod\ 1067$*, onde* m *é o valor do bloco e* a *é o bloco encriptado. Obteve*

$$721,602,229,1033,229,132,762,109,252,602,306,821,143,744,904.$$

E então enviou para Bob.

Bob ao receber de Alice a mensagem encriptada, vai em busca de desencripta-la. Como ele detém que $d = 49$ *e* $n = 1067$*, então para cada um dos blocos, utilizando-se da relação* $m \equiv a^{49} \ mod\ 1067$ *(onde* m *é o valor numérico do bloco), obtém*

$$1025,271,423,131,423,132,499,109,912,271,82,52,924,162,710.$$

Que quebrando os blocos, temos

$$10252714231314231324991099122718252924162710.$$

E agora quebrando a pré-decodificação, a real mensagem é

$$Aprendendo\ a\ criptografia\ RSA$$

4.3.2 Assinando uma mensagem

Até agora, vimos como uma pessoa, chamemos de Bob, poderia estabelecer um sistema de criptografia de chave pública, permitindo receber seguramente mensagens de qualquer um. Suponha que Bob receba uma mensagem de sua amiga Alice, pedindo-o para transferir uma grande quantia de dinheiro para sua conta. Isto prova que a mensagem realmente veio de Alice, e não de alguém passando-se por ela. Então torna-se necessário a Alice provar que ela é, de fato, a autora da mensagem enviada a Bob. Isto é o que chamamos de assinar uma mensagem.

Neste caso, ambos o emissor e o receptor, constroem um sistema de criptografia de chave pública, consistindo na tripla (n,e,d). Duas chaves públicas são necessárias.

- O emissor (Alice) compartilha n_A e e_A, mantendo d_A secreto.
- O receptor (Bob) publica n_B e e_B, mantendo d_B secreto.

4.3.3 Transmitindo uma mensagem assinada

- Para enviar uma mensagem m, o emissor começa pondo sua assinatura ao calcular

$$m_1 \equiv m^{d_A} \ mod \ n_A.$$

Ela então codifica m_1 com a chave pública do receptor:

$$m_2 \equiv m_1^{e_B} \ mod \ n_B.$$

O emissor então envia m_2.

- Para desencriptar a mensagem assinada, o receptor começa por recuperar m_1, desencriptando-a com a sua chave secreta d_B:

$$m_1 \equiv m_2^{d_B} \ mod \ n_B.$$

Com efeito, pela Proposição 4.3.2,

$$m_2^{d_B} \equiv m_1^{e_B d_B} \equiv m_1 \ mod \ n_B.$$

Em seguida, ele recupera a mensagem original usando a chave pública do emissor:

$$m \equiv m_1^{e_A} \ mod \ n_A.$$

De fato, usando a Proposição 4.3.2

$$m_1^{e_A} \equiv m^{d_A e_A} \equiv m \ mod \ n_A.$$

Se a mensagem for enviada por um impostor, isto ficará óbvio ao receptor após a desencriptação.

4.3.4 Aplicações

O sistema de criptografia *RSA* é extensivamente usado na internet para transmitir dados sensíveis como informações de cartões de crédito. O sistema bancário também é protegido por encriptação *RSA*. Assim são aplicações de segurança que utilizam como base os algoritmos de criptografia RSA, entre elas: Certificados de Segurança, Assinaturas Digitais, S/Mime e PGP; protocolo SSL, TLS e IPSec. No entanto, os algoritmos *RSA* requerem cálculos longos e complexos.

4.3.5 Construindo primos grandes

O Teorema do número primo, nos diz, em termos simples, a probabilidade de um número aleatório de N dígitos ser um número primo. Para construir um número primo de cem dígitos, simplesmente geramos aleatoriamente números de cem dígitos e testamos se são primos. Tal Teorema nos assegura que depois de uma média de 115 tentativas, obteríamos um número primo (assumindo que gerássemos apenas números ímpares).

Teorema 4.3.1 (Teorema do número primo)**.** *Seja $\pi(N) = \#\{p \leq N : p \text{ primo}\}$ (isto é, $\pi(N)$ é a quantidade de números primos menores ou iguais a N). Desde que N seja relativamente grande, temos*

$$\pi(N) \approx \frac{N}{lnN}$$

Observação 4.3.1. *A prova deste teorema foge ao escopo deste trabalho, e não será discutida aqui.*

Observação 4.3.2. *Para construir um número primo de cem dígitos, simplesmente geramos aleatoriamente números de cem dígitos e testamos se são primos. O que pode ser feito pelo Teorema dos números primos por*

$$\pi(10^{101}) \approx \frac{10^{101}}{ln10^{101}} \;\; e \;\; \pi(10^{100}) \approx \frac{10^{100}}{ln10^{100}}$$

Assim, chamando de Q a quantidade de primos de cem dígitos e de I a quantidade de números ímpares com cem dígitos, temos:

$$Q = \frac{10^{101}}{ln10^{101}} - \frac{10^{100}}{ln10^{100}} = \frac{100.10^{101} - 101.10^{100}}{100.101ln10}$$

e

$$I = \frac{1}{2}.9.10^{100}$$

Fazendo a probalidade,

$$P(Q/I) = \frac{\frac{100.10^{101} - 101.10^{100}}{100.101ln10}}{\frac{1}{2}.9.10^{100}} \approx 0,0087$$

O que nos assegura que depois de uma média de 115 tentativas, obteríamos um número primo (assumindo que gerássemos apenas números ímpares).

5. O USO DE SOFTWARES COMPUTACIONAIS NO ESTUDO DA CRIPTOGRAFIA

5.1. Introdução

O uso das novas tecnologias de informação e comunicação no ensino deve ter um contexto do processo de ensino-aprendizagem. Assim devemos ter o cuidado ao fazermos o plano de aula e elaborar as atividade de tal forma que se construa um conhecimento em nosso caso matemático no uso dos software computacionais.

Utilizaremos os softwares GeoGebra e Python, porém nosso intuito não é de apresentarmos um curso sobre tais softwares, e sim de utilizarmos arquivos destes como um recurso didático. Apresentaremos elementos necessários para se obter os arquivos a serem utilizados, assim como também uma apresentação dos arquivos. O Python, também será utilizado como uma calculadora.

Recomendamos que se faça o atalho (na área de trabalho) de cada um dos softwares e de cada arquivo indicado, para facilitar a realização dos momentos de aprendizagens (Capítulo 6) .

5.2. O GeoGebra

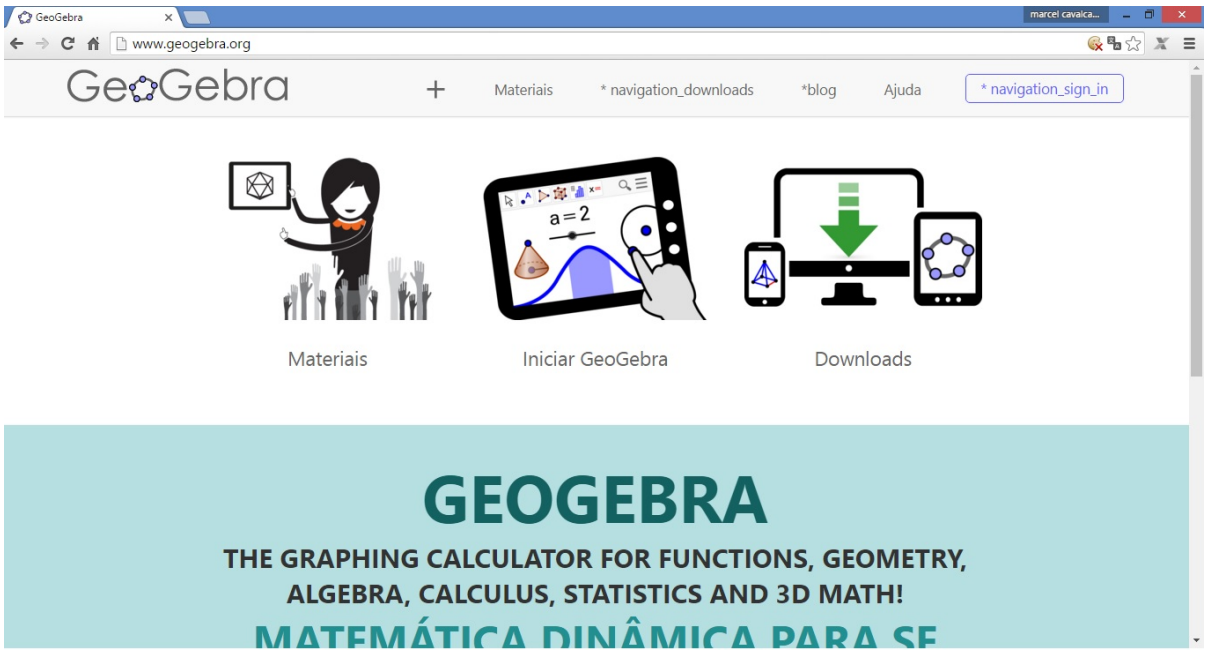
O GeoGebra é um software de matemática dinâmica para todos os níveis de ensino que reúne Geometria, Álgebra, Planilha de Cálculo, Gráficos, Probabilidade, Estatística e Cálculos Simbólicos em um único pacote fácil de se usar. Possui uma comunidade de milhões de usuários em praticamente todos os países. Líder na área de softwares de matemática dinâmica, apoiando o ensino e a aprendizagem em Ciência, Tecnologia, Engenharia e Matemática.

