



# Programa de Pós-Graduação em Matemática em Rede Nacional

## **Decomposição de números fatoriais em fatores primos: algoritmos para situações complementares ao Teorema de Legendre**

**Leonardo Henrique Melo de Carvalho**



**PROFMAT**

Rio Claro  
2022



UNIVERSIDADE ESTADUAL PAULISTA “JÚLIO DE MESQUITA FILHO”  
Instituto de Geociências e Ciências Exatas  
Câmpus de Rio Claro

# Decomposição de números fatoriais em fatores primos: algoritmos para situações complementares ao Teorema de Legendre

**Leonardo Henrique Melo de Carvalho**

Dissertação apresentada como parte dos requisitos para obtenção do título de Mestre em Matemática, junto ao Programa de Pós-Graduação – Mestrado Profissional em Matemática em Rede Nacional, do Instituto de Geociências e Ciências Exatas da Universidade Estadual Paulista “Júlio de Mesquita Filho”, Câmpus de Rio Claro.

Orientador  
**Prof. Dr. Jamil Viana Pereira**

**Rio Claro**  
**2022**

C331d

Carvalho, Leonardo Henrique Melo de

Decomposição de números fatoriais em fatores primos: : algoritmos para situações complementares ao Teorema de Legendre / Leonardo Henrique Melo de Carvalho. -- Rio Claro, 2022

90 p. : il., tabs. + 1 CD-ROM

Dissertação (mestrado profissional) - Universidade Estadual Paulista (Unesp), Instituto de Geociências e Ciências Exatas, Rio Claro

Orientador: Jamil Viana Pereira]

1. Legendre. 2. Fatores primos. 3. Teorema Fundamental da Aritmética. 4. Aritmética. 5. Teorema de Legendre. I. Título.

Sistema de geração automática de fichas catalográficas da Unesp. Biblioteca do Instituto de Geociências e Ciências Exatas, Rio Claro. Dados fornecidos pelo autor(a).

Essa ficha não pode ser modificada.

# TERMO DE APROVAÇÃO

Leonardo Henrique Melo de Carvalho

## DECOMPOSIÇÃO DE NÚMEROS FATORIAIS EM FATORES PRIMOS: ALGORITMOS PARA SITUAÇÕES COMPLEMENTARES AO TEOREMA DE LEGENDRE

Dissertação APROVADA como requisito parcial para a obtenção do grau de Mestre no Curso de Pós-Graduação – Mestrado Profissional em Matemática em Rede Nacional, do Instituto de Geociências e Ciências Exatas da Universidade Estadual Paulista “Júlio de Mesquita Filho”, pela seguinte banca examinadora:

---

Prof. Dr. Jamil Viana Pereira  
Orientador

---

Prof. Dr. Robinson Alves Lemos  
FACSAL/UNEMAT/Tangará da Serra (MT)

---

Profa. Dra. Suzete Maria Silva Afonso  
IGCE/UNESP/Rio Claro (SP)

**Rio Claro, 05 de abril de 2022**

# Agradecimentos

De início, agradeço minha família pelo tempo e o amor empregado para que esse trabalho fosse possível. Vocês mais do que ninguém representam o que há de mais vital e importante em minha vida. Em especial, deixo agradecimentos a Lucilene de Carvalho (mãe), João Carlos Batista de Carvalho (pai), Israel Henrique de Carvalho (irmão) e Luzia Toterio Carvalho (avó).

Segundo, não poderia deixar de agradecer a minha esposa Ana Paula Brandão Carvalho de Melo. Por mais, que ainda seja curto o caminho que juntos percorremos, este por si só já é capaz de concretizar uma certeza... somos almas além do tempo. O amor que respalda nossas vidas está em tudo e não diferentemente, está nesse momento.

Em terceiro, quero agradecer aos meus professores. A estes, não me restrinjo há algum período. Dedico a todos... Todos vocês foram e são extremamente importantes em minha formação. Em cada ação do meu cotidiano, seja profissional ou pessoalmente, tenho em vocês um exemplo. Em especial, gostaria de agradecer a dois professores que marcaram minha jornada.

Primeiramente, a professora Sibeli Frohini, que mais que ser minha professora, me trilhou para a área da pesquisa e me incentivou a fazer o mestrado. Segundo, ao professor Jamil Viana Pereira, meu orientador nesse trabalho, a este deixo meus agradecimentos pela confiança e por ter insistido em pontos de minha formação que me tornaram o profissional que sou hoje.

Por fim, agradeço aos meus amigos. Amigos que formei durante minha vida. De fato, não são muitos, mas são importantes em minha jornada. Destes destaco, Gustavo Henrique Massarente, que se mostrou mais que um simples companheiro, mas um dos irmãos que a vida, as vezes, nos felicita.

Assim, finalizo dedicando a Deus. Ele se manifesta em todas as pessoas anteriormente citadas. Agradeço pelas bênçãos e a significância que trouxe a minha vida.

**Meu muito obrigado a todos.**

*“A matemática é o alfabeto no qual Deus escreveu o Universo.”*  
Galileu Galilei

# Resumo

O principal objetivo desse trabalho é desenvolver um algoritmo baseado no Teorema de Legendre, que conta quantos fatores primos existem na decomposição do fatorial de um número natural  $n$ , bem como, descrever uma atividade que envolvesse Teoria dos Números e programação computacional básica, voltada para estudantes da educação básica. Detalhamos a construção dos objetos do Teorema de Legendre sobre o ponto de vista das duas funções envolvidas. Em seguida, descrevemos dois algoritmos que tratam situações complementares ao Teorema de Legendre, sendo um deles apresentado no livro “Aritmética” da coleção PROFMAT HEFEZ (2016), que teve sua base teórica explicitada detalhadamente, complementando o que foi apresentado na referência e outro desenvolvido de forma independente pelos autores do trabalho, com sua respectiva base teórica construída em detalhes. Por fim, foi construída uma atividade voltada aos estudantes de educação básica que integrasse aspectos da Teoria dos Números, aprendidos durante essa fase de ensino, bem como, ideias de programação básica usando o “VisualG”.

**Palavras-chave:** Legendre, Fatores primos, Teorema Fundamental da Aritmética, Aritmética, Teorema de Legendre.

# Abstract

The goal of this work is to develop an algorithm based on Legendre's Theorem, which counts how many prime factors there are in the factorial decomposition of a natural number  $n$ , as well as to describe an activity that involves Number Theory and computer programming addressed for basic education students. We seek to understand the construction of Legendre's Theorem objects from the point of view of the two functions involved. Next, we describe two algorithms that deal with situations that are complementary to Legendre's Theorem, one of which presented in the book "Aritmética" HEFEZ (2016) of the PROFMAT collection, which had its theoretical basis explained, and a complementary one, independently developed by the authors, with the respective theoretical basis built in detail. Finally, the construction of an activity for basic education students was described, integrating aspects of Number Theory learned during this teaching phase with basic programming ideas using "VisualG".

**Keywords:** Legendre, Prime factors, Fundamental Theorem of Arithmetic, Arithmetic, Theorem's Legendre.



# Lista de Figuras

4.1	Fluxograma relacionado aos passos lógicos interpretados no algoritmo apresentado por Abramo Hefez, encontrado no Livro Aritmética da coleção PROFMAT. . . . .	31
4.2	Fluxograma relacionado aos passos lógicos interpretados no algoritmo das matrizes, desenvolvido de forma inédita pelos autores. . . . .	44
4.3	Variação $\alpha \times p \times t$ para o Algoritmo p-ádica. . . . .	47
4.4	Variação $\alpha \times p \times t$ para o Algoritmo das matrizes. . . . .	47
4.5	Sobreposição dos dados apresentados pelas Figuras 4.3 e 4.4. . . . .	47
4.6	Histograma de frequência baseado na Observação 4.50. . . . .	48
5.1	Fluxograma dos divisores . . . . .	56
5.2	Fluxograma do MDC . . . . .	58
5.3	Fluxograma do MMC . . . . .	61
5.4	Fluxograma da classificação entre primos e compostos . . . . .	64
5.5	Fluxograma do TFA . . . . .	66

# Sumário

<b>1</b>	<b>Introdução</b>	<b>10</b>
<b>2</b>	<b>Apresentação dos objetos de estudo e suas propriedades</b>	<b>11</b>
2.1	A função que conta fatores primos . . . . .	11
2.2	A função fatorial . . . . .	13
2.3	Composição da função que conta fatores primos com a fatorial . . . . .	14
<b>3</b>	<b>A equação <math>E_p(n) = \alpha</math></b>	<b>16</b>
3.1	Estudo da equação fixando $p$ e $n$ . . . . .	16
3.2	Estudo da equação fixando $p$ e $\alpha$ . . . . .	17
3.3	Estudo da equação fixando $n$ e $\alpha$ . . . . .	17
<b>4</b>	<b>A equação <math>E_p(n!) = \alpha</math></b>	<b>19</b>
4.1	Estudo da solução fixando $p$ e $n$ (Teorema de Legendre) . . . . .	19
4.2	Estudo da solução fixando $p$ e $\alpha$ . . . . .	26
4.2.1	Algoritmo p-ádica . . . . .	28
4.2.2	Algoritmo das matrizes, construído pelos autores . . . . .	34
4.2.3	Duelo computacional: p-ádica X matrizes . . . . .	46
4.3	Estudo da solução fixando $n$ e $\alpha$ . . . . .	49
<b>5</b>	<b>Uso de linguagem de programação no processo de ensino e aprendizagem</b>	<b>51</b>
5.1	O “Pensamento computacional” e o ensino da matemática . . . . .	51
5.1.1	Pensamento computacional . . . . .	51
5.1.2	Ensino da matemática: pensamento computacional . . . . .	52
5.2	Ação nacional de educação e o pensamento computacional . . . . .	53
5.3	Proposta de atividade envolvendo conceitos de teoria dos números . . . . .	54
5.3.1	<i>Software</i> . . . . .	54
5.3.2	Condições iniciais . . . . .	55
5.4	Desenvolvimento de algoritmos auxiliares . . . . .	55
5.4.1	Divisores de um $n$ natural . . . . .	55
5.4.2	Mínimo Divisor Comum de dois naturais . . . . .	58
5.4.3	Mínimo Múltiplo Comum de dois naturais . . . . .	60
5.4.4	Classificando os naturais entre primos e compostos . . . . .	63
5.4.5	Teorema Fundamental da Aritmética . . . . .	65
5.5	Atividade final . . . . .	68
5.5.1	1ª etapa . . . . .	70

---

5.5.2	2 <sup>a</sup> etapa . . . . .	70
5.5.3	3 <sup>a</sup> , 4 <sup>a</sup> e 5 <sup>a</sup> etapas . . . . .	75
5.5.4	6 <sup>a</sup> , 7 <sup>a</sup> e 8 <sup>a</sup> etapas . . . . .	75
5.5.5	9 <sup>a</sup> , 10 <sup>a</sup> e 11 <sup>a</sup> etapas . . . . .	76
5.5.6	12 <sup>a</sup> e 13 <sup>a</sup> etapas . . . . .	76
<b>A</b>	<b>Função Gama</b>	<b>79</b>
<b>B</b>	<b>Linhas de programação - Algoritmo p-ádica e algoritmo das matrizes</b>	<b>83</b>

# 1 Introdução

O presente trabalho tem seu principal objetivo relacionado ao Teorema de Legendre que conta a quantidade de fatores de um dado  $p$  primo existentes no fatorial de um  $n$  natural. Além disso, descrever uma aplicação prática, voltada ao ensino da matemática na educação básica, que aborde algumas temáticas de Teoria dos Números.

Iniciamos nossas investigações trazendo resultados sobre duas funções, uma que relaciona um  $n$  natural com quantidade de fatores de um dado  $p$  primo em sua decomposição ( $E_p(n)$ ), e outra que associa um dado natural  $n$  com seu fatorial ( $!(n)$ ).

As análises descritas acima produziram resultados que foram usados de base para o seguimento do trabalho. Como o teorema em questão tem seu resultado sobre o fatorial de um número natural, passamos a analisar as propriedades da composição entre as duas funções descritas anteriormente.