Apesar da facilidade de uso que o software possui, não iremos aqui apresentar um curso sobre o GeoGebra, e sim dar orientações para que se tenham os arquivos necessários para as aplicações no Capítulo 6.

Podemos encontrar o software no site *www.geogebra.org* . Na página principal do site, temos várias janelas de navegação, todas com o intuito de auxiliar no uso e instalação do software. As principais janelas são “materiais” onde dispõe de boa base para o uso, “downloads” onde tem o instalador do software para diversos sistemas operacionais, com todas instruções possíveis para instalação, e “comunidade” onde existem diversas comunidades no mundo com implementações no GeoGebra de variados temas em matemática, física, entre outras ciências.

Ao abrir o site, podemos clicar com o botão direito do mouse, e então selecionar a opção de traduzir para o português. Desta forma teremos como ilustra a figura.

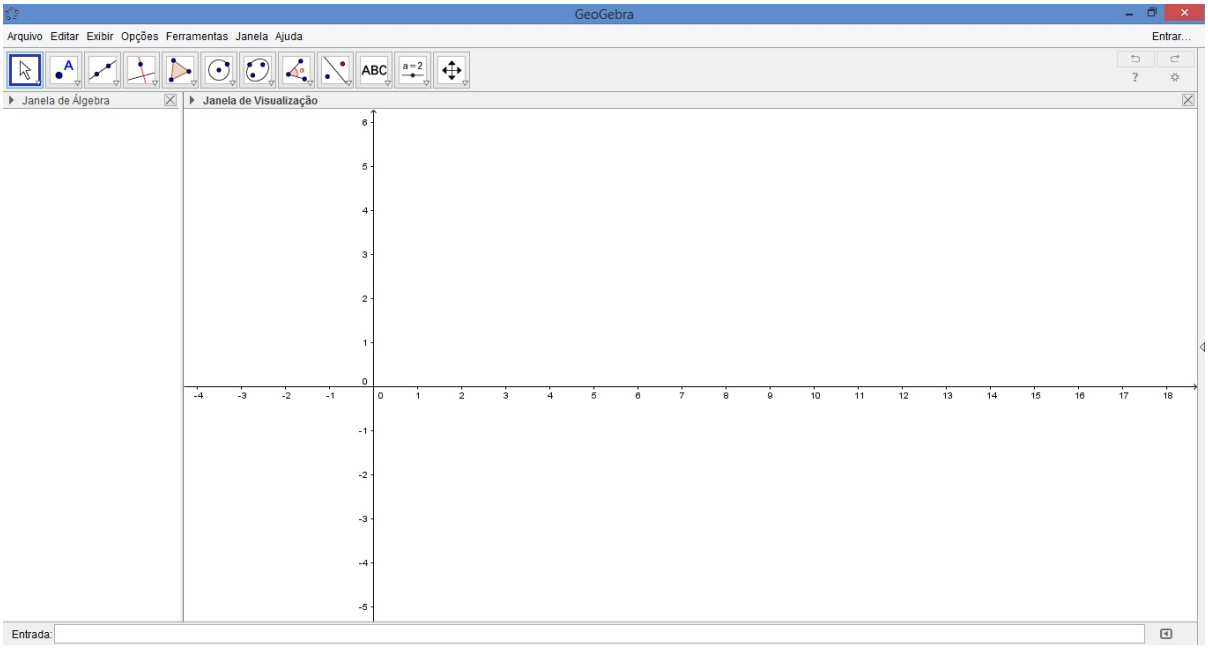
Figura 5.1 Janela de visualização do site do GeoGebra



Fonte: Adaptado de http://www.geogebra.org/ pelo Autor (2015)

Para utilizarmos o software, precisamos baixa-lo clicando no link downloads, em seguida no link que corresponde ao sistema operacional da máquina em trabalho, e então executa-lo. Abrindo-se o arquivo temos a janela de visualização representada como na figura.

Figura 5.2 Janela de visualização do GeoGebra



Fonte: Autor (2015)

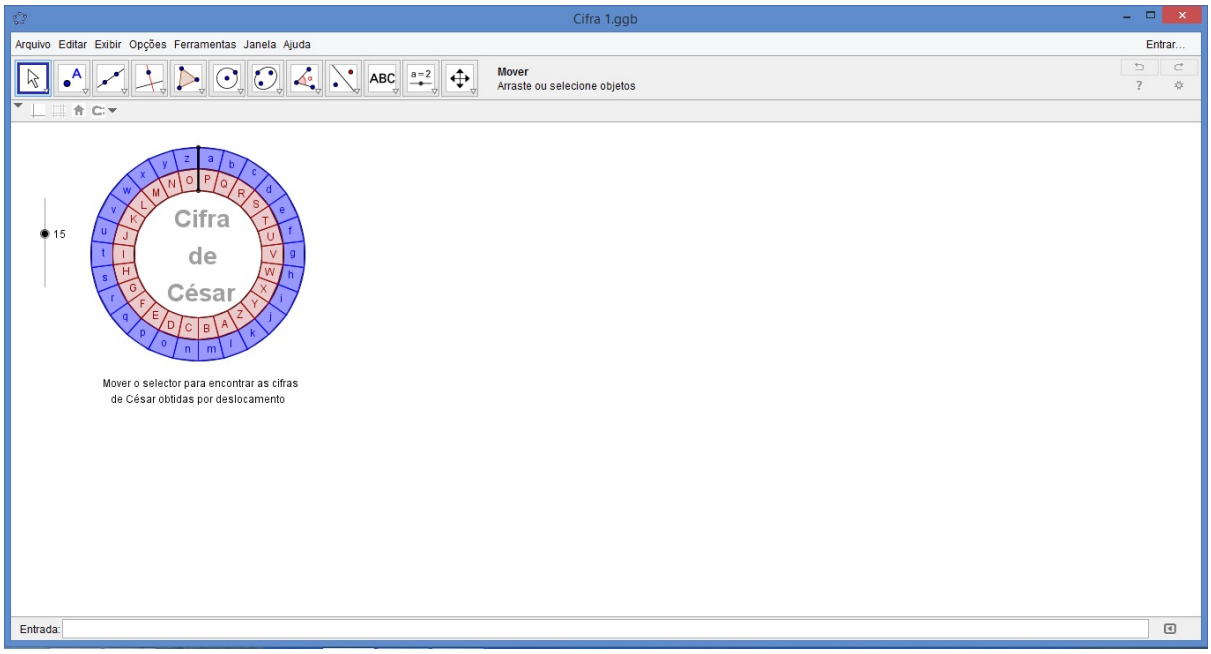
Quanto ao uso de materiais disponibilizados no site, podemos acessa-los clicando no link Materiais. Em seguida, fazendo-se a busca com uma palavra chave, da qual aparecerá algumas estruturas prontas do GeoGebra. Ao clicar em um dos resultados da busca, aparecerá na tela, a escolha, para que seja utilizada online. Podendo-se também ser feito o seu download, clicando-se no último link superior direito, no qual aparecerá uma lista, e então escolhe-se a opção baixar.

5.2.1 Cifra de Cesar

Dentre os materiais implementado no GeoGebra, temos alguns com inspiração na cifra de Cesar, que tem uma grande vantagem de facilitar o uso a encriptação e desencriptação com uma maior agilidade. Um deles é a Roleta de Cesar, o qual adaptamos dando origem aos materiais Cifra 1, Cifra 2 e Cifra 3.

O arquivo Cifra 1, quando aberto, se apresenta conforme a figura.