Logo após, estratificamos nossa análise em alguns casos de conveniência. Como a função composta ( $E_p \circ !(n)$ ) possui três variáveis ( $p$ ,  $n$  e  $\alpha$ ) cada caso trabalha fixando duas e tomando a outra como variável livre. Esse procedimento foi realizado tanto para a função  $E_p(n)$  como para a  $E_p \circ !(n)$ .

No momento em que abordamos o caso de  $p$  e  $\alpha$ , para a função  $E_p \circ !(n)$ , introduzimos dois algoritmos que buscam encontrar os valores de  $n$  natural que satisfazem a equação. Um dos algoritmos foi encontrado no livro “Aritmética” da coleção PROFMAT, escrito por (HEFEZ, 2016)<sup>1</sup>. O outro algoritmo foi construído de forma independente pelos autores do trabalho e traz uma roupagem diferente da encontrada nos textos do livro de Abramo Hefez.

Por fim, o trabalho segue com uma atividade prática, voltada ao ensino da matemática na educação básica. A atividade explora temáticas da Teoria dos Números aplicada ao ensino de programação básica. Nela buscamos descrever um projeto que poderia ser desenvolvido durante as aulas de matemática que visassem o ensino de conceitos como MDC, MMC, Teorema fundamental da Aritmética, conceito de divisores e classificação de naturais em compostos e primos a partir da criação de linhas de programação computacionais.

---

<sup>1</sup>Matemático egípcio, naturalizado brasileiro, que atua como professor titular no Instituto de Matemática da Universidade Federal Fluminense.

# 2 Apresentação dos objetos de estudo e suas propriedades

## 2.1 A função que conta fatores primos

Nesse capítulo, fixado um primo, investigaremos o funcionamento da função que, a cada número natural, associa a quantidade de fatores do primo fixado. Para tanto, focaremos nas propriedades e caracterização de tal função, apresentando exemplos para maior compreensão.

Para início do trabalho denota-se  $\mathbb{N}$  como o conjunto dos Naturais,  $\mathbb{N} = \{1, 2, 3, 4, \dots\}$ , além disso, classificamos um número natural como primo, quando o mesmo possui apenas dois divisores naturais distintos. A fim de caracterizarmos nossa função, teremos como base o Teorema Fundamental da Aritmética (TFA<sup>1</sup>), enunciado abaixo e seguido por algumas colocações relevantes à função que desejamos definir.

**Teorema 2.1.** (Teorema Fundamental da Aritmética - TFA<sup>1</sup>) *Todo número natural  $n \geq 2$  possui uma decomposição única por fatores primos, isto é, dado  $n \geq 2$  existem únicos primos distintos  $p_1, p_2, \dots, p_r$  e naturais  $\alpha_1, \alpha_2, \dots, \alpha_r$  tais que  $n = p_1^{\alpha_1} p_2^{\alpha_2} \dots p_r^{\alpha_r}$ .*

**Corolário 2.2.** *Fixado  $p$  primo e  $n \in \mathbb{N}$  existem únicos  $q \in \mathbb{N}$  e  $\alpha \in \mathbb{N} \cup \{0\}$  tais que  $\text{mdc}(p, q) = 1$  e  $n = p^\alpha q$ .*

*Demonstração.* Vamos iniciar com a existência, se  $n = 1$ , tome  $\alpha = 0$  e  $q = 1$ . Se  $n \geq 2$  e  $p$  não divide  $n$ , tome  $\alpha = 0$  e  $q = n$ . E, se  $n \geq 2$  e  $p$  divide  $n$ , na decomposição do TFA, existem primos  $p, p_1, p_2, \dots, p_r$  e naturais  $\alpha, \alpha_1, \alpha_2, \dots, \alpha_r$  tais que  $n = p^\alpha p_1^{\alpha_1} p_2^{\alpha_2} \dots p_r^{\alpha_r}$ . Assim, basta tomar  $q = p_1^{\alpha_1} p_2^{\alpha_2} \dots p_r^{\alpha_r}$ .

Para provar a unicidade, seja  $n = p^\alpha q$  e  $n = p^{\alpha'} q'$ , com  $\text{mdc}(p, q) = 1$  e  $\text{mdc}(p, q') = 1$ . Então, supondo  $\alpha \geq \alpha'$  temos

$$\begin{aligned} p^\alpha q &= p^{\alpha'} q' \Rightarrow \\ p^{\alpha - \alpha'} q &= q' \Rightarrow \\ p^{\alpha - \alpha'} &| q'. \end{aligned}$$

Como  $\text{mdc}(p, q') = 1$ , então  $\alpha = \alpha'$  levando  $q = q'$ , provando a unicidade da decomposição de  $n$ . □

---

<sup>1</sup>Teorema demonstrado no Apêndice A.

É a partir da colocação anterior que daremos início às tratativas da função que conta fatores primos, sendo esta bem definida como exposto no Corolário 2.2. Nela buscamos encontrar a quantidade máxima de fatores de um primo fixado, na decomposição de  $n \in \mathbb{N}$ .

**Definição 2.3.** Fixando  $p$  primo e adotando  $n \in \mathbb{N}$ , temos que a função  $E_p : \mathbb{N} \rightarrow \mathbb{N} \cup \{0\}$  é definida por  $E_p(n) = \alpha$ , se  $n = p^\alpha q$ , com  $\text{mdc}(p, q) = 1$ .

**Observação 2.4.** Tomando o Corolário 2.2 e a Definição 2.3, segue,

- i)  $E_p(n) = 0$  se e, só se,  $p \nmid n$ .
- ii) Escrever  $n = p^\alpha q$ , com  $\text{mdc}(p, q) = 1$ , é equivalente dizer que  $\alpha$  é a potência de  $p$ , tal que,  $p^\alpha$  divide  $n$ .
- iii) Tomando  $p$  e  $p'$  primos distintos, então  $E_p(p) = 1$  e  $E_p(p') = 0$ .
- iv)  $E_p$  não é injetiva. Basta tomar  $p$  primo e  $\alpha \in \mathbb{N}$  e  $q, q' \in \mathbb{N}$  primos com  $p$  e entre si, então  $m = p^\alpha q$  e  $n = p^\alpha q'$  são distintos e  $E_p(m) = E_p(n) = \alpha$ .
- v)  $E_p$  não é monótona. Basta fixar  $\alpha > 0$  e tomar  $a = p^\alpha$ ,  $b = p^\alpha + 1$ ,  $c = p^{\alpha+1} - 1$  e  $d = p^{\alpha+1}$ , então  $a < b < c < d$  e  $E_p(a) = \alpha > 0 = E_p(b)$ , enquanto  $E_p(c) = 0 < \alpha + 1 = E_p(d)$ .

Após as considerações tomadas dispomos da proposição abaixo, que expõe as propriedades dessa função.

**Proposição 2.5.** Fixado  $p$  primo, a função  $E_p : \mathbb{N} \rightarrow \mathbb{N} \cup \{0\}$ , possui as seguintes propriedades:

P1.  $E_p(mn) = E_p(m) + E_p(n)$ .

P2.  $E_p$  é sobrejetiva.

P3. Se  $E_p(m) = \alpha_1$  e  $E_p(n) = \alpha_2$ , com  $\alpha_1 \neq \alpha_2$ , então  $E_p(m+n) = \min\{\alpha_1, \alpha_2\}$ .

P4. Se  $E_p(m) = E_p(n) = \alpha$ , então  $E_p(m+n) \geq \alpha$  e  $E_p(m+n)$  pode assumir um valor  $\alpha + r$ ,  $r \geq 1$ .

*Demonstração.*

P1. Se  $E_p(m) = \alpha_1$  e  $E_p(n) = \alpha_2$ , então

$$\begin{aligned} m &= p^{\alpha_1} q, \text{ com } \text{mdc}(p, q) = 1 \\ n &= p^{\alpha_2} q', \text{ com } \text{mdc}(p, q') = 1. \end{aligned}$$

Notamos que  $\text{mdc}(p, qq') = 1$ , assim,

$$\begin{aligned} mn &= p^{\alpha_1} q p^{\alpha_2} q' \Rightarrow \\ mn &= p^{\alpha_1 + \alpha_2} q q' \Rightarrow \\ E_p(mn) &= \alpha_1 + \alpha_2 \Rightarrow \\ E_p(mn) &= E_p(m) + E_p(n). \end{aligned}$$

P2. Tomando  $\alpha = 0$  temos  $E_p(n) = 0$  para  $p \nmid n$ , e dado  $\alpha \in \mathbb{N}$ , temos  $\alpha = E_p(p^\alpha)$ .

P3. Segue da definição de  $E_p$  que,

$$\begin{aligned} m &= p^{\alpha_1}q, \text{ com } \text{mdc}(p, q) = 1 \\ n &= p^{\alpha_2}q', \text{ com } \text{mdc}(p, q') = 1. \end{aligned}$$

Então, supondo sem perda de generalidade que  $\alpha_1 < \alpha_2$ , podemos escrever que

$$\begin{aligned} m + n &= p^{\alpha_1}q + p^{\alpha_2}q' \Rightarrow \\ m + n &= p^{\alpha_1} \left( q + p^{\alpha_2 - \alpha_1}q' \right). \end{aligned}$$

Segue da igualdade anterior que  $E_p(n + m) \geq \alpha_1$ . Para ocorrer  $E_p(n + m) > \alpha_1$ , o número  $q + p^{\alpha_2 - \alpha_1}q'$  deve ser divisível por  $p$ , o que implicaria em  $q$  ser divisível por  $p$ , uma contradição. Portanto,  $E_p(n + m) = \alpha_1$ .

P4. A demonstração de que  $E_p(n + m) \geq \alpha$  é análoga a realizada no item P3. Em particular, para  $n = p^\alpha$  e  $m = p^\alpha(p^r - 1)$ ,  $r \geq 1$  tem-se

$$E_p(n + m) = E_p(p^\alpha + p^\alpha(p^r - 1)) = E_p(p^{\alpha+r}) = \alpha + r.$$

□

**Exemplo 2.6.** Tomando  $m = 7^2$  e  $n = 7^3$ , temos que  $E_7(m) = E_7(m + n) = 2$ . Agora, dado  $m = 5 \cdot 2^2$  e  $n = 11 \cdot 2^2$ , então  $E_2(m) = E_2(n) = 2$ , já  $E_2(m + n) = 6$ .

Segue dos itens acima a caracterização da função. Nela podemos definir,  $E_p : \mathbb{N} \rightarrow \mathbb{N} \cup \{0\}$  como,

$$E_p(n) = \begin{cases} 0; & \text{se } p \nmid n \\ \alpha; & \text{se } p^\alpha \mid n \text{ e } p^{\alpha+1} \nmid n, \text{ com } \alpha > 0 \end{cases}.$$

## 2.2 A função fatorial

Uma vez que a função fatorial é parte importante do conteúdo abordado nesse trabalho, apresentaremos a seguir sua definição e algumas propriedades.

**Definição 2.7.** A função  $f : \mathbb{N} \rightarrow \mathbb{N}$  é definida para  $f(1) = 1$  e para  $n \geq 2$  recursivamente por  $f(n) = n \cdot f(n - 1)$ . A notação usual para  $f(n)$  é  $n!$ .

**Observação 2.8.** Analisando a definição recursiva de  $n!$  é possível obter que  $n! = n(n - 1)(n - 2) \dots 3 \cdot 2 \cdot 1$ .

Sobre sua monotonicidade expomos o seguinte.

**Proposição 2.9.** A função fatorial é estritamente crescente.

*Demonstração.* Por definição,

$$(n+1)! = (n+1) \cdot n! \Rightarrow \frac{(n+1)!}{n!} = n+1 > 1.$$

□

Tomando o resultado acima, dispomos dois corolários.

**Corolário 2.10.** *A função fatorial é injetiva.*

**Corolário 2.11.** *Tomando a Proposição 2.9 e os casos em que  $2! = 2$  e  $3! = 6$ , temos que  $E_p$  não é sobrejetiva.*

Embora não seja necessário para a proposta desse trabalho, um fato curioso e desvelado durante a pesquisa acerca da função fatorial, é que ela pode ser estendida do conjunto números naturais para o conjunto dos reais positivos. Não detalharemos as características sobre tal função, mas deixaremos essa discussão à ser feita em no Apêndice A.

## 2.3 Composição da função que conta fatores primos com a fatorial

Passaremos a trabalhar uma composição entre a função  $E_p$  e a função fatorial. Para representar tal composição utilizaremos a notação  $E_p^!$ , observe que  $E_p^!(n) = E_p(n!)$ .

**Definição 2.12.** Fixando  $p$  primo e adotando  $n \in \mathbb{N}$ , temos que a função  $E_p^! : \mathbb{N} \rightarrow \mathbb{N} \cup \{0\}$  é definida por  $E_p^!(n!) = \alpha$ , se  $n! = p^\alpha q$ , com  $\text{mdc}(p, q) = 1$ .

Note que a definição apresentada é análogo à Definição 2.3, uma vez que  $n! \in \mathbb{N}$ . Assim, tomamos as propriedades dessa composição.

**Proposição 2.13.** *Fixado  $p$  primo e  $E_p^! : \mathbb{N} \rightarrow \mathbb{N} \cup \{0\}$ , valem as seguintes propriedades:*

$$P1. E_p^!(n!) = E_p\left(\frac{n!}{(n-m)!}\right) + E_p^!((n-m)!).$$

$$P2. E_p^! \text{ é crescente }^2.$$

$$P3. \text{ Se } p \leq n < 2p, \text{ então } E_p^!(n!) = 1.$$

$$P4. E_p^! \text{ não é injetiva.}$$

$$P5. E_p^! \text{ não é sobrejetiva.}$$

<sup>2</sup>Seja  $f$  uma função, tal que,  $f : A \rightarrow B$  e  $f(a) = b$ , onde  $a \in A$  e  $b \in B$ . Tomando  $a' \in A$

- $a > a'$  e  $f(a) > f(a')$ , então  $f$  é estritamente crescente.
- $a > a'$  e  $f(a) \geq f(a')$ , então  $f$  é crescente.
- $a > a'$  e  $f(a) < f(a')$ , então  $f$  é estritamente decrescente.
- $a > a'$  e  $f(a) \leq f(a')$ , então  $f$  é decrescente.



P6. Dado  $p$  primo fixo,  $E_p(n!) = 0$  se, e só se,  $n < p$ .

*Demonstração.*

P1. Como  $n! = \frac{n!}{(n-m)!} \cdot (n-m)!$ , aplicando o item P1 da Proposição 2.5, temos

$$E_p(n!) = E_p\left(\frac{n!}{(n-m)!}\right) + E_p((n-m)!).$$

P2. Tomando  $m > n$  e  $p$  primo, temos que a quantidade de fatores  $p$  na decomposição de  $n!$  existem em igual ou maior quantidade na decomposição de  $m!$ . Implicando que a função seja crescente.

P3. Observe que  $E_2(2!) = E_2(3!) = 1$ , logo há validade quando  $p = 2$ .

Agora, se  $p > 2$ , temos que  $p^2 \mid (2p)!$  e  $p^3 \nmid (2p)!$ , logo  $E_p((2p)!) = 2$ . Como  $E_p((2p)!) = E_p((2p-1)!) + E_p(2p)$ , temos  $E_p((2p-1)!) = 1$ . Assim,

$$\begin{aligned} p \leq n \leq 2p-1 &\Rightarrow \\ p! \leq n! \leq (2p-1)! &\Rightarrow \\ E_p(p!) \leq E_p(n!) \leq E_p((2p-1)!) &\Rightarrow \\ 1 \leq E_p(n!) \leq 1 &\Rightarrow \\ E_p(n!) = 1. & \end{aligned}$$

P4. Tomemos como contraposição o caso em que  $p = 2$ , onde  $E_2(2!) = E_2(3!) = 1$ .

P5. Suponhamos  $E_p((p^2-1)!) = \alpha$ . Pelo item P1 da Proposição 2.5 calculamos o valor de  $E_p((p^2)!)$ .

$$\begin{aligned} E_p((p^2)!) &= E_p(p^2) + E_p((p^2-1)!) \\ &= 2 + \alpha. \end{aligned}$$

Assim, dado que a função é crescente, temos que, não existe  $n$ , tal que,  $E_p(n!) = \alpha + 1$ .

P6.  $E_p(n!) = 0$  se, e só se,  $p \nmid n!$ , é equivalente a  $n < p$ .

□

### 3 A equação $E_p(n) = \alpha$

Note que a equação, possui os valores  $p$  primo,  $n \in \mathbb{N}$  e  $\alpha \in \mathbb{N} \cup \{0\}$ , então fixaremos dois desses valores e encontraremos o terceiro em função dos fixados. Buscando uma organização, dividimos nos diferentes casos:

Tabela 3.1: Localização e assunto-chave das subseções da Seção 3.

Caso	Localização
$p$ e $n$ previamente fixados	<a href="#">Subseção 3.1</a>
$p$ e $\alpha$ previamente fixados	<a href="#">Subseção 3.2</a>
$n$ e $\alpha$ previamente fixados	<a href="#">Subseção 3.3</a>

Fonte: Autoria nossa.

#### 3.1 Estudo da equação fixando $p$ e $n$

Fixados  $n \in \mathbb{N}$  e  $p$  primo, temos  $\alpha$ , como variável livre e desejamos determinar possíveis expressões de  $\alpha$  que resolvam a equação  $E_p(n) = \alpha$ . Uma condição interessante e de fácil observação está no caso em que  $n$  é primo. Se  $n$  é distinto de  $p$ , então  $\alpha = 0$  e se  $n$  é igual a  $p$ , então  $\alpha = 1$ .

Visando uma melhor organização subdividiremos nossas abordagens como segue abaixo:

Tabela 3.2: Indicativo e localização de cada caso abordado na subseção 3.1.

Fixando	Localização
$n = 1$ e $p$ primo	<a href="#">Proposição 3.2</a>
$n \geq 2$ e $p$ primo	<a href="#">Proposição 3.3</a> <a href="#">Proposição 3.4</a>

Fonte: Autoria nossa.

**Observação 3.1.** Temos que o TFA garante condições para discutirmos casos em que  $n \geq 2$ , o caso  $n = 1$  deve ser estudado separadamente.

**Proposição 3.2.** Fixado  $n = 1$  e  $p$  primo, temos  $E_p(n) = 0$ .

*Demonstração.* Dado  $n = 1$  e  $p$  primo, então  $n = p^0 \cdot 1$ , com  $\text{mdc}(n, 1) = 1$ . Logo,  $E_p(n) = 0$ . □

**Proposição 3.3.** Fixados  $n \in \mathbb{N}$ ,  $n \geq 2$  e  $p$  primo,  $\text{mdc}(n, p) = 1$  se, e somente se,  $E_p(n) = 0$ .

*Demonstração.* Se  $\text{mdc}(n, p) = 1$  então  $p$  não divide  $n$ , logo  $E_p(n) = 0$ . Reciprocamente, se  $E_p(n) = 0$ , então  $n = p^0 \cdot 1$ , com  $\text{mdc}(n, p) = 1$ . □

**Proposição 3.4.** Fixados  $n \in \mathbb{N}$ ,  $n \geq 2$  e  $p$  primo, então existe única solução  $\alpha \in \mathbb{N} \cup \{0\}$  para a equação  $E_p(n) = \alpha$ .

*Demonstração.* Consequência direta da Definição 2.3 e do Corolário 2.2. □

### 3.2 Estudo da equação fixando $p$ e $\alpha$

Fixados  $\alpha \in \mathbb{N} \cup \{0\}$  e  $p$  primo, temos  $n$ , como variável livre e desejamos determinar possíveis expressões de  $n$  que resolvam a equação  $E_p(n) = \alpha$ .

**Proposição 3.5.** Fixados  $\alpha \in \mathbb{N} \cup \{0\}$  e  $p$  primo, existem infinitas soluções do tipo  $n = p^\alpha q$ , com  $\text{mdc}(p, q) = 1$  para a equação  $E_p(n) = \alpha$ .

*Demonstração.*  $E_p(n) = \alpha$ , se  $n = p^\alpha q$ , sendo  $\text{mdc}(p, q) = 1$ . Fixado  $p$  e  $\alpha \in \mathbb{N} \cup \{0\}$ , existem infinitos  $q$  que satisfazem tal condição, portanto existem infinitas soluções do tipo  $n = p^\alpha q$ , com  $\text{mdc}(p, q) = 1$ . □

### 3.3 Estudo da equação fixando $n$ e $\alpha$

Fixados  $n \in \mathbb{N}$  e  $\alpha \in \mathbb{N} \cup \{0\}$ , temos  $p$  primo, como variável livre e desejamos determinar possíveis expressões de  $p$  que resolvam a equação  $E_p(n) = \alpha$ .

Visando uma melhor organização subdividiremos nossas abordagens como seguem abaixo:

Tabela 3.3: Indicativo e localização de cada caso abordado na subseção 3.3.

Fixando	Localização
$n = 1$ e $\alpha = 0$	<a href="#">Proposição 3.6</a>
$n = 1$ e $\alpha > 0$	<a href="#">Proposição 3.7</a>
$n \geq 2$ e $\alpha = 0$	<a href="#">Proposição 3.8</a>
$n$ primo e $\alpha > 0$	<a href="#">Proposição 3.9</a>
$n$ composto e $\alpha > 0$	<a href="#">Proposição 3.10</a>

Fonte: Autoria nossa.

**Proposição 3.6.** Fixando  $n = 1$  e  $\alpha = 0$ , então a equação  $E_p(n) = \alpha$  possui infinitas soluções.

*Demonstração.* Para todo  $p$  primo, temos  $n = p^0 \cdot 1$ , implicando em  $E_p(1) = 0$ . □

**Proposição 3.7.** Fixando  $n = 1$  e  $\alpha > 0$ , então a equação  $E_p(n) = \alpha$  não possui solução.

*Demonstração.* Sendo  $\alpha > 0$  e  $p$  primo, temos  $p^\alpha q > 1$ , para todo  $q \in \mathbb{N}$ . □

Agora tomando  $n \geq 2$  atrelamos nossas colocações às garantias existente pelo TFA. Disso, propomos os resultados que seguem.

**Proposição 3.8.** *Fixados  $n \geq 2$  e  $\alpha = 0$ , a equação  $E_p(n) = \alpha$  possui infinitas soluções.*

*Demonstração.* Se  $p$  é um primo que não aparece na decomposição de  $n$ , então  $n = p^0 \cdot q$ , com  $\text{mdc}(p, q) = 1$ . Logo,  $E_p(n) = 0$ . Como existem infinitos primos nessas condição, então  $E_p(n) = 0$  tem infinitas soluções.  $\square$

**Proposição 3.9.** *Fixando  $n$  primo e:*

- i.  $\alpha = 1$ , então  $p = n$  é solução única da equação  $E_p(n) = \alpha$ .*
- ii.  $\alpha > 1$ , então não existe  $p$  primo que satisfaça a equação  $E_p(n) = \alpha$ .*

*Demonstração.* Sejam  $n$  e  $p$  primos e  $\alpha \in \mathbb{N}$ ,  $p^\alpha \mid n$  se, e só se,  $n = p$  e  $\alpha = 1$ . Por consequência, se  $\alpha > 1$ , então não existe  $p$  primo, tal que,  $p^\alpha \mid n$ .  $\square$

**Proposição 3.10.** *Fixados  $n$  composto, tal que,  $n = p_1^{\alpha_1} p_2^{\alpha_2} \dots p_u^{\alpha_u}$  (decomposição do TFA), e  $\alpha \in \mathbb{N}$ , temos:*

- i. caso  $\alpha = \alpha_s$ , para algum  $s \in \{1, 2, \dots, u\}$ , então  $p = p_s$  é solução de  $E_p(n) = \alpha$ , observando que pode haver mais que uma solução.*
- ii. caso  $\alpha \neq \alpha_s$ , para todo  $s \in \{1, 2, \dots, u\}$ , então não existe primo  $p$  solução de  $E_p(n) = \alpha$ .*

*Demonstração.* Adotando  $n = p_1^{\alpha_1} \dots p_u^{\alpha_u}$  com  $s \in \{1, 2, \dots, u\}$ , temos que os únicos candidatos a solução são  $p_1, p_2, \dots, p_u$ , então  $p_s$  é solução se, e só se,  $\alpha = \alpha_s$ .  $\square$

## 4 A equação $E_p(n!) = \alpha$

Nesse momento, investigaremos as soluções da equação quando tomamos a forma fatorial de  $n \in \mathbb{N}$ . Para tanto, prossigamos como na seção anterior, quando fixamos algumas variáveis e outras não.