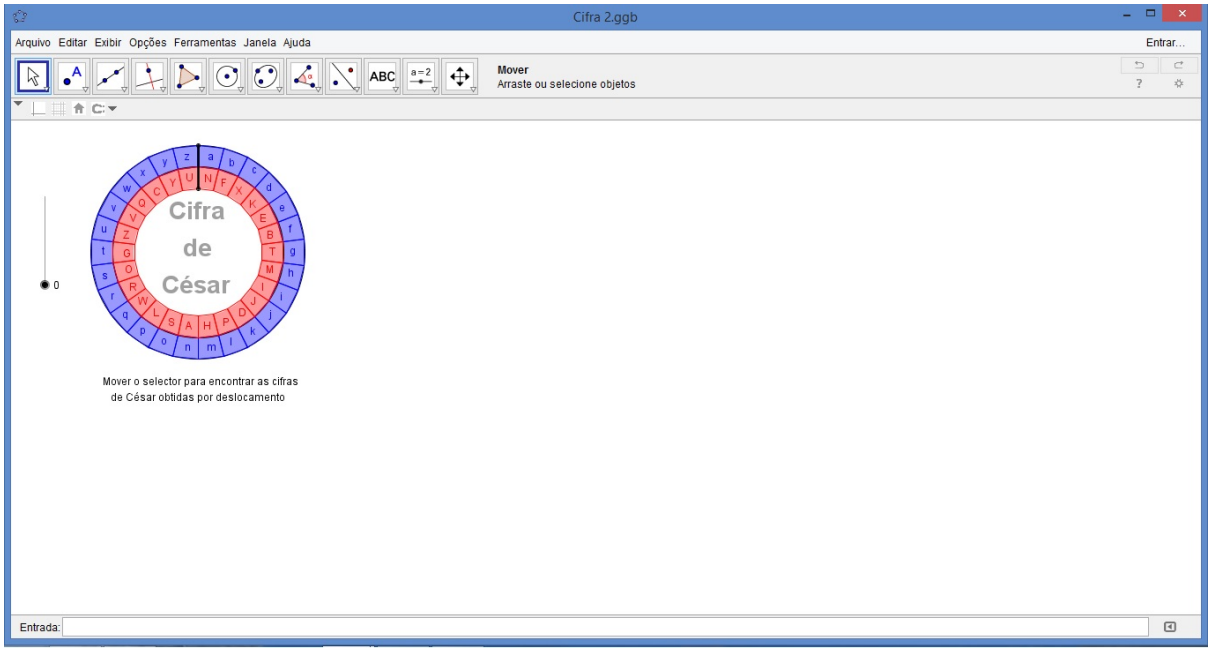
Figura 5.3 Janela de visualização do Cifra 1



Fonte: Autor (2015)

O arquivo Cifra 2, se apresenta como ilustra a figura.

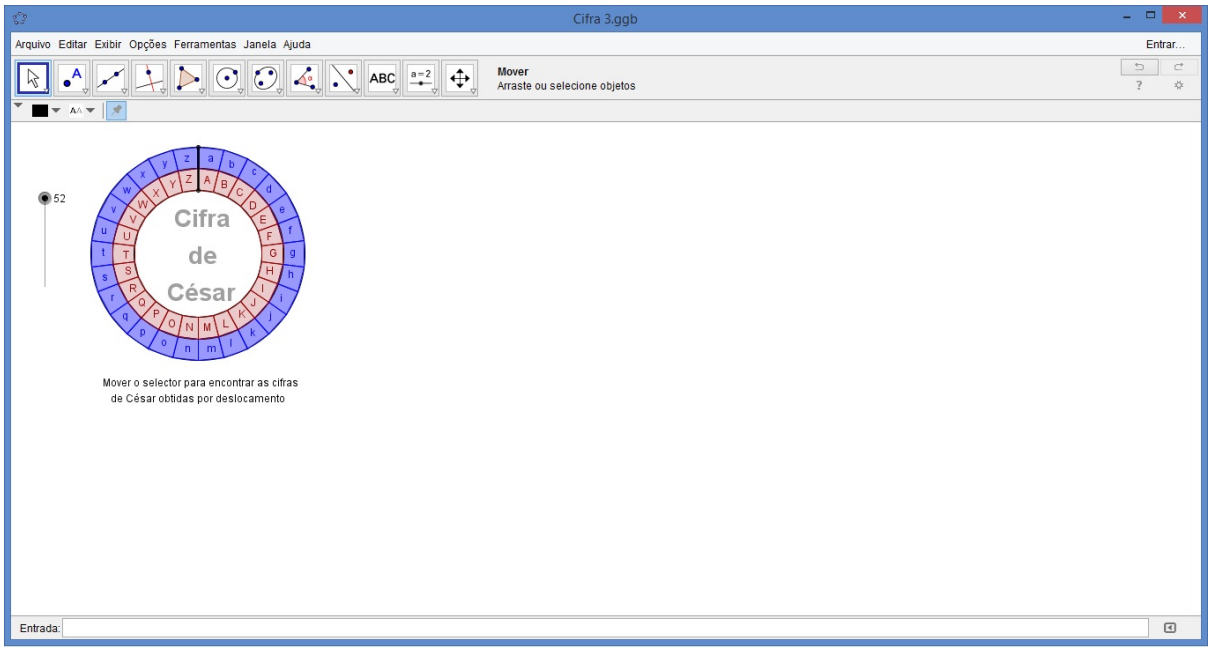
Figura 5.4 Janela de visualização do Cifra 2



Fonte: Autor (2015)

Já o arquivo Cifra 3, conforme a figura.

Figura 5.5 Janela de visualização do Cifra 3



Fonte: Autor (2015)

Cada arquivo possue duas coroas circulares. A maior, que é fixa, tem disposta o alfabeto em sua ordem natural. Já a menor, possui o alfabeto em uma determinada ordem (a depender

do arquivo), e é movimentada (pelo usuário) através do controle deslizante que se encontra ao seu lado.

Comparando-se os arquivo, abertos no GeoGebra, observemos que os arquivos são diferentes. O Cifra 1, além de na coroa menor o alfabeto estar disposto em sua ordem natural, o controle deslizante varia de (0) à (25) (não permitinto haver repetições de cifragens). O Cifra 3 em relação ao Cifra 1 só muda que o controle deslizante varia de (-52) à (52) (permitindo haver repetições de cifragens). Já o Cifra 2 em relação ao Cifra 1 só muda que na coroa menor o alfabeto estar disposto em uma ordem aleatória.

Estes arquivos devem ser baixados, e então salvos na área de trabalho. Daremos os nomes de “Cifra 1.ggb” , “Cifra 2.ggb” e “Cifra 3.ggb”, respctivamente.

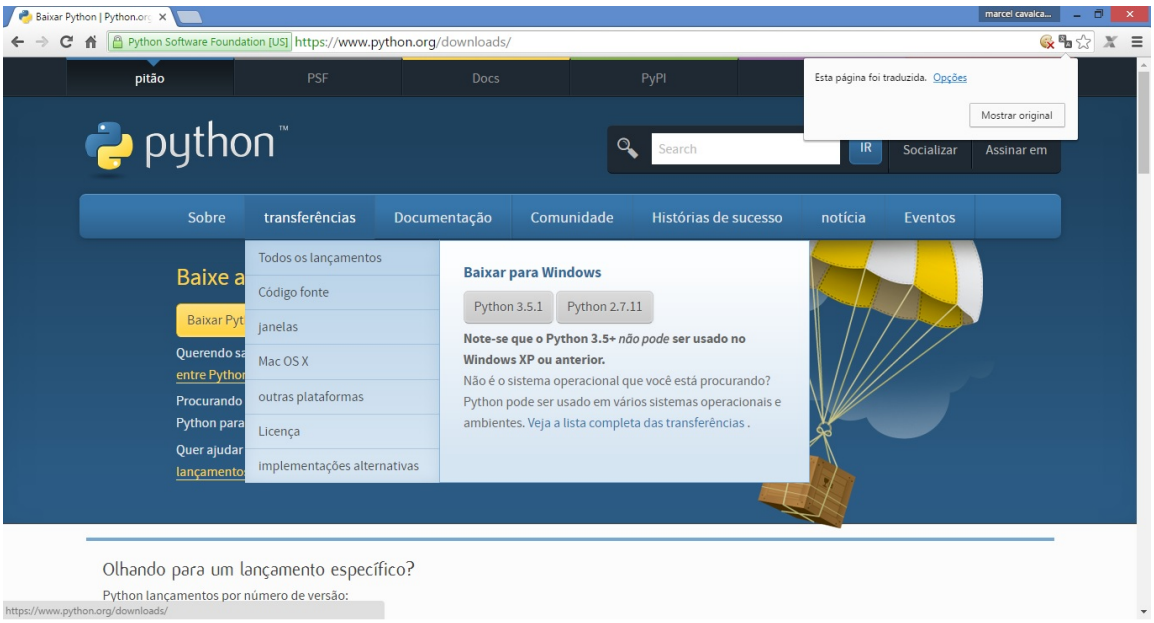
5.3. O Python

Python é uma linguagem de programação criada por Guido van Rossum em 1991. Os objetivos do projeto da linguagem eram produtividade e legibilidade. Em outras palavras, Python é uma linguagem que foi criada para produzir código bom e fácil de manter de maneira rápida.