Tabela 4.1: Localização e assunto-chave das subseções da Seção 4.

Caso	Localização
$p$ e $n$ previamente fixados	<a href="#">Subseção 4.1</a>
$p$ e $\alpha$ previamente fixados	<a href="#">Subseção 4.2</a>
$n$ e $\alpha$ previamente fixados	<a href="#">Subseção 4.3</a>

Fonte: Autoria nossa.

### 4.1 Estudo da solução fixando $p$ e $n$ (Teorema de Legendre)

Para uma melhor organização tome a tabela abaixo que destaca os resultados principais abordados nesse momento.

Tabela 4.2: Resultados principais da sub-seção 4.1.

Estudo	Resultado
Existência e unicidade	<a href="#">Proposição 4.1</a>
Caracterização	<a href="#">Teorema 4.5</a> <a href="#">Teorema 4.10</a>

Fonte: Autoria nossa.

Tomemos  $E_p(n!) = \alpha$ , com  $n \in \mathbb{N}$  e  $p$  primo fixos. Portanto, temos  $\alpha$  como variável livre e desejamos determinar possíveis expressões de  $\alpha \in \mathbb{N} \cup \{0\}$  que resolvam a equação.

Fica evidente que  $n!$  pode e deve ser entendido como um valor natural sobre o ponto de vista da função. Isso concerne para o entendimento que as propriedades no estudo da equação  $E_p(n) = \alpha$  sejam parcialmente conservadas. Inicialmente, tomemos a condição que segue.

**Proposição 4.1.** *Fixos  $p$  primo e  $n \in \mathbb{N}$ , sempre existirá uma única solução na equação  $E_p(n!) = \alpha$ .*

*Demonstração.*

*Existência:*

- i.  $n < p$ , então  $p \nmid n!$ , logo  $E_p(n!) = 0$ .
- ii.  $n = p$ , então  $p \mid n!$  e  $p^2 \nmid n!$ , logo  $E_p(n!) = 1$ .
- iii.  $n > p$ , então existe  $\alpha \in \mathbb{N}$ , tal que,  $p^\alpha \mid n$  e  $p^{\alpha+1} \nmid n$ , logo  $E_p(n!) = \alpha$ .

*Unicidade:*

Diretamente da decomposição de  $n!$  dada pelo TFA.

□

Além da existência e unicidade, um dos resultados de Legendre fornece uma caracterização da solução. Vamos fixar alguns conceitos antes de chegarmos a ele.

**Definição 4.2.** Sejam  $a \in \mathbb{N} \cup \{0\}$  e  $b \in \mathbb{N}$  temos que  $\left[ \frac{a}{b} \right]$ , representa o quociente inteiro da divisão de  $a$  por  $b$ . Matematicamente,

$$a = b \cdot \left[ \frac{a}{b} \right] + r, \text{ com } 0 \leq r < b.$$

**Lema 4.3.** Sejam  $p$  primo,  $s \in \mathbb{N}$  e  $k \in \mathbb{N} \cup \{0\}$ , se  $p^s \nmid (k+1)$ , então  $\left[ \frac{k}{p^s} \right] = \left[ \frac{k+1}{p^s} \right]$ .

*Demonstração.* Se  $p^s \nmid (k+1)$ , então  $(k+1) = p^s \left[ \frac{k+1}{p^s} \right] + r$ , com  $0 < r < p^s$ .

Manipulando algebricamente, temos que:

$$\begin{aligned} (k+1) &= p^s \left[ \frac{k+1}{p^s} \right] + r \Rightarrow \\ k &= p^s \left[ \frac{k+1}{p^s} \right] + r - 1. \end{aligned}$$

Como  $0 < r < p^s$ , então  $0 \leq r - 1 < p^s - 1$ . Assim, concluindo que,

$$\left[ \frac{k}{p^s} \right] = \left[ \frac{k+1}{p^s} \right].$$

□

O Lema abaixo, pode ser entendido como um complemento do Lema 4.3.

**Lema 4.4.** Sejam  $p$  primo,  $s \in \mathbb{N}$  e  $k \in \mathbb{N} \cup \{0\}$ , se  $p^s \mid (k+1)$ , então  $\left[ \frac{k}{p^s} \right] = \left[ \frac{k+1}{p^s} \right] - 1$ .

*Demonstração.* Se  $p^s \mid (k+1)$ , então  $p^s \nmid k$ . Usando isso, temos

$$\begin{aligned} k &= p^s \left[ \frac{k}{p^s} \right] + r, \text{ com } 0 < r < p^s \Rightarrow \\ k+1 &= p^s \left[ \frac{k}{p^s} \right] + (r+1), \text{ com } 1 < r+1 \leq p^s. \end{aligned}$$

Como  $p^s \mid (k+1)$  implica que  $r+1 = p^s$ . Então,

$$\begin{aligned} k+1 &= p^s \left[ \frac{k}{p^s} \right] + p^s \Rightarrow \\ k+1 &= p^s \left( \left[ \frac{k}{p^s} \right] + 1 \right) \Rightarrow \\ \left[ \frac{k}{p^s} \right] + 1 &= \left[ \frac{k+1}{p^s} \right] \Rightarrow \\ \left[ \frac{k}{p^s} \right] &= \left[ \frac{k+1}{p^s} \right] - 1. \end{aligned}$$

□

**Teorema 4.5.** (Teorema de Legendre) Sendo  $n, u \in \mathbb{N}$  e  $p$  primo, tal que,  $p^u \leq n < p^{u+1}$ , então:

$$E_p(n!) = \left[ \frac{n}{p} \right] + \left[ \frac{n}{p^2} \right] + \dots + \left[ \frac{n}{p^u} \right].$$

*Demonstração.* Fixando  $p$  primo, procedemos por indução sobre  $n$ .

Para  $n = 1$  temos  $p^0 \leq 1 < p$  e  $E_p(1!) = 0 = \left[ \frac{1}{p} \right]$ .

Para  $k \in \mathbb{N} \cup \{0\}$ , seja  $u$ , tal que,  $p^u \leq k < p^{u+1}$  e suponhamos como hipótese de indução que

$$E_p(k!) = \left[ \frac{k}{p} \right] + \dots + \left[ \frac{k}{p^u} \right].$$

Avaliemos  $E_p((k+1)!)$ . Uma vez que  $p^u \leq k < p^{u+1}$ , temos que  $k+1 \leq p^{u+1}$ . Dividiremos a demonstração em duas partes, que depende da divisibilidade pelas potências de  $p^s$ ,  $s \in \{1, 2, \dots, u+1\}$ .

i) Se  $p^s \nmid k+1$ , para todo  $s = 1, 2, \dots, u, u+1$ , então pelo Lema 4.3,  $\left[ \frac{k}{p^s} \right] = \left[ \frac{k+1}{p^s} \right]$ .

Além disso,  $E_p(k+1) = 0$ , pois  $p \nmid k+1$ .

Temos então,

$$\begin{aligned} E_p((k+1)!) &= E_p(k+1) + E_p(k!) \\ &= 0 + \left[ \frac{k}{p} \right] + \dots + \left[ \frac{k}{p^u} \right] \\ &= \left[ \frac{k+1}{p} \right] + \dots + \left[ \frac{k+1}{p^u} \right], \end{aligned}$$

pois  $\left[ \frac{k+1}{p^u} \right] = \left[ \frac{k}{p^u} \right] \geq 1$  e  $\left[ \frac{k+1}{p^{u+1}} \right] = \left[ \frac{k}{p^{u+1}} \right] = 0$ .

Portanto, nesse caso o resultado é válido para todo  $k+1$ .

Se não acontece i), seja  $j \in \{1, 2, \dots, u, u+1\}$ , o maior índice, tal que,  $p^j \mid k+1$ .

ii)-A Supondo  $j < u + 1$ , tomemos  $i \in \{1, 2, \dots, j\}$  e  $i' \in \{j + 1, j + 2, \dots, u + 1\}$ .

Note que,  $p^{i'} \nmid k + 1$ . Assim, pelo Lema 4.3,

$$\left[ \frac{k}{p^{i'}} \right] = \left[ \frac{k + 1}{p^{i'}} \right].$$

e, pelo Lema 4.4,

$$\left[ \frac{k}{p^i} \right] = \left[ \frac{k + 1}{p^i} \right] - 1$$

Além disso,  $E_p(k + 1) = j$ , pois  $p^j$  é a maior potência, tal que,  $p^j \mid k + 1$ .

Portanto,

$$\begin{aligned} E_p((k + 1)!) &= E_p(k + 1) + E_p(k!) \\ &= j + \left[ \frac{k}{p} \right] + \dots + \left[ \frac{k}{p^j} \right] + \left[ \frac{k}{p^{j+1}} \right] + \dots + \left[ \frac{k}{p^u} \right] \\ &= j + \left[ \frac{k + 1}{p} \right] - 1 + \dots + \left[ \frac{k + 1}{p^j} \right] - 1 + \left[ \frac{k + 1}{p^{j+1}} \right] + \dots + \left[ \frac{k + 1}{p^u} \right] \\ &= j - j + \left[ \frac{k + 1}{p} \right] + \dots + \left[ \frac{k + 1}{p^u} \right] \\ &= \left[ \frac{k + 1}{p} \right] + \dots + \left[ \frac{k + 1}{p^u} \right]. \end{aligned}$$

Observe que

$$\left[ \frac{k + 1}{p^u} \right] = \left[ \frac{k}{p^u} \right] \geq 1 \text{ e que } \left[ \frac{k + 1}{p^{u+1}} \right] = \left[ \frac{k}{p^{u+1}} \right] = 0.$$

ii)-B Agora, se  $j = u + 1$  e  $i \in \{1, 2, \dots, u, u + 1\}$ , temos pelo Lema 4.4 que,

$$\left[ \frac{k}{p^i} \right] = \left[ \frac{k + 1}{p^i} \right] - 1.$$

Além do que  $E_p(k + 1) = u + 1$ . Assim,

$$\begin{aligned} E_p((k + 1)!) &= E_p(k + 1) + E_p(k!) \\ &= (u + 1) + \left[ \frac{k}{p} \right] + \left[ \frac{k}{p^2} \right] + \dots + \left[ \frac{k}{p^u} \right] + \left[ \frac{k}{p^{u+1}} \right] \\ &= (u + 1) + \left[ \frac{k + 1}{p} \right] - 1 + \left[ \frac{k + 1}{p^2} \right] - 1 + \dots + \left[ \frac{k + 1}{p^u} \right] - 1 + \left[ \frac{k + 1}{p^{u+1}} \right] - 1 \\ &= (u + 1) + \left[ \frac{k + 1}{p} \right] + \left[ \frac{k + 1}{p^2} \right] + \dots + \left[ \frac{k + 1}{p^u} \right] + \left[ \frac{k + 1}{p^{u+1}} \right] - (u + 1) \\ &= \left[ \frac{k + 1}{p} \right] + \left[ \frac{k + 1}{p^2} \right] + \dots + \left[ \frac{k + 1}{p^u} \right] + \left[ \frac{k + 1}{p^{u+1}} \right] \\ &= \left[ \frac{k + 1}{p} \right] + \dots + \left[ \frac{k + 1}{p^{u+1}} \right]. \end{aligned}$$



Sendo,  $\left[ \frac{k}{p^{u+1}} \right] = \left[ \frac{k+1}{p^{u+1}} \right] - 1$ , implicando que  $\left[ \frac{k+1}{p^{u+1}} \right] = 1$  e  $\left[ \frac{k+1}{p^{u+2}} \right] = 0$ .

Portanto, nesse caso o resultado é valido para  $k+1$ .

□

A seguir exploraremos uma outra forma apresentar o Teorema de Legendre. Nessa nova abordagem tomaremos a decomposição  $p$ -ádica de  $\alpha$ , bem como, alguns resultados preliminares para melhor embasamento.

**Proposição 4.6.** *Seja  $n = n_u p^u + n_{u-1} p^{u-1} + \dots + n_1 p + n_0$ , com  $0 \leq n_s < p$ ,  $s = 0, 1, 2, \dots, u$ .*

- i) Se  $n_l = p - 1$ , com  $l = 0, 1, 2, \dots, v$ ,  $v < u$  e  $n_{v+1} < p - 1$ , então  $n + 1 = n_u p^u + n_{u-1} p^{u-1} + \dots + p^{v+1} (n_{v+1} + 1)$ .*
- ii) Se  $n_s = p - 1$ , para todo  $s = 0, 1, 2, \dots, u$ , então  $n + 1 = p^{u+1}$ .*
- iii) Se  $n_0 < p - 1$ , então  $n + 1 = n_u p^u + n_{u-1} p^{u-1} + \dots + n_1 p + n'_0$ , com  $n'_0 = n_0 + 1$ .*
- iv) Se  $n_0 > 0$ , então  $n - 1 = n_u p^u + n_{u-1} p^{u-1} + \dots + n_1 p + n'_0$ , sendo  $n'_0 = n_0 - 1$ .*
- v) Se  $n_l = 0$ , com  $l = 0, 1, 2, \dots, v$ ,  $v < u$  e  $n_{v+1} > 0$ , então  $n - 1 = n_u p^u + n_{u-1} p^{u-1} + \dots + p^{v+1} (n_{v+1} - 1) + (p - 1) (p^v + \dots + p + 1)$ .*

*Demonstração.*

i) Tomando as hipóteses da Proposição, temos que:

$$\begin{aligned}
 n + 1 &= n_u p^u + n_{u-1} p^{u-1} + \dots + n_{v+1} p^{v+1} + (p - 1) p^{v-1} + \dots + (p - 1) p + (p - 1) + 1 \\
 &= n_u p^u + n_{u-1} p^{u-1} + \dots + n_{v+1} p^{v+1} + (p - 1) (p^v + \dots + p + 1) + 1 \\
 &= n_u p^u + n_{u-1} p^{u-1} + \dots + n_{v+1} p^{v+1} + (p - 1) \left( \frac{p^{v+1} - 1}{p - 1} \right) + 1 \\
 &= n_u p^u + n_{u-1} p^{u-1} + \dots + p^{v+1} (n_{v+1} + 1).
 \end{aligned}$$

ii) Tomando as hipóteses temos que,

$$\begin{aligned}
 n + 1 &= (p - 1) p^u + (p - 1) p^{u-1} + \dots + (p - 1) p + (p - 1) + 1 \\
 &= (p - 1) (p^u + p^{u-1} + \dots + p + 1) + 1 \\
 &= (p - 1) \left( \frac{p^{u+1} - 1}{p - 1} \right) + 1 \\
 &= p^{u+1} - 1 + 1 \\
 &= p^{u+1}.
 \end{aligned}$$

iii) Segue de forma trivial.

iv) Segue de forma trivial.

v) Tomando a hipótese estabelecida:

$$\begin{aligned}
 n - 1 &= n_u p^u + n_{u-1} p^{u-1} + \dots + n_{v+1} p^{v+1} - 1 \\
 &= n_u p^u + n_{u-1} p^{u-1} + \dots + p^{v+1} (n_{v+1} - 1 + 1) - 1 \\
 &= n_u p^u + n_{u-1} p^{u-1} + \dots + p^{v+1} (n_{v+1} - 1) + p^{v+1} - 1 \\
 &= n_u p^u + n_{u-1} p^{u-1} + \dots + p^{v+1} (n_{v+1} - 1) + (p - 1) (p^v + \dots + p + 1).
 \end{aligned}$$

□

**Proposição 4.7.** *Seja  $n \in \mathbb{N}$  e  $p$  um primo fixado, a decomposição de um  $n$  na forma  $n = n_u p^u + n_{u-1} p^{u-1} + \dots + n_1 p + n_0$ , com  $0 \leq n_s < p$ , sendo  $s = 0, 1, 2, \dots, u$ , existe e é única.*

*Demonstração.*

*Existência:*

Provemos por indução. Fixado um  $p$  primo qualquer, quando  $n = 1$ , temos  $n_0 = 1$  e  $n_t = 0$  para  $t \geq 1$ .

Seja  $s = 1, 2, \dots, u$ ,  $n \in \mathbb{N}$  e  $p$  primo, tomemos como hipótese de indução que  $n = n_u p^u + n_{u-1} p^{u-1} + \dots + n_1 p + n_0$ , sendo  $0 \leq n_s < p$ ,  $s = 0, 1, 2, \dots, u$ .

Pela Proposição 4.6, temos garantido a caracterização do sucessor de  $n$ , o que prova sua existência.

*Unicidade:*

Provemos por indução.

Se  $n = 1$ , temos  $n_0 = 1$  e  $n_t = 0$ ,  $t \geq 1$ .

Suponhamos que exista alguma outra decomposição com algum  $n_t \neq 0$ . Daí  $n_t p^t > 1$  o que contradiz  $\sum_{r=0}^u n_r p^r = 1$ .

Sendo  $k \in \mathbb{N}$  suponhamos a unicidade para todo  $n \leq k$ , assim  $k = n_u p^u + n_{u-1} p^{u-1} + \dots + n_1 p + n_0$  de forma única, bem como os valores menores que  $k$ .

Avaliemos  $k + 1$ .

Suponhamos,

$$\begin{aligned}
 k + 1 &= r_l p^l + r_{l-1} p^{l-1} + \dots + r_1 p + r_0 \\
 k + 1 &= s_m p^m + s_{m-1} p^{m-1} + \dots + s_1 p + s_0.
 \end{aligned}$$

Daí, se  $r_0 > 0$  e  $s_0 > 0$  temos pela Proposição 4.6 item iv) que

$$\begin{aligned}
 k &= r_l p^l + r_{l-1} p^{l-1} + \dots + r_1 p + r_0 - 1 \\
 k &= s_m p^m + s_{m-1} p^{m-1} + \dots + s_1 p + s_0 - 1.
 \end{aligned}$$

Como, por hipótese, a representação de  $k$  é única.

$$\begin{aligned} r_0 - 1 &= s_0 - 1 \Rightarrow r_0 = s_0. \\ r_1 p &= s_1 p \Rightarrow r_1 = s_1. \\ &\vdots \\ r_l p^l &= s_m p^m \Rightarrow r_l = s_m \Rightarrow l = m. \end{aligned}$$

Se  $r_0 > 0$  e  $s_0 = 0$ , temos pela Proposição 4.6 no item iv) que  $k = r_l p^l + r_{l-1} p^{l-1} + \dots + r_1 p + r_0 - 1$  e pelo item v) que o termo independente de  $k$  é  $p - 1$ . Daí, segue que  $r_0 - 1 = p - 1$  o que é uma contradição, pois  $0 < r_0 < p$ .

Caso em que  $r_0 = 0$  e  $s_0 > 0$  é análogo.

Por fim, se  $r_l = 0$ , com  $l = 0, 1, 2, \dots, v$  e  $s_t = 0$ , com  $t = 0, 1, 2, \dots, v'$ , temos pela Proposição 4.6 item v), que

$$\begin{aligned} k &= r_l p^l + r_{l-1} p^{l-1} + \dots + p^{v+1} (r_{v+1} - 1) + (p - 1) (p^v + \dots + p + 1) \\ k &= s_m p^m + s_{m-1} p^{m-1} + \dots + p^{v'+1} (s_{v'+1} - 1) + (p - 1) (p^{v'} + \dots + p + 1) \end{aligned}$$

Como  $k$  tem decomposição única segue que  $v = v'$ , portanto,

$$\begin{aligned} p^{v+1} (r_{v+1} - 1) &= p^{s+1} (s_{v'+1} - 1) \Rightarrow r_{v+1} = s_{v'+1}. \\ r_{v+2} p^{v+2} &= s_{v+2} p^{v+2} \Rightarrow r_{v+2} = s_{v+2}. \\ &\vdots \\ r_l p^l &= s_m p^m \Rightarrow r_l = s_m. \end{aligned}$$

□

**Definição 4.8.** Seja  $n \in \mathbb{N}$ ,  $p$  primo, temos  $n = n_u p^u + n_{u-1} p^{u-1} + \dots + n_1 p + n_0$ , com  $s = 1, 2, \dots, u$  e  $0 \leq n_s < p$ , como a forma  $p$ -ádica de  $n$ .

**Proposição 4.9.** Se  $n = n_u p^u + n_{u-1} p^{u-1} + \dots + n_1 p + n_0$ , com  $p$  primo e  $0 \leq n_s < p$  sendo  $s = 1, 2, \dots, u$ , então  $n \geq (n_u + n_{u-1} + \dots + n_1 + n_0)$ .

*Demonstração.* Provemos por indução. Tendo  $n = 1$ , então  $n_0 = 1$  ficando provado.

Supondo com hipótese de indução que para algum  $k \in \mathbb{N}$  temos  $k = k_u p^u + k_{u-1} p^{u-1} + \dots + k_1 p + k_0$ , com  $0 \leq k_s < p$ ,  $s = 0, 1, 2, \dots, u$  e  $k \geq k_u + k_{u-1} + \dots + k_1 + k_0$ . Avaliando para o caso  $k + 1$  temos da Proposição 4.6 que

- i)  $k + 1 = k_u p^u + k_{u-1} p^{u-1} + \dots + k_1 p + k_0 + 1$ , quando  $k_0 < p - 1$ .
- ii)  $k + 1 = k_u p^u + k_{u-1} p^{u-1} + \dots + p^{v+1} (k_{v+1} + 1)$ , quando  $k_0 = k_1 = \dots = k_{v-1} = k_v = p - 1$  e  $k_{v+1} < p - 1$ , para algum  $v < u$ .
- iii)  $k + 1 = p^{u+1}$ , quando  $k_0 = k_1 = \dots = k_{u-1} = k_u = p - 1$ .

No caso i), temos da hipótese de indução que  $k \geq k_u + k_{u-1} + \dots + k_1 + k_0$ , implicando que  $k + 1 \geq k_u + k_{u-1} + \dots + k_1 + k_0 + 1$ .

Caso aconteça ii), temos da hipótese de indução

$$\begin{aligned} k &\geq k_u + k_{u-1} + \dots + k_{v+1} + (v+1)(p-1) \Rightarrow \\ k+1 &\geq k_u + k_{u-1} + \dots + k_{v+1} + (v+1)(p-1) + 1 \geq k_u + k_{u-1} + \dots + k_{v+1} + 1. \end{aligned}$$

Por fim, o caso iii) é trivial pois  $k+1 > 1$ .

□

**Teorema 4.10.** *Seja  $n \in \mathbb{N}$ ,  $p$  primo,  $n = n_u p^u + n_{u-1} p^{u-1} + \dots + n_1 p + n_0$ , com  $0 \leq n_s < p$ , para  $s = 1, 2, \dots, u$ . Então,*

$$E_p(n!) = \frac{n - (n_0 + n_1 + \dots + n_{u-1} + n_u)}{p-1}.$$

*Demonstração.* Segue das hipóteses que

$$\begin{aligned} \left[ \frac{n}{p} \right] &= n_u p^{u-1} + n_{u-1} p^{u-2} + \dots + n_1 \\ \left[ \frac{n}{p^2} \right] &= n_u p^{u-2} + n_{u-1} p^{u-3} + \dots + n_2 \\ &\vdots \\ \left[ \frac{n}{p^u} \right] &= n_u. \end{aligned}$$

Tomando o Teorema de Legendre (4.5), temos:

$$\begin{aligned} E_p(n!) &= \left[ \frac{n}{p} \right] + \left[ \frac{n}{p^2} \right] + \dots + \left[ \frac{n}{p^u} \right] \\ &= n_u \cdot p^{u-1} + n_{u-1} \cdot p^{u-2} + \dots + n_1 + n_u \cdot p^{u-2} + n_{u-1} \cdot p^{u-3} + \dots + n_2 + \dots + n_u \\ &= n_u (p^{u-1} + p^{u-2} + \dots + p + 1) + n_{u-1} (p^{u-2} + p^{u-3} + \dots + p + 1) + \dots + n_1 \\ &= n_u \cdot \frac{p^u - 1}{p-1} + n_{u-1} \cdot \frac{p^{u-1} - 1}{p-1} + \dots + n_1 \cdot \frac{p-1}{p-1} \\ &= \frac{n_u \cdot p^u + n_{u-1} \cdot p^{u-1} + \dots + n_1 \cdot p + n_0 - (n_u + n_{u-1} + \dots + n_1 + n_0)}{p-1} \\ &= \frac{n - (n_0 + n_1 + \dots + n_{u-1} + n_u)}{p-1}. \end{aligned}$$

Observe que a Proposição 4.9 garante que o numerador não é negativo.