Para iniciantes, a linguagem oferece a simplicidade, interatividade e várias bibliotecas inclusas. Permitindo que seja possível criar algo interessante e utilizável com grande facilidade. Porém, apesar desta facilidade, não iremos aqui ensinar programação, e sim utilizaremos os programas como recurso didático. Daremos orientações para que se tenha os arquivos necessários para as aplicações no Capítulo 5. Como também daremos orientações para utilização do Python como uma calculadora, para as aplicações no Capítulo 5.

Podemos encontrar o software no site *www.python.org/download* , que ao se solicitar a tradução para português (através do navegador), e se colocar o cursor sobre o link “transferência” aparecerá opções de downloads para diversas plataformas (sistemas operacionais). Veja a ilustração na figura.

Figura 5.6 Janela de visualização do site do Python

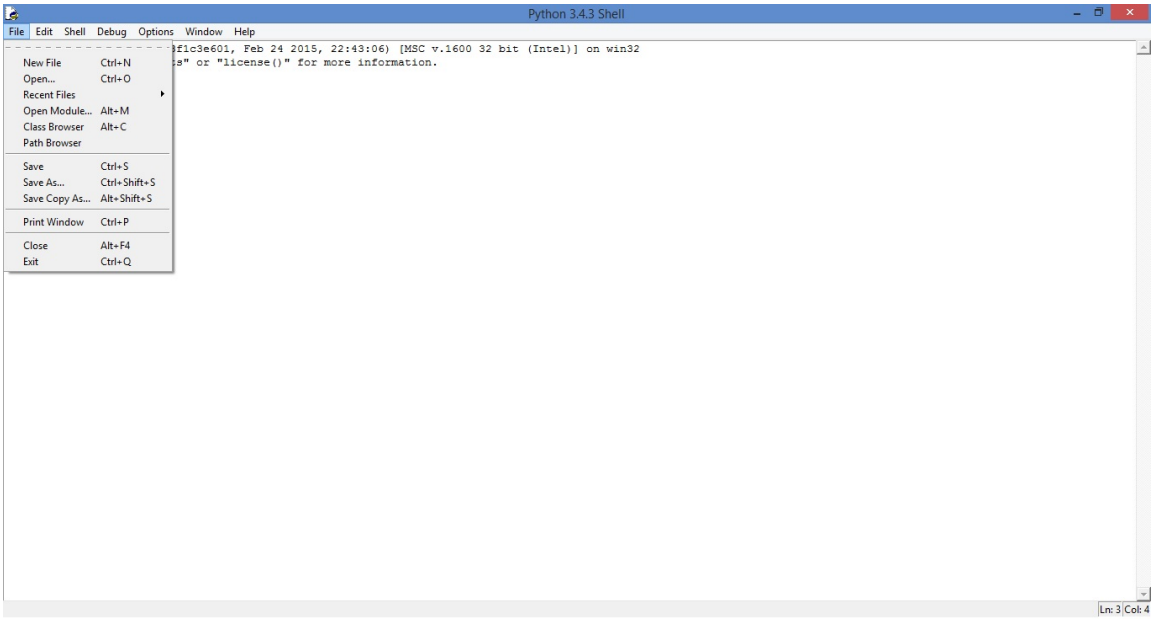


Fonte: Autor (2015)

O IDLE (Integrated DeveLopment Environment) é o ambiente de desenvolvimento Python padrão, que deverá ter vindo junto no pacote do download do Python. Para localizar o IDLE, podemos realizar uma pesquisa por “IDLE” através do recurso pesquisa da plataforma (sistema). Após localizado, é interessante criar um atalho na área de trabalho, para facilitar o trabalho.

Para criar um arquivo em Python, primeiro se deve abrir o IDLE, então ir em “File” e escolher a opção “New File”. Após digitados os comandos do programa, para salvar deve-se ir em “File” e escolher a opção “Save As...”. Para abrir um arquivo, basta ir em “File” e escolher “Open...”. Veja na figura.

Figura 5.7 Janela de visualização do IDLE do Python



Fonte: Autor (2015)

Agora, para o arquivo rodar, é necessário pressionar simultaneamente no teclado “Ctrl” e “F5”.

5.3.1 Fatorar um número com o Python

Apresentaremos um programa que implentamos em Python que mostrou-se enficiente para fatorar um número com até 8 dígitos. O algoritmo criado para este programa está apresentado na figura abaixo.

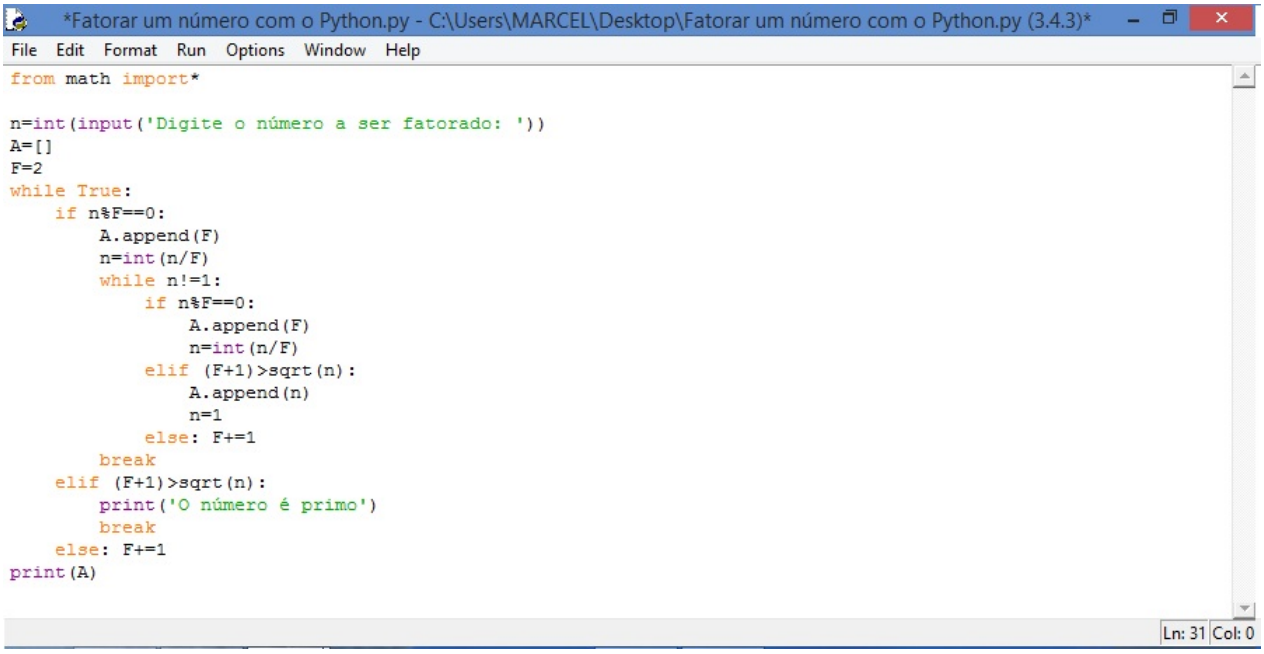
Figura 5.8 Algoritmo

```
Sejam,
    i) A é uma lista vazia.
    ii) F assumindo o primo 2.



Digite o número n a ser fatorado (deve ser inteiro).
Enquanto verdadeiro faça:
    I) Se o resto da divisão de n por F é 0, então:
            1) a lista A recebe o valor de F;
            2) n passa a ser o quociente da divisão de n por F;
            3) enquanto n é diferente de 1:
                3.1) se o resto da divisão de n por F é 0:
                    3.1.1) a lista A recebe F;
                    3.1.2) n passa a ser o quociente da divisão de n por F.
                3.2)senão se (F+1) é maior do que a raíz quadrada de n:
                    3.2.1) a lista A recebe n, pois n é primo;
                    3.2.2) n passa a ser 1, estratégia para parar e sair.
                3.3) senão F é adicionado de 1.
            4) parar e sair.
    II) Senão se (F+1) é maior do que a raíz quadrada de n:
            1) anunciar que o número é primo;
            2) parar e sair.
    III) Senão F é adicionado de 1.
Anunciar a lista A.
```

Fonte: Autor (2015)

Como utilizaremos este programa, para cria-lo é necessário digitar no IDLE, os dados contidos no algoritmo da figura acima, passados para a linguagem do Python, conforme a figura abaixo.