□

## 4.2 Estudo da solução fixando $p$ e $\alpha$

Para uma melhor organização tome a tabela que segue. Nela estão destacados os principais resultados abordados nesse momento.

Tabela 4.3: Resultados principais da sub-seção 4.2.

Estudo	Resultado
Existência	<a href="#">Proposição 4.11</a>
Caracterização	<a href="#">Corolário 4.12</a>

Fonte: Autoria nossa.

Tomemos  $E_p(n!) = \alpha$ , com  $p$  e  $\alpha$  previamente fixados. Portanto, temos  $n$ , como variável livre e desejamos determinar possíveis expressões de  $n$  que resolvam a equação.

**Proposição 4.11.** *Fixados  $\alpha \in \mathbb{N} \cup \{0\}$  e  $p$  primo,*

- i) se  $\alpha = 0$ , então a equação possui exatamente  $p - 1$  soluções distintas.*
- ii) se  $\alpha > 0$  e existir  $n' \in \mathbb{N}$ , tal que,  $E_p(n') = \alpha$ , então a equação possui exatamente  $p$  soluções naturais distintas, sendo a menor delas múltipla de  $p$ .*

*Demonstração.*

- i) Se  $\alpha = 0$ , então  $E_p(n!) = \alpha$  para  $n = 1, 2, \dots, p - 1$ . Além disso, se  $n \geq p$ , então  $E_p(n!) > \alpha$ . Logo, existem  $p - 1$  soluções distintas.
- ii) Como  $E_p(n!) = \alpha$  possui solução natural temos pelo Princípio da Boa Ordenação (PBO) que o conjunto de soluções da equação possui menor elemento  $n_0$ . Assim, como  $E_p(n_0!) = \alpha$ , temos que  $p \mid n_0$ . De fato, se  $p \nmid n_0$ , temos  $E_p((n_0 - 1)!) = E_p(n_0!)$ , o que contradiz  $n_0$  ser o menor elemento. Temos então que  $p \mid n_0$ .

Daí, para  $1 \leq m \leq p - 1$ , temos que  $p \nmid n_0 + m$ . Portanto, a quantidade de fatores  $p$  em  $(n_0 + m)!$  é igual à quantidade em  $n_0!$ , isto é,

$$E_p(n_0!) = E_p((n_0 + 1)!) = \dots = E_p((n_0 + p - 1)!) = \alpha.$$

Observe que se  $n \geq n_0 + p$ ,  $E_p(n!) > \alpha$ . Portanto,  $n_0, n_0 + 1, \dots, n_0 + p - 1$  são as únicas soluções da equação.

□

**Corolário 4.12.** *Seja  $\alpha \in \mathbb{N}$ ,  $p$  primo fixos, se existir solução para  $E_p(n!) = \alpha$ , então escrevendo  $n = mp + r$ , com  $0 \leq r < p$ , todas as soluções da equação são dadas por  $mp, mp + 1, \dots, mp + p - 1$ , isto é*

$$E_p((mp)!) = E_p((mp + 1)!) = \dots = E_p((mp + p - 1)!) = \alpha.$$

*Demonstração.* Basta observar que a demonstração do item ii) da Proposição 4.11, onde menor solução é um múltiplo de  $p$ . □

Prosseguindo, abordaremos dois algoritmos que descrevem o processo para se obter o menor  $n$  natural que satisfaz a equação.

### 4.2.1 Algoritmo $p$ -ádica

Baseado no livro de Aritmética da coleção do PROFMAT, escrito por (HEFEZ, 2016)<sup>1</sup>, esse algoritmo descreve uma forma para aferirmos a existência ou não de solução para equação, bem como, caso exista, quem são os naturais que á satisfazem. Além do mais, essa seção apresenta detalhamentos ausentes no livro referenciado, deixando as condições teóricas do algoritmo evidentes. Assim, trazendo maior solidez para as etapas em que o algoritmo é desenvolvido.

Para iniciarmos, seja  $m \in \mathbb{N}$ , tal que,

$$m = m_u p^u + m_{u-1} p^{u-1} + \dots + m_1 p + m_0,$$

sendo  $0 \leq m_s < p$ , para  $s = 1, 2, \dots, u$ , com  $m_u \neq 0$ , buscando soluções do tipo  $mp$

**Lema 4.13.** *Sejam  $\alpha \in \mathbb{N}$  e  $p$  primo, fixos. Tomando  $m \in \mathbb{N}$ , onde  $m = m_u p^u + m_{u-1} p^{u-1} + \dots + m_1 p + m_0$  com  $0 \leq m_s < p$ , para  $s = 0, 1, 2, \dots, u$  e  $m_u \neq 0$ ;  $\alpha = \alpha_t p^t + \alpha_{t-1} p^{t-1} + \dots + \alpha_1 p + \alpha_0$ , com  $0 \leq \alpha_l < p$ , para  $l = 0, 1, 2, \dots, t$  e  $\alpha_t \neq 0$ , se  $E_p((mp)!) = \alpha$ , então*

$$\alpha(p-1) = m_u(p^{u+1} - 1) + m_{u-1}(p^u - 1) + \dots + m_0(p-1).$$

Além disso,  $t \geq u$ .

*Demonstração.* Dado que  $mp$  é solução e aplicando o Teorema 4.10, temos que

$$\begin{aligned} \alpha = E_p((mp)!) &= \frac{mp - (m_0 + m_1 + \dots m_{u-1} + m_u)}{p-1} \Rightarrow \\ \alpha(p-1) &= mp - (m_0 + m_1 + \dots m_{u-1} + m_u) \Rightarrow \\ \alpha(p-1) &= m_u p^{u+1} + m_{u-1} p^u + \dots + m_1 p^2 + m_0 p - m_u - m_{u-1} - \dots - m_1 p - m_0 \Rightarrow \\ \alpha(p-1) &= m_u(p^{u+1} - 1) + m_{u-1}(p^u - 1) + \dots + m_0(p-1). \end{aligned}$$

Para provar que  $t \geq u$ , aplica-se a igualdade  $(p^{n+1} - 1) = (p-1)(p^n + p^{n-1} + \dots + p + 1)$ , obtendo

$$\alpha = m_u(p^u + p^{u-1} + \dots + p + 1) + m_{u-1}(p^{u-1} + p^{u-2} + \dots + p + 1) + \dots + m_1(p-1) + m_0.$$

Logo,  $p^u$  está na decomposição de  $\alpha$  e disto concluímos que  $t \geq u$ . □

**Lema 4.14.** *Seja  $p$  primo, se  $0 \leq m_s < p$ , para  $s = 1, 2, \dots, u-1$ , então  $m_{u-1}(p^u - 1) + \dots + m_0(p-1) < p^{u+1} - 1$ .*

*Demonstração.* Tomando que  $0 \leq m_s < p$ , para  $s = 1, 2, \dots, u-1$ , temos  $m_s(p^{s+1} - 1) \leq (p-1)(p^{s+1} - 1)$ , portanto

---

<sup>1</sup>Matemático egípcio, naturalizado brasileiro, que atua como professor titular no Instituto de Matemática da Universidade Federal Fluminense.

$$\begin{aligned}
 m_{u-1}(p^u - 1) + \dots + m_0(p - 1) &\leq (p - 1)(p^u - 1) + \dots + (p - 1)(p - 1) \Rightarrow \\
 m_{u-1}(p^u - 1) + \dots + m_0(p - 1) &\leq (p - 1)(p^u + p^{u-1} + \dots + p + 1 - u) \Rightarrow \\
 m_{u-1}(p^u - 1) + \dots + m_0(p - 1) &< (p - 1)(p^u + p^{u-1} + \dots + p + 1) \Rightarrow \\
 m_{u-1}(p^u - 1) + \dots + m_0(p - 1) &< (p - 1)\left(\frac{p^{u+1} - 1}{p - 1}\right) \Rightarrow \\
 m_{u-1}(p^u - 1) + \dots + m_0(p - 1) &< p^{u+1} - 1.
 \end{aligned}$$

□

**Corolário 4.15.** Tomando as hipóteses do Lema 4.13 temos que  $m_u = \left\lfloor \frac{\alpha(p-1)}{p^{u+1}-1} \right\rfloor$  e  $\alpha \geq m_u(p^u + p^{u-1} + \dots + p + 1)$ .

*Demonstração.* Usando o Lema 4.13, temos

$$\begin{aligned}
 \alpha(p-1) &= m_u(p^{u+1} - 1) + m_{u-1}(p^u - 1) + \dots + m_0(p - 1) \Rightarrow \\
 \frac{\alpha(p-1)}{(p^{u+1} - 1)} &= m_u + \frac{1}{(p^{u+1} - 1)}(m_{u-1}(p^u - 1) + \dots + m_0(p - 1)). \quad (4.1)
 \end{aligned}$$

Tomando o Lema 4.14 temos que

$$0 \leq \frac{1}{(p^{u+1} - 1)}(m_{u-1}(p^u - 1) + \dots + m_0(p - 1)) < 1.$$

Portanto, a parte inteira de ambos os lados da equação (4.1)  $m_u = \left\lfloor \frac{\alpha(p-1)}{p^{u+1}-1} \right\rfloor$ , em particular

$$\begin{aligned}
 m_u &\leq \frac{\alpha(p-1)}{p^{u+1}-1} \Rightarrow \\
 \alpha &\geq m_u(p^u + p^{u-1} + \dots + p + 1).
 \end{aligned}$$

□

**Proposição 4.16.** Fixados  $p$  primo e  $\alpha \in \mathbb{N}$ , tome  $u \in \mathbb{N}$ , tal que,  $m_u = \left\lfloor \frac{\alpha(p-1)}{p^{u+1}-1} \right\rfloor \neq 0$  e  $\left\lfloor \frac{\alpha(p-1)}{p^l-1} \right\rfloor = 0$ , para  $l > u + 1$ . Então  $m = m_u p^u + m_{u-1} p^{u-1} + \dots + m_1 p + m_0$ , para  $0 \leq m_s < p$ ,  $s = 1, 2, \dots, u$ , é solução de  $E_p((mp)!) = \alpha$  se, e só se,  $m' = m - m_u p^u$  é solução de  $E_p(m'!) = \alpha'$ , com  $\alpha' = \alpha - m_u(p^u + p^{u-1} + \dots + p + 1)$   $\alpha' \geq 0$ .

*Demonstração.* Observe inicialmente, que o Corolário 4.15 garante que  $\alpha' \geq 0$  e consequentemente que a equação  $E_p(m'!) = \alpha'$  é bem definida.

Como  $\left[ \frac{\alpha(p-1)}{p-1} \right] = \alpha \neq 0$  e a sequência  $x_n = \frac{\alpha(p-1)}{p^{n+1}-1}$  converge para zero, sempre existirá  $m_u$  nas condições exigidas. Note que, utilizando o Teorema 4.10,

$$\begin{aligned} E_p((mp)!) &= E_p(((m_u p^u + m')p)!) \\ &= \frac{(m_u p^u + m')p - (m_u + m_{u-1} + \dots + m_1 + m_0)}{p-1} \\ &= \frac{m'p - (m_{u-1} + \dots + m_1 + m_0)}{p-1} + \frac{m_u(p^{u+1} - 1)}{p-1} \\ &= E_p((m'p)!) + m_u(p^u + p^{u-1} + \dots + p + 1) \\ &= E_p((m'p)!) + \alpha - \alpha'. \end{aligned}$$

Isto é,

$$E_p((mp)!) = E_p((m'p)!) + \alpha - \alpha'.$$

Logo,  $E_p((mp)!) = \alpha$  se, e só se,  $E_p((m'p)!) = \alpha'$ .

□

**Proposição 4.17.** Tomando as notações do Lema 4.13 e  $\alpha = p^t + p^{t-1} + \dots + p$ , com  $t > 0$  temos que a equação não possui solução.

*Demonstração.* Segue do Lema 4.13 e do fato que  $n_j \leq p-1$  que,

$$\begin{aligned} \alpha(p-1) &\leq (p-1)(p^{u+1} + p^u + \dots + p + 1 - (u+1)) \Rightarrow \\ \alpha &\leq p^{u+1} + p^u + \dots + p + 1 - (u+1) < p^{u+1} + p^u + \dots + p + 1 \Rightarrow \\ p^t + p^{t-1} + \dots + p + 1 &< p^{u+1} + p^u + \dots + p + 1 \Rightarrow \\ t &< u + 1. \end{aligned}$$

Como  $t \geq u$ , chega-se a única possibilidade  $t = u$  daí,

$$\begin{aligned} m_u &= \left[ \frac{\alpha(p-1)}{p^{u+1}-1} \right] \\ &= \left[ \frac{(p^u + p^{u-1} + \dots + p)(p-1)}{p^{u+1}-1} \right] \\ &= \left[ \frac{p^{u+1} - p}{p^{u+1}-1} \right] = 0. \end{aligned}$$

O que é uma contradição pois  $m_u \neq 0$ .

□

Por fim, note que o Corolário 4.15 garante que a nova constante  $\alpha'$  nunca será negativa e a quantidade de reduções a problemas equivalentes são finitas, uma vez a cada passo, tem-se  $\alpha' < \alpha$ , levando a uma sequência  $\alpha > \alpha' > \dots > \alpha^{(h)} > \alpha^{(h+1)} = 0$ .



**Algoritmo 1.** Para resolver  $E_p(n!) = \alpha$ , com  $p$ , primo, e  $\alpha \in \mathbb{N} \cup \{0\}$  fixados previamente, inicia-se buscando  $m = m_u p^u + m_{u-1} p^{u-1} + \dots + m_1 p + m_0$  com  $m_u \neq 0$  e  $0 \leq m_s < p$  para  $s = 1, 2, \dots, u$ , tal que,  $E_p((mp)!) = \alpha$ . Para isso, seguimos os passos abaixo:

1º: Decomponha  $\alpha$  na forma  $p$ -ádica e avalie as situações abaixo:

- i) Se  $\alpha$  respeita a Proposição 4.17 segue que a equação não possui solução.
- ii) Se não ocorre i), então tome a Proposição 4.16 e reduza para o caso  $E_p((m'p)!) = \alpha'$ .

2º: Tomando  $E_p((m'p)!) = \alpha'$  reinicie os passos do algoritmo.

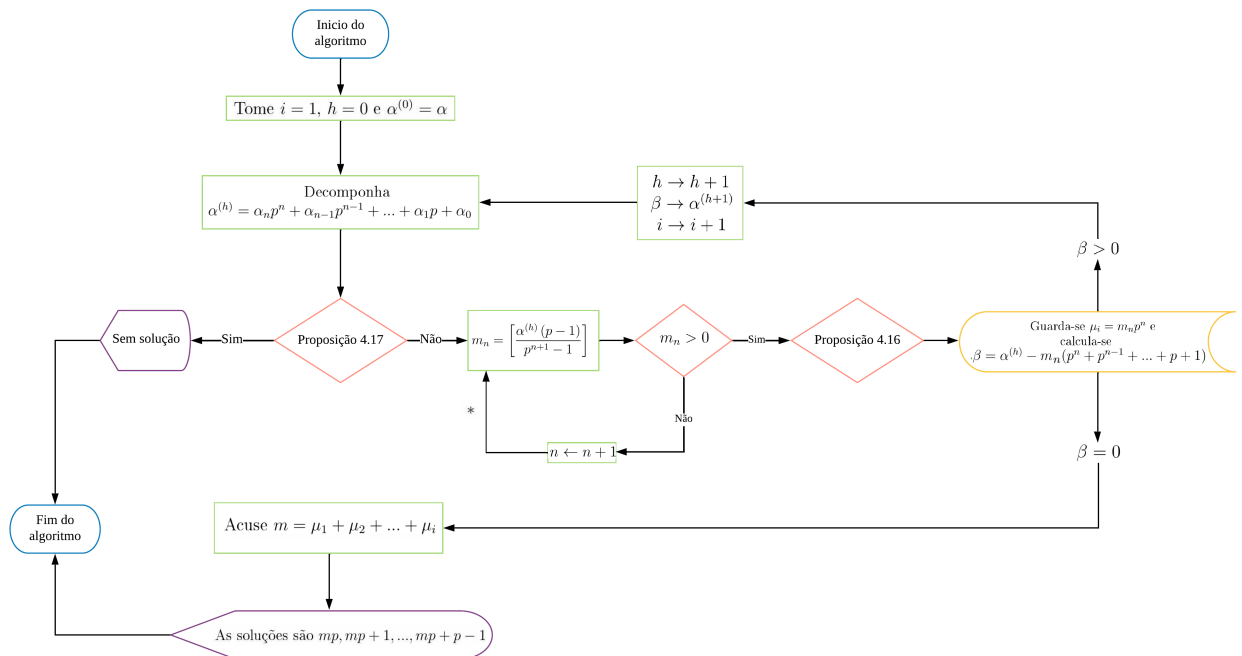
Repita os passos do algoritmo até que:

- 1: Encontre-se, para alguma redução, um valor de  $\alpha^{(h)}$ , que atenda a decomposição exposta na Proposição 4.17.
- 2: Encontre-se uma redução  $h$ , tal que,  $\alpha^{(h)} = 0$ .

Ao concluir o processo tem-se os coeficientes  $m_u, m_{u-1}, \dots, m_1$  e  $m_0$  e, conseqüentemente, encontra-se  $m = m_u p^u + m_{u-1} p^{u-1} + \dots + m_1 p + m_0$ .

Visualmente podemos expor as condições acima no fluxograma que segue.

Figura 4.1: Fluxograma relacionado aos passos lógicos interpretados no algoritmo apresentado por Abramo Hefez, encontrado no Livro Aritmética da coleção PROFMAT.



Fonte: Autoria nossa.

A seguir tomemos os exemplos que seguem afim de mostrar na prática o algoritmo exposto.

**Exemplo 4.18.** Encontre os valores de  $n$  que satisfazem  $E_7(n!) = 25$ .

*Solução.*

**1º Passo:** Decomponha  $\alpha$

$$25 = 3 \cdot 7 + 4$$

**2º Passo:** Existência de solução

Como  $3 \cdot 7 + 4 \neq 11$ , então temos solução.

**3º Passo:** Encontrando o maior coeficiente

$$m_1 = \left\lfloor \frac{25(7-1)}{7^2-1} \right\rfloor = 3.$$

**4º Passo:** Construção do caso equivalente

Como  $\alpha' = 25 - 3(7+1) = 1$ , temos que  $E_7(n!) = 25$  é equivalente a  $E_7(n!) = 1$ . E guardamos o valor  $\mu_1 = 3 \cdot 7 = 21$ .

Repetindo os processos para o caso  $E_7(n!) = 1$ .

**5º Passo:** Decomponha  $\alpha'$

$$1 = 1 \cdot 11^0$$

**6º Passo:** Existência de solução

Como  $1 \cdot 11^0 \neq 11$ , então temos solução.

**7º Passo:** Encontrando o maior coeficiente

$$m_0 = \left\lfloor \frac{1(7-1)}{7-1} \right\rfloor = 1.$$

**8º Passo:** Construção do caso equivalente

Como  $\alpha'' = 1 - 1(1) = 0$ , então guardamos o valor  $\mu_2 = 1$ .

Assim, concluímos que  $m = 21 + 1 = 22$ . Portanto,  $mp = 154$  é a menor solução da equação, concluindo que o conjunto

$$\{154, 155, 156, 157, 158, 159, 160\}$$

representa todas soluções.

◇

**Exemplo 4.19.** Encontre os valores de  $n$  que satisfazem  $E_{11}(n!) = 855$ .

*Solução.*

**1º Passo:** Decomponha  $\alpha$

$$855 = 7 \cdot 11^2 + 8$$

**2º Passo:** Existência de solução

Como  $7 \cdot 11^2 + 8 \neq 11^2 + 11 = 132$ , então temos solução.

**3º Passo:** Encontrando o maior coeficiente

$$m_2 = \left\lfloor \frac{855(11-1)}{11^3-1} \right\rfloor = 6.$$

**4º Passo:** Construção do caso equivalente

Como  $\alpha' = 855 - 6(11^2 + 11 + 1) = 57$ , temos que  $E_{11}(n!) = 855$  é equivalente a  $E_{11}(n') = 57$ . E guardamos o valor  $\mu_1 = 6 \cdot 11^2 = 726$ .

Repetindo os processos para o caso  $E_{11}(n') = 57$ .

**5º Passo:** Decomponha  $\alpha'$

$$57 = 5 \cdot 11 + 2$$

**6º Passo:** Existência de solução

Como  $5 \cdot 11 + 2 \neq 11$ , então temos solução.

**7º Passo:** Encontrando o maior coeficiente

$$m_1 = \left\lfloor \frac{57(11-1)}{11^2-1} \right\rfloor = 4.$$

**8º Passo:** Construção do caso equivalente

Como  $\alpha'' = 57 - 4(11 + 1) = 9$ , temos que  $E_{11}(n') = 57$  é equivalente a  $E_{11}(n'') = 9$ . E guardamos o valor  $\mu_2 = 4 \cdot 11 = 44$ .

Repetindo os processos para o caso  $E_{11}(n'') = 9$ .

**9º Passo:** Decomponha  $\alpha''$

$$9 = 9 \cdot 11^0$$

**10º Passo:** Existência de solução

Como  $9 \neq 11^0 = 1$ , então temos solução.

**11º Passo:** Encontrando o maior coeficiente

$$m_0 = \left\lfloor \frac{9(11-1)}{11^1-1} \right\rfloor = 9.$$

**12º Passo:** Construção do caso equivalente

Como  $\alpha''' = 9 - 9 = 0$ , então guardamos  $\mu_3 = 9 \cdot 11^0 = 9$

Assim, concluímos que  $m = 726 + 44 + 9 = 779$ . Portanto,  $mp = 8569$  é a menor solução da equação, concluindo que o conjunto

$$\{8569, 8570, 8571, 8572, 8573, 8574, 8575, 8576, 8577, 8578, 8579\}$$

representa todas soluções.

◇

**Exemplo 4.20.** Encontre os valores de  $n$  que satisfazem  $E_3(n!) = 92$ .

*Solução.*

Iniciemos, tomando que  $92 = 3^4 + 3^2 + 2$ , portanto  $\alpha$  não possui a decomposição igual a Proposição 4.17.

Tomando a Proposição 4.16, temos que

$$m_4 = \left\lfloor \frac{92(3-1)}{3^5-1} \right\rfloor = 0.$$

Reduzindo o índice do coeficiente para 3, temos

$$m_4 = \left\lfloor \frac{92(3-1)}{3^4-1} \right\rfloor = 2.$$

E, que  $\alpha' = 92 - 2(3^3 + 3^2 + 3 + 1) = 12$ . Portanto, reduzimos nosso problema para o caso  $E_3((3m')!) = 12$ .

Tomando  $12 = 3^2 + 3$ , portanto  $\alpha$  possui a decomposição igual a Proposição 4.17, concluindo que não existe solução.

◇

## 4.2.2 Algoritmo das matrizes, construído pelos autores

A partir de agora, apresentaremos um algoritmo diferente do apresentado por (HEFEZ, 2016) no livro Aritmética da coleção PROFMAT e, cuja parte teórica foi refinada nesse trabalho. Apesar de não sabermos se consiste em um método inédito, vale ressaltar que se trata de uma construção independente desenvolvida pelos autores, sendo que todos os resultados foram modelados ao decorrer da construção do trabalho.

Para tanto, tomemos o mesmo princípio utilizado no Algoritmo p-ádica, onde focamos em encontrar o menor natural solução da equação. Para isso, iniciemos com a definição abaixo.