Figura 5.9 Fatorar um número com o Python



Fonte: Autor (2015)

Agora é importante não esquecer de salva-lo como fatoracao, na área de trabalho, assim aparecerá “fatoracao.py”.

5.3.2 Calculadora com o Python

Este recurso é utilizado através do IDLE do Python. É necessário, apenas abri-lo, não devendo ir no “File” abrir mais nada. Agora é necessário conhecer algumas operações. Como mostra a figura.

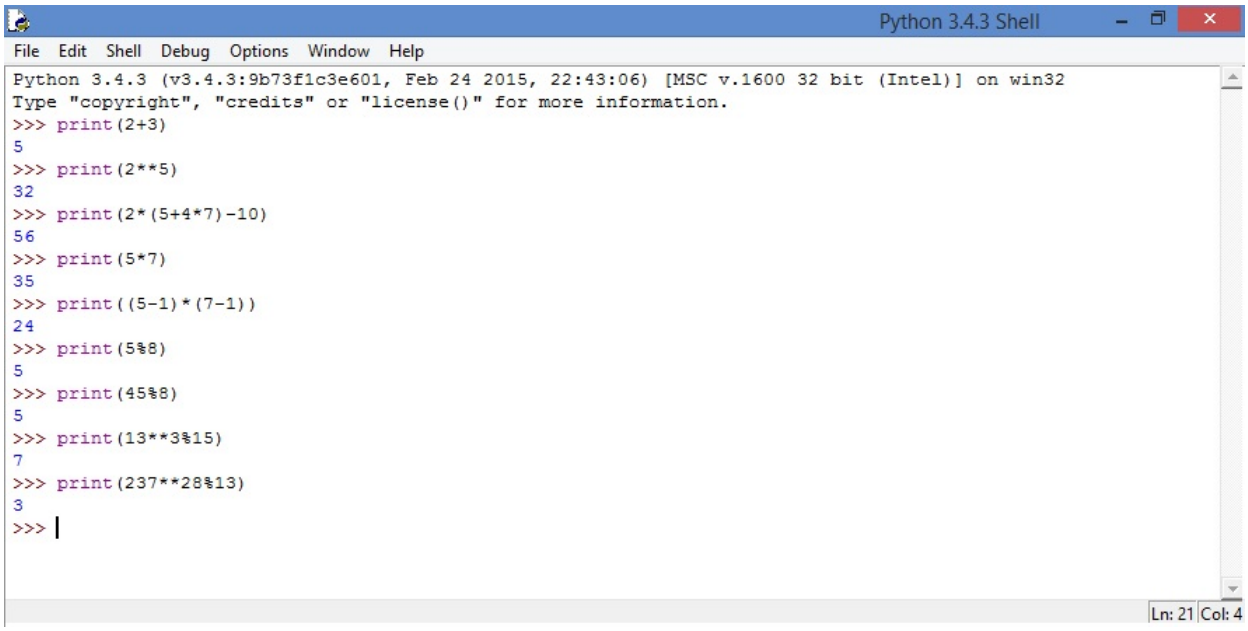
Figura 5.10 Operações em Python

```
(+)  adição
(-)  subtração
(*)  multiplicação
(/)  divisão
(**) potencia
(%)  congruencia modular
( )  único símbolo de agrupamento
```

Fonte: Autor (2015)

Para calcular é necessário digitar a expressão numérica dentro dos parenteses do comando “print( )”, e então pressionar a tecla enter, assim será apresentado o resultado na linha logo abaixo. Vejamos alguns exemplos na figura abaixo.

Figura 5.11 Calculadora com o Python



Fonte: Autor (2015)

6. MOMENTOS DE APRENDIZAGENS

6.1. Introdução

Iremos apresentar em cada momento de aprendizagem alguns conceitos sobre criptografia para o uso de codificação e decodificação no ensino básico.

A abordagem de conteúdos que estimulem a curiosidade e que despertem o processo de ensino-aprendizagem e a construção de novos conhecimentos é importantíssimo no ensino de matemática, sobretudo com o uso de novas tecnologias, pois esta metodologia vem despertando o aluno a cada dia e de forma eficaz.

Algumas reflexões podem ser feitas a respeito da importância do tema criptografia no currículo de matemática do ensino médio e para nos apoiarmos veremos o que diz a Lei de Diretrizes e Bases da Educação Nacional [BRASIL, lei 9394/96] que apresenta o ensino médio com as seguintes finalidades:

- a consolidação e o aprofundamento dos conhecimentos adquiridos no ensino fundamental, possibilitando o prosseguimento de estudos;
- a preparação básica para o trabalho e a cidadania do educando, para continuar aprendendo, de modo a ser capaz de se adaptar com flexibilidade a novas condições de ocupação ou aperfeiçoamento posteriores;
- o aprimoramento do educando como pessoa humana, incluindo a formação ética e o desenvolvimento da autonomia intelectual e do pensamento crítico;
- a compreensão dos fundamentos científico-tecnológicos dos processos produtivos, relacionando a teoria com a prática, no ensino de cada disciplina.

Assim na etapa final da educação básica, espera-se que o estudante esteja preparado para atuar na sociedade, na qual está inserido, de forma efetiva, sabendo se comunicar claramente, resolver problemas do dia a dia e do trabalho, tomar decisões, trabalhar com eficiência e em cooperação.

6.2. Proposta Pedagógica

Nas últimas décadas surgiu uma tendência denominada de formação de professores reflexivos, que visa analisar a atual situação da formação docente. Nessa perspectiva, a do professor reflexivo, as reflexões a respeito da prática educativa e da criação de novas metodologias, devem ir ao encontro de novas possibilidades de formação profissional.

Muitos profissionais acreditam que, para serem bons professores, é necessário apenas que conheçam o conteúdo a ser ensinado, sem preocupação com outras dimensões de um processo tão complexo que é o de ensino e aprendizagem.

Para ter sucesso em nossa proposta usaremos o ensino construtivista, este inspirado nas ideias do suíço Jean Piaget $(1896 - 1980)$. O método procura instigar a curiosidade, fazendo

a aprendizagem realiza-se por meio da resolução de problemas, de questões desafiadoras ou motivadoras.

O objetivo do construtivismo é que o aluno adquira autonomia, dando ênfase ao aspecto cognitivo do tema abordado, ou seja, faz-se com que o aluno tenha entusiasmo em aprender determinado assunto e com isso obtenha avanço no ensino-aprendizagem.

Para motivar o tema, vamos solicitar a intervenção do professor de história. Sugerindo a ele, que coloque a turma para assistir o filme “The imitation game". Após, discuta sobre as principais guerras da humanidade, destacado a questão da criptografia.

PLANO DE AULA

IDENTIFICAÇÃO
COMPONENTE CURRICULAR: Matemática.
CONTEÚDO: Criptografia.
DURAÇÃO: 10 horas/aulas.
PÚBLICO ALVO: Educandos do terceiro ano do ensino médio público.

OBJETIVO GERAL
Criar condições, para que o educando possa compreender os aspectos fundamentais do estudo de criptografia.

OBJETIVO ESPECÍFICO
Incentivar o educando a buscar o conhecimento necessário, para manipular com segurança, as propriedades que concerne a codificação e decodificação de informações, propiciando a este um novo instrumento matemático capaz de compor soluções de situações-problemas encontrados em seu cotidiano.

DISCRIMINAÇÃO DOS CONTEÚDOS

a) contexto histórico;

b) cifra de César;

c) conceitos iniciais de criptografia e aritmética modular;

d) GeoGebra e softwares online, no ensino de criptografia;

e) criptografia RSA;

f) Python, auxiliando o ensino de aritmética modular.

METODOLOGIA

- Momento 1 (2 horas/aulas)

Tendo já ocorrido a intervenção do professor de história, com o filme “The imitation game", a aula iniciará com a aplicação da situação problema 1. Logo após inicia-se um debate entre alunos e professor a respeito do problema. Deve-se dar destaque à importância do resto na divisão euclidiana, assim como quociente, correlacionando com o efeito cíclico.

Feito o debate a respeito do problema 1, o professor aplica cada um dos demais problemas, seguido da socialização os alunos.

A aula termina com uma exposição sobre o uso de tecnologias na criptografia, sobretudo do uso do GeoGebra na criptografia, dando preparo para a ocorrência do Momento 2.

- Momento 2 (2 horas/aulas)

Utilizaremos os arquivos: ´´Cifra 1.ggb", ´´Cifra 2.ggb"e ´´Cifra 3.ggb". Estes devem ser baixados conforme orienta o Capítulo 5, e então, salvos na área de trabalho.

A aula iniciará com a aplicação da situações problema 1, que utiliza os arquivos citato logo acima. Terminada da ativitidade, vem a socialização entre professor e alunos. Nesta, o professor além de fazer correlação com o Momento 1, deve aprensenta-los a decodificação por frequência.

Após o término do debate sobre o Problema 1, o professor faz uma exposição (com o data show) do vídeo “Demonstração Máquina Enigma", disponível em

$$www.ufrgs.br/museu/comunicacao/divulgacao/audiovisual/maquina-enigma-1$$

Aplica-se a situação problema 2 com o auxílio do site *www.amenigma.com*, e então, faz-se mais uma rodada de socialização.

- Momento 3 (2horas/aulas)

A aula iniciará com a aplicação da situação problema 1, que foi retirado das Olimpíadas Brasileiras das Escolas Públicas (OBMEP), e que tem um papel importantíssimo para introduzir o tema de congruências modulares. Terminado o tempo para a realização do problema, há a solialização, onde o professor deve orienta-la correlacionando com os Momentos 1 e 2.

Após a discussão do problema 1, o professor inicia uma apresentação (com o data show), onde mostrará alguns aspectos inerentes à congruencia modular, como: definição, propriedades, relação com números relativamente primos e exemplos.

Aplica-se o problema 2. Com o momento de reflexão dos alunos findado, culmina-se com a socialização de todos.

Segue-se com a aplicação a situação problema 3, e sua rodada de socialização.

Propõe-se uma atividade extra para casa.

- Momento 4 (2horas/aulas)

A aula iniciará com a aplicação do problema 1, este com intuito de apresentar o matemático Leonard Euler e a função ϕ de euler. Assim como fixar o aprendizado quanto a função ϕ de Euler, e convencer que é possível fatorar n, conhecendo-se n e $\phi(n)$. Terminado, há a socialização.

Vem a aplicação do problema 2, que introduz para o aluno uma relação importante do processo de encriptação e desencriptação. Essa relação guarda o segredo do por que o processo de criptografia RSA funcionar tão bem. Socializa-se

No problema 3 o aluno irá fazer a encriptação, usando a linha pedida no problema. Há a socialização.

Já no problema 4, o aluno irá fazer a desencriptação do exercício anterior, usando para tal a chave encontrada no problema 2. Socializa-se.

O professor apresenta (com o data show) o processo de criptografia RSA, através de um exemplo. E então depois entra-se em debate com todas as atividades do momento em questão.

- Momento 5 (2horas/aulas)

Utilizaremos o arquivo $fatoracao.py$ e o Software IDLE (Python), orientados no Capítulo 4, e então, salvos na área de trabalho.

O professor iniciará a aula apresentando (com data show) como abrir e manusear o arquivo $fatoracao.py$. O mesmo para a calculadora no Python.

Segue-se com a aplicação do problema 1, que utiliza o software Python para auxiliar em fatorações, e mostra que o processar da fatoração é duradouro a depender do número. Socializa-se.

Logo depois vem a aplicação do problema 2 que contextualiza o processo de criptografia RSA, utilizando o software Python. Socializa-se

Propõe-se a atividade extra para casa.

RECURSOS

a) Quadro branco e pincel;

b) data show;

c) lista de exercício;

d) computador com software GeoGebra e Python instalado.

AVALIAÇÃO

Os educandos estarão sendo avaliados quanto:

a) Ao envolvimento com a aula;;

b) A coerência ou não, das ideias e conceitos por eles apresentados a respeito do tema em discussão;

c) Participação em grupo para resolução de listas de exercícios.

6.3. Momento 1

Aqui será apresentado alguns conceitos iniciais para o uso de codificação no dia a dia, tais conceitos vão ser abordados seguindo a teoria construtivista. Os alunos irão inicialmente tentar resolver cada problema e logo depois de cada problema o professor entra em cena com alguns conceitos extras e uma socialização com todos.

Problema 1

O primeiro código secreto que se tem notícia foi utilizado pelo militar e governante romano Júlio César (100 a.E.C. - 44 a.E.C.), na época da transição do final do período republicano da Roma Antiga. Tal metodogia de criptografia ficou conhecida como a cifra de Cesar, na qual cada letra do alfabeto é substituida por uma letra subsequente. Seguindo a mesma ideia a tabela abaixo cada letra minúscula da primeira linha esta associada a uma letra maiúscula da segunda

linha, sendo a letra *a* da primeira linha associada a letra *D* da segunda linha, a *b* com a *F* e assim sucessivamente.

Figura 6.1 tabela

a	b	c	d	e	f	g	h	i	j	k	l	m
D	E	F	G	H	I	J	K	L	M	N	O	P

n	o	p	q	r	s	t	u	v	w	x	y	z
Q	R	S	T	U	V	W	X	Y	Z	A	B	C

Fonte: Autor (2015)

Figura 6.2 tabela

a	b	c	d	e	f	g	h	i	j	k	l	m
C												

n	o	p	q	r	s	t	u	v	w	x	y	z

Fonte: Autor (2015)

i) A letra *f* mudou em quantas posições comparada com a primeira e segunda linha?

ii) O que ocorre se a letra **a** mudar em 26 posições? E se for 52? E sendo 28?

iii) Para cada um dos valores dados no item anterior, realize a divisão euclidiana por 26. O que se pode concluir ao se comparar com a sua resposta do item (ii)?

Problema 2

Use a tabela da Figura 5.2 e faça a codificação da frase“estou aprendendo matematica".

Problema 3

Use 99 para espaço em palavras e a tabela abaixo para fazer a codificação da frase “estudar é importante"

Figura 6.3 tabela

a	b	c	d	e	f	g	h	i	j	k	l	m
10	11	12	13	14	15	16	17	18	19	20	21	22

n	o	p	q	r	s	t	u	v	w	x	y	z
23	24	25	26	27	28	29	30	31	32	33	34	35

Fonte: Autor (2015)

Problema 4

Com a mesma tabela do problema anterior, decifre “1428292430991025271423131423132499122718252924162710151810"

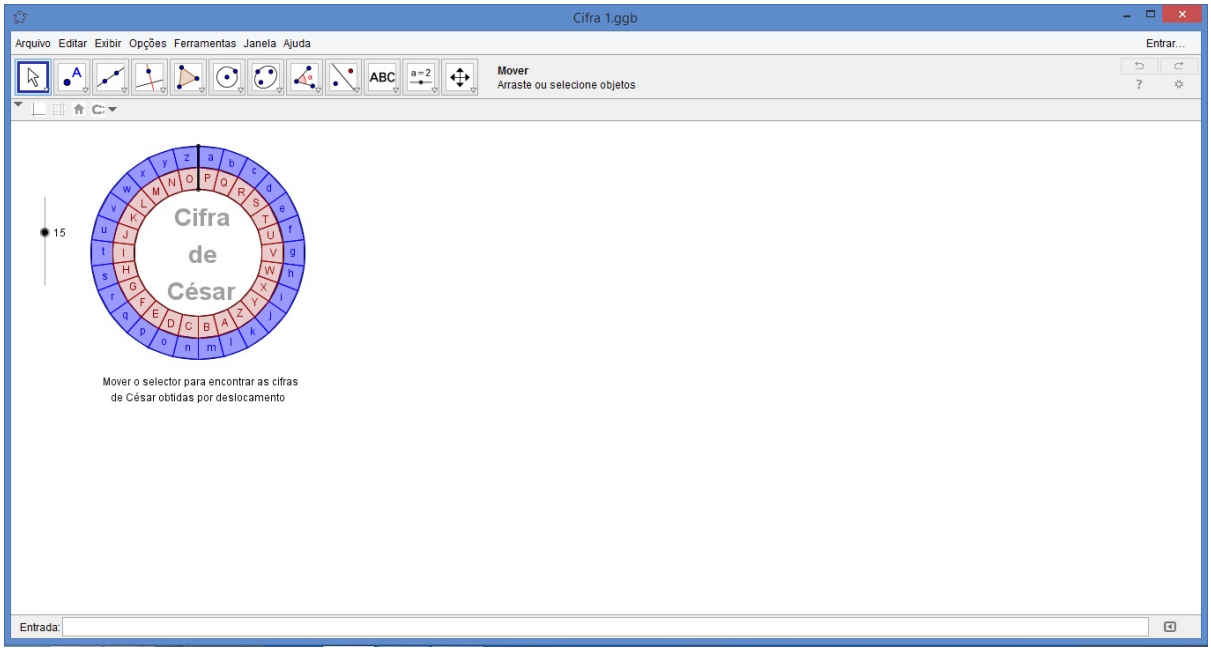
6.4. Momento 2

Dando continuidade ao tema, nesta aula 2, podemos observar a importância do ensino de criptografia com o uso do software computacional GeoGebra e perbecemos que desta forma aumentamos o grau de ensino-aprendizagem.