**Definição 4.21.** Tomando  $p$  primo, seja  $M^1$  a matriz  $p \times p$ , formada pelos primeiros  $p^2$  múltiplos de  $p$  ordenados de forma crescente e dispostos em linhas,  $M^2$  a matriz formada pelos  $p^2$  seguintes, da mesma forma que a anterior, e assim sucessivamente, isto é,

$$M^1 = \begin{pmatrix} p & 2p & \cdots & p^2 \\ p^2 + p & p^2 + 2p & \cdots & 2p^2 \\ \vdots & \vdots & \ddots & \vdots \\ p^2(p-1) + p & p^2(p-1) + 2p & \cdots & p^3 \end{pmatrix};$$

$$M^2 = \begin{pmatrix} p^3 + p & p^3 + 2p & \cdots & p^3 + p^2 \\ p^3 + p^2 + p & p^3 + p^2 + 2p & \cdots & p^3 + 2p^2 \\ \vdots & \vdots & \ddots & \vdots \\ p^3 + p^2(p-1) + p & p^3 + p^2(p-1) + 2p & \cdots & 2p^3 \end{pmatrix}$$

E  $L^k$  a matriz  $p \times p$ , onde  $l_{ij}^k = E_p(m_{ij}^k)$ , onde  $m_{ij}^k$  é elemento da matriz  $M^k$ .

**Observação 4.22.** Temos para  $k = 1$  a matriz  $M^1$ , cujos elementos são os  $p^2$  primeiros múltiplos de  $p$ , começando em  $p$  e terminando em  $p^3$ . Já para  $k = 2$ , temos a matriz  $M^2$ , formada por  $p^2$  múltiplos de  $p$ , começando em  $p^3 + p$  e terminando em  $2p^3$ . E recorrentemente para os demais valores de  $k$ .

De forma genérica estamos tomando que,

$$M^k = \begin{pmatrix} m_{11}^k & m_{12}^k & \cdots & m_{1p}^k \\ m_{21}^k & m_{22}^k & \cdots & m_{2p}^k \\ \vdots & \vdots & \ddots & \vdots \\ m_{p1}^k & m_{p2}^k & \cdots & m_{pp}^k \end{pmatrix} \Rightarrow L^k = \begin{pmatrix} E_p(m_{11}^k) & E_p(m_{12}^k) & \cdots & E_p(m_{1p}^k) \\ E_p(m_{21}^k) & E_p(m_{22}^k) & \cdots & E_p(m_{2p}^k) \\ \vdots & \vdots & \ddots & \vdots \\ E_p(m_{p1}^k) & E_p(m_{p2}^k) & \cdots & E_p(m_{pp}^k) \end{pmatrix}.$$

**Exemplo 4.23.** Para  $p = 3$ , temos:

$$\begin{aligned} M^1 &= \begin{pmatrix} 3 & 6 & 9 \\ 12 & 15 & 18 \\ 21 & 24 & 27 \end{pmatrix} \Rightarrow L^1 = \begin{pmatrix} E_3(3) & E_3(6) & E_3(9) \\ E_3(12) & E_3(15) & E_3(18) \\ E_3(21) & E_3(24) & E_3(27) \end{pmatrix} = \begin{pmatrix} 1 & 1 & 2 \\ 1 & 1 & 2 \\ 1 & 1 & 3 \end{pmatrix}; \\ M^2 &= \begin{pmatrix} 30 & 33 & 36 \\ 39 & 42 & 45 \\ 48 & 51 & 54 \end{pmatrix} \Rightarrow L^2 = \begin{pmatrix} E_3(30) & E_3(33) & E_3(36) \\ E_3(39) & E_3(42) & E_3(45) \\ E_3(48) & E_3(51) & E_3(54) \end{pmatrix} = \begin{pmatrix} 1 & 1 & 2 \\ 1 & 1 & 2 \\ 1 & 1 & 3 \end{pmatrix}; \\ M^3 &= \begin{pmatrix} 57 & 60 & 63 \\ 66 & 69 & 72 \\ 75 & 78 & 81 \end{pmatrix} \Rightarrow L^3 = \begin{pmatrix} E_3(57) & E_3(60) & E_3(63) \\ E_3(66) & E_3(69) & E_3(72) \\ E_3(75) & E_3(78) & E_3(81) \end{pmatrix} = \begin{pmatrix} 1 & 1 & 2 \\ 1 & 1 & 2 \\ 1 & 1 & 4 \end{pmatrix}; \\ M^4 &= \begin{pmatrix} 84 & 87 & 90 \\ 93 & 96 & 99 \\ 102 & 105 & 108 \end{pmatrix} \Rightarrow L^4 = \begin{pmatrix} E_3(84) & E_3(87) & E_3(90) \\ E_3(93) & E_3(96) & E_3(99) \\ E_3(102) & E_3(105) & E_3(108) \end{pmatrix} = \begin{pmatrix} 1 & 1 & 2 \\ 1 & 1 & 2 \\ 1 & 1 & 3 \end{pmatrix}. \end{aligned}$$

**Observação 4.24.** No exemplo 4.23, note que:

$$E_3(9!) = E_3(9) + E_3(6) + E_3(3) = 4,$$

que coincide com a soma dos elementos destacados na matriz

$$L^1 = \begin{pmatrix} \mathbf{E_3(3)} & \mathbf{E_3(6)} & \mathbf{E_3(9)} \\ E_3(12) & E_3(15) & E_3(18) \\ E_3(21) & E_3(24) & E_3(27) \end{pmatrix} = \begin{pmatrix} \mathbf{1} & \mathbf{1} & \mathbf{2} \\ 1 & 1 & 2 \\ 1 & 1 & 3 \end{pmatrix}.$$

E

$$E_3(21!) = E_3(21) + E_3(18) + E_3(15) + E_3(12) + E_3(9) + E_3(6) + E_3(3) = 9,$$

o que também coincide com a soma dos elementos destacados na matriz abaixo

$$L^1 = \begin{pmatrix} \mathbf{E_3(3)} & \mathbf{E_3(6)} & \mathbf{E_3(9)} \\ \mathbf{E_3(12)} & \mathbf{E_3(15)} & \mathbf{E_3(18)} \\ \mathbf{E_3(21)} & E_3(24) & E_3(27) \end{pmatrix} = \begin{pmatrix} \mathbf{1} & \mathbf{1} & \mathbf{2} \\ \mathbf{1} & \mathbf{1} & \mathbf{2} \\ \mathbf{1} & 1 & 3 \end{pmatrix}.$$

Logo, o somatório de alguns elementos da matriz  $L^1$  ordenados de forma crescente e dispostos em linhas, parece coincidir com o cálculo da  $E_3(n!)$  sendo quando  $n$  é múltiplo de 3.

Usando o que já está definido vamos em busca de uma caracterização plena dos elementos de  $m_{ij}^k$ .

**Proposição 4.25.** Tomando  $p$  primo,  $k = 2, 3, 4, \dots$  e duas matrizes consecutivas temos que

$$m_{ij}^k = m_{ij}^{k-1} + p^3,$$

com  $i, j = \{1, 2, \dots, p\}$ .

*Demonstração.* Tomando a Definição 4.21 note que sempre existirão  $p^2$  múltiplos de  $p$  entre  $m_{ij}^k$  e  $m_{ij}^{k-1}$ , implicando que  $m_{ij}^k - m_{ij}^{k-1} = p \cdot p^2$ . Portanto

$$m_{ij}^k = m_{ij}^{k-1} + p^3.$$

□

**Proposição 4.26.** Tomando  $p$  primo e  $i, j = 1, 2, \dots, p$ , temos que os elementos de  $M^1$  são caracterizados por  $m_{ij}^1 = (i-1)p^2 + jp$ .

*Demonstração.* Tomando  $i = j = 1$ , temos que  $m_{11}^1 = p$ , conforme a Definição 4.21. Agora, por recorrência suponhamos que para algum  $i = 2, 3, \dots, p$  e  $j = 1, 2, \dots, p-1$ , temos valido  $m_{ij}^1 = (i-1)p^2 + jp$ . Logo,

$$\begin{aligned} m_{ij}^1 + p &= (i-1)p^2 + jp + p \Rightarrow \\ m_{i(j+1)}^1 &= (i-1)p^2 + (j+1)p. \end{aligned}$$

Por fim, por recorrência suponhamos que para algum  $i = 2, 3, \dots, p-1$  e  $j = p$  temos valido  $m_{ip}^1 = (i-1)p^2 + p^2$ . Logo,

$$\begin{aligned} m_{ip}^1 + p &= (i-1)p^2 + p^2 + p \Rightarrow \\ m_{(i+1)p}^1 &= ip^2 + p. \end{aligned}$$

□

**Lema 4.27.** Tomando  $M^k$ , tal que,  $k \in \mathbb{N}$  e  $i, j = 1, 2, \dots, p$ , com  $p$  primo, temos que

$$m_{ij}^k = (i - 1)p^2 + jp + (k - 1)p^3.$$

*Demonstração.* Tomando a Proposição 4.25, podemos dispor que:

$$\begin{aligned} m_{ij}^1 &= m_{ij}^1 \\ m_{ij}^2 &= m_{ij}^1 + p^3 \\ m_{ij}^3 &= m_{ij}^2 + p^3 \\ &\vdots \\ m_{ij}^{k-1} &= m_{ij}^{k-2} + p^3 \\ m_{ij}^k &= m_{ij}^{k-1} + p^3 \end{aligned}$$

Somando todos os elementos do membro da direita e todos os elementos do membro da esquerda, temos que:

$$\begin{aligned} \sum_{s=1}^k m_{ij}^s &= m_{ij}^1 + \sum_{s=1}^{k-1} m_{ij}^s + (k - 1)p^3 \Rightarrow \\ \sum_{s=1}^k m_{ij}^s - \sum_{s=1}^{k-1} m_{ij}^s &= m_{ij}^1 + \sum_{s=1}^{k-1} m_{ij}^s + (k - 1)p^3 - \sum_{s=1}^{k-1} m_{ij}^s \Rightarrow \\ m_{ij}^k &= m_{ij}^1 + (k - 1)p^3. \end{aligned}$$

Tomando a Proposição 4.26 concluímos que,

$$\begin{aligned} m_{ij}^k &= m_{ij}^1 + (k - 1)p^3 \Rightarrow \\ m_{ij}^k &= (i - 1)p^2 + jp + (k - 1)p^3. \end{aligned}$$

□

**Proposição 4.28.** Tomando  $n \in \mathbb{N}$ , múltiplo de  $p$ , a decomposição

$$n = (i - 1)p^2 + jp + (k - 1)p^3,$$

com  $i, j = 1, 2, \dots, p$  e  $k \in \mathbb{N}$  existe e é única.

*Demonstração.*

A existência segue do Lema 4.27, uma vez que as matrizes  $M^k$  contém todos os múltiplos de  $p$ .

Tomemos que  $n$  admita duas formas de decomposição, tal que,

$$\begin{aligned} n &= (i - 1)p^2 + jp + (k - 1)p^3 \\ n &= (i' - 1)p^2 + j'p + (k' - 1)p^3. \end{aligned}$$

Então,

$$\begin{aligned} 0 &= (i - i')p^2 + (j - j')p + (k - k')p^3 \Rightarrow \\ 0 &= j - j' + (i - i')p + (k - k')p^2 \Rightarrow \\ j' - j &= (i - i')p + (k - k')p^2 \Rightarrow \\ p &\mid (j' - j). \end{aligned}$$

Dado que  $j, j' = 1, 2, \dots, p$ , temos que  $0 \leq |j' - j| \leq p - 1$ . Logo,

$$\begin{aligned} p \mid (j' - j) &\Rightarrow \\ j' &= j. \end{aligned}$$

Assim, tendo  $j' = j$  e dispondo de raciocínio análogo temos que  $0 \leq |i' - i| \leq p - 1$ , logo  $i' = i$ . Por fim, dispondo que  $j' = j$  e  $i' = i$ , temos

$$\begin{aligned} 0 &= (k - k')p \Rightarrow \\ 0 &= k - k' \Rightarrow \\ k' &= k. \end{aligned}$$

Ficando devidamente provada a unicidade. □

**Corolário 4.29.** Fixos  $p$  primo e  $\alpha \in \mathbb{N}$ , se existir  $n!$ , tal que,  $E_p(n!) = \alpha$ , então existem  $i, j$  e  $k$ , tais que,  $E_p(m_{ij}^k!) = \alpha$ , sendo  $m_{ij}^k$  o menor natural solução da equação  $E_p(n!) = \alpha$ .

*Demonstração.* Segue do Corolário 4.12, que a menor solução é um múltiplo de  $p$ , logo