Problema 1

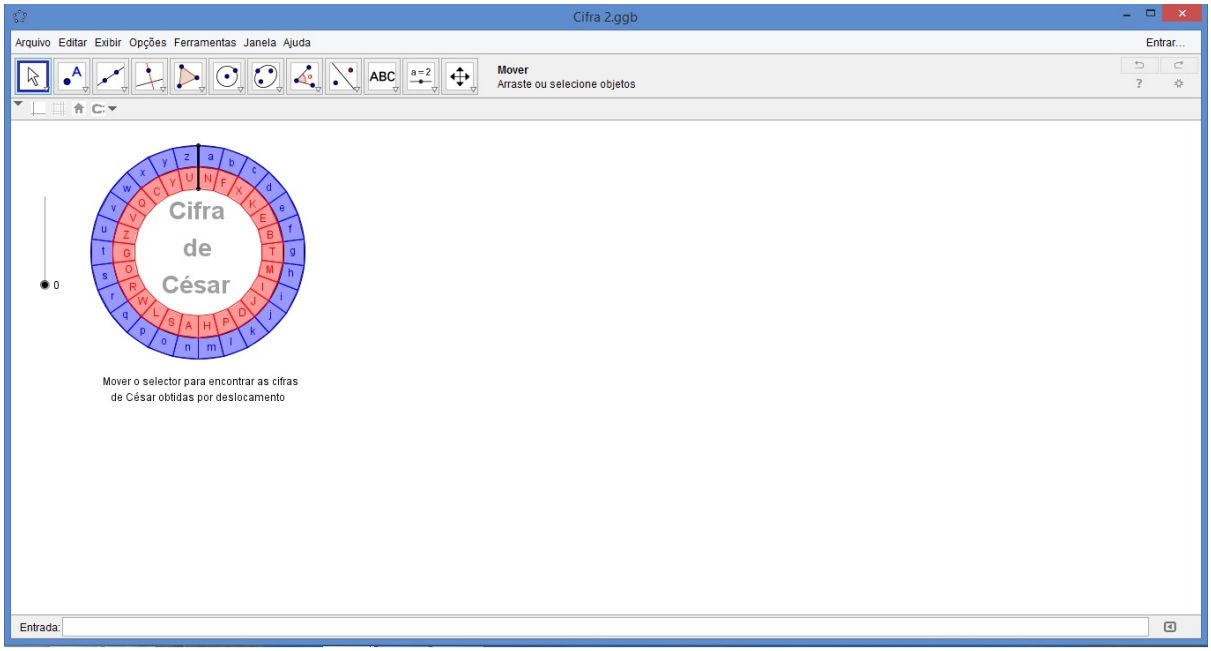
Abra o arquivo “Cifra 1"na área de trabalho de seu computador. Faça a codificação do nome de sua escola usando o software GeoGebra. Caso use o ícone “Cifra 2"qual seria a nova codificação do nome de sua escola e sendo com o ícone “Cifra 3"? Qual a diferença entre os três casos?

Figura 6.4 CIFRA 1



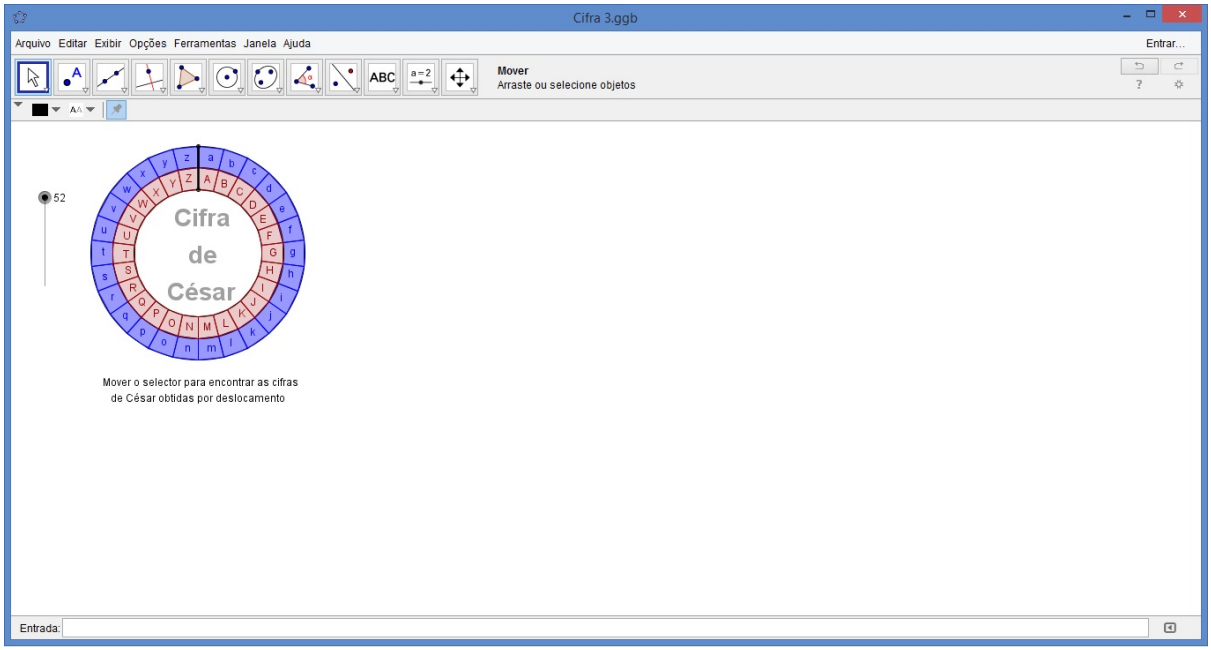
Fonte: Autor (2015)

Figura 6.5 CIFRA 2



Fonte: Autor (2015)

Figura 6.6 CIFRA 3



Fonte: Autor (2015)

Problema 2

Codifique o seu nome, utilizando a Máquima Enigma online, disponível em:

http : *//www.amenigma.com*

Figura 6.7 Máquina Enigma

Fonte: Autor (2015)

6.5. Momento 3

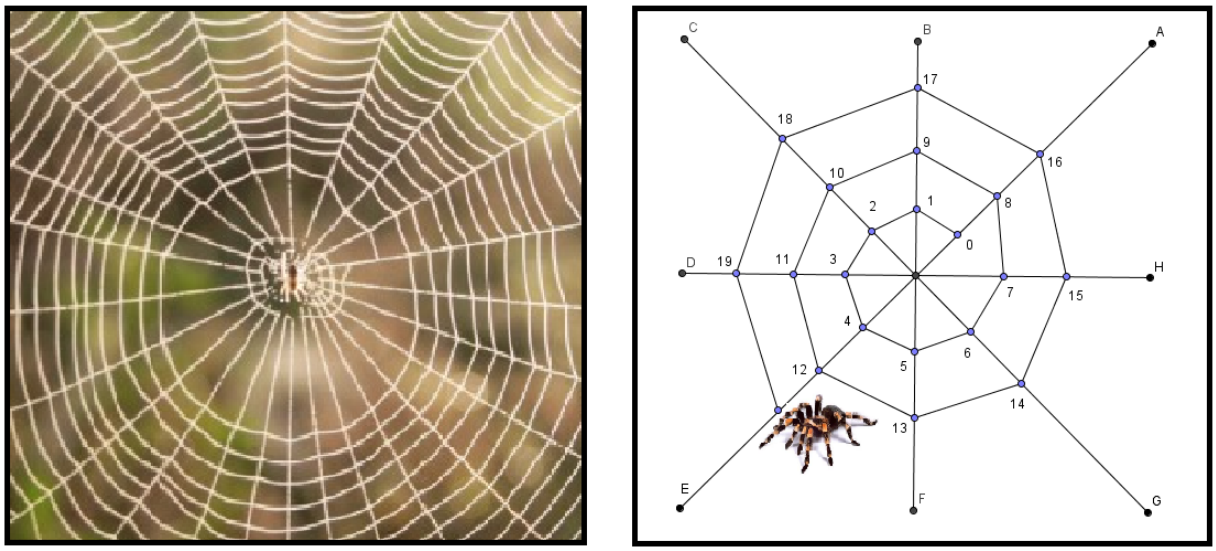
Este momento tem o objetivo de introduzir e fixar as ideias de aritmética modular, assim como inteligar o tema ao estudo de criptografia.

Vamos apresentar inicialmente uma questão retirada do banco de questões do site da OBMEP (Olimpíadas Brasileira de Matemática das Escola Públicas).

Problema 1

A, B, C, D, E, F, G e **H** são os fios de apoio que uma aranha usa para construir sua teia, conforme mostra a figura. A aranha continua seu trabalho.

Figura 6.8 teia de aranha



Fonte: Autor (2015)

Sobre qual fio de apoio estará o número 118?

Problema 2

Ache o resto da divisão

a) de 7^{10} por 50;

b) de 2^{100} por 11;

c) de 5^{21} por 127;

O próximo problema que foi uma adaptação de um retirado de [Rousseau & Saint-Aubin, 2015], tem um caráter considerável para o ensino-aprendizagem de criptografia interligada a aritmética modular.

Problema 3

Apresentamos um sistema de criptografia simples. O espaço entre as palavras é representado pelo número 0, enquanto 27 corresponde ao ponto e 28 à vírgula. As letras $A, \ldots, Z$ são representadas pelos números 1, ..., 26, conforme organizada na seguinte tabela:

Figura 6.9 tabela

a	b	c	d	e	f	g	h	i	j	k	l	m
1	2	3	4	5	6	7	8	9	10	11	12	13

n	o	p	q	r	s	t	u	v	w	x	y	z
14	15	16	17	18	19	20	21	22	23	24	25	26

Fonte: Autor (2015)

Eis como codificar uma palavra:

- substituimos os símbolos pelos números associados;
- multiplicamos cada número por 2;
- reduzimos cada resultado módulo 29;
- mapeamos cada número de volta a seu símbolo correspondente, resultando numa palavra codificada.

Por exemplo, para codificar a palavra “PAR", primeiro a mapeamos para a sequência 16,1,18. Dobramos estes números, obtemos 32,2,36, restando 3,2,5 após a redução módulo 29. Substituindo os inteiros pelos símbolos associados leva à codificação final de “CBE".

a) Codifique a palavra “SIM".

b) Explique por que esta codificação é reversível, e como proceder para decodificá-la.

c) Decodifique a palavra “IAZB".

Problema Extra

O objetivo desta questão é mostrar que nenhum número da forma $4n+3$ pode ser escrito como a soma dos quadrados de dois inteiros.

(1) Mostre que o quadrado de qualquer inteiro só ser pode congruente a 0 ou 1 módulo 4.

(2) Use (1) para mostrar que se x e y são inteiros então x^2+y^2 só pode ser congruente a 0, 1 ou 2 módulo 4.

(3) Use (2) para mostrar que um inteiro da forma $4n+3$ não pode ser escrito como soma de dois quadrados de inteiros.

Este resultado é um caso particular de um teorema comunicado por Fermat em uma carta a Roberval datada de 1640. Fermat também sabia que qualquer primo da forma $4n+1$ pode ser escrito como soma de dois quadrados de inteiros.

6.6. Momento 4

O momento presente tem um papel central para o entendimento do funcionamento do sistema de criptografia RSA.

Problema 1

A função $\phi(n)$ foi desenvolvida pelo matemático e físico suiço Leonhard Euler (1707-1783), é definida como a quantidade de números naturais $0, 1, ..., n-1$ que são primos com natural n. Por exemplo, para $n = 10$, temos $\phi(10) = 4$, pois os números $1,\ 3,\ 7$ e 9 são menores que 10 e primos com 10. Sabe-se que para $n = p.q$, ou seja, o produto de dois números primos, tem-se que $\phi(n) = (p-1).(q-1)$. Diante disso, para $n = 851$, sendo o produto de dois primos, com $\phi(851) = 792$. Determine a forma fatorada de 851.

Problema 2

Um sistema de criptografia de chave pública é inicialmente estabelecido pela pessoa (ou organização), que chamaremos de receptor, que quer receber mensagens de uma maneira segura. É o receptor que estabelece o sistema e publica como enviar suas mensagens.
Considere que Bob deseja receber uma informação sigilosa de Alice. Estabelecido um sistema de criptografia RSA com $n = 5 \times 13 = 65$ e com a chave encriptação $e = 29$ (pública) . Encontre a chave de desencriptação d (privada) que satisfaça $e.d \equiv 1\ mod\ \phi(n)$.

Problema 3

A chave de desencriptação d encontrada por Bob será mantida em seu poder, assim como, a fatoração de n. O que é uma grande dificuldade, pois no meio real os primos usados na fatoração de n são muito grandes e dessa forma fica impossível em tempo hábil calcular a chave d.
Bob (receptor) pública então as chaves $n = 65$ e $e = 29$. Alice (emissora) deseja enviar a mensagem $m = 24$ para Bob e irá usar as chaves publicada por ele. Ajude Alice, calculando a mensagem criptografada "a"tal que $a \equiv m^e\ mod\ n$.

Problema 4

Alice envia a mensagem "a"para Bob. Para ele conseguir voltar para mensagem real "m "vai precisar de sua chave de desincriptação d. Ajude Bob, calculando a desincriptação da mensagem, ou seja, encontrar m tal que $m \equiv a^d\ mod\ n$.

6.7. Momento 5

O momento presente tem um papel de concretizar o aprendizado de fatoração e da criptografia RSA com o uso do software Python.

Problema 1

Utilizando-se do arquivo “fatoracao.py” , encontre os fatores primos de:

a) 7927001

b) 353093680447

c) 14371749415924438039

Em algum caso ocorreu algum problema? Se sim, descreva o problema, e explique o por quê.

O próximo problema foi adaptado de [COUTINHO, S.C, 2011], tem um papel central no entendimento do sistema de criptografia RSA.

Problema 2

A chave pública utilizada pelo banco de Toulouse para codificar suas mensagens é a seguinte: $n = 10403$ e $e = 8743$. Recentemente os computadores do banco receberam, de local indeterminado, a seguinte mensagem:

$$4746 - 8214 - 9372 - 9009 - 4453 - 8198$$

Usando as ferramentas do Python para facilitar os cálculos, responda:

a) Qual a fatoração do número 10403?

b) Quanto vale $\phi(n)$?

c) Calcule a chave de desencriptação.

d) Desencripte a mensagem recebida pelo banco.

Problema Extra

Escreva sobre o sistema criptográfico RSA.

7. CONSIDERAÇÕES FINAIS

Este trabalho teve como objetivo mostrar que a criptografia RSA pode ser trabalhada no ensino básico. A maneira diferente como são trabalhadas as sequencias do tema caracterizam um ensino motivador que contribuirá para o sucesso no ensino da Matemática.

Essa proposta, é importante, pois incentiva o professor trabalhar a realidade dos alunos, ensinando-os a desenvolver habilidades com o seu cotidiano de maneira excepcional pois usa ferramentas computacionais, assim, permite aos alunos a criticidade, tão diferente da abstração, porém também importante. Saber trabalhar essas questões é de fundamental importância para o sucesso no ensino de criptografia no ensino básico.

Acreditamos que esta pesquisa contribuirá para uma melhor exposição das práticas de muitos profissionais envolvidos com o ensino da Matemática e servirá para discutimos qual o melhor caminho para fazer com que os alunos diminuam as dificuldades nesta disciplina.

REFERÊNCIAS

BRASIL, P.C.N. **Parâmetros Curriculares Nacionais: Ensino Médio**. Matemática. Brasília: MEC/SEF, 1999.

COUTINHO, S.C. **Números Inteiros e Criptografia RSA**. 2ª Ed. Rio de Janeiro: IMPA, 2011.

EVARISTO, J. \& PERDIGÃO, E. **Introdução à Álgebra Abstrata**. 1ª Ed. Maceió: edufal, 2002.

HEFEZ, A. **Elementos de Aritmética**. 2ª edição: Rio de Janeiro, SBM 2011.

LEMOS, M. **Criptografia, números primos e algoritmos** (publicações matemáticas do impa). 4ª Ed. Rio de Janeiro: IMPA, 2010.

OLGIN, C.A. **Currículo no Ensino Médio: uma experiência com o tema criptografia**. Dissertação. Canoas- RS, 2011.

RIBENBOIM, P. **Números Primos: Velhos Mistérios e Novos Recordes**. 1ª edição: Rio de Janeiro, IMPA, 2012.

ROUSSEAU, C. \& SAINT-AUBIN, Y. **Matemática e Atualidade: volume 1**. 1ª edição: Rio de Janeiro, SBM, 2015.

SANTOS, J.P.O. **Introdução à Teoria dos Números**. 3ª edição: Rio de Janeiro, IMPA, 2005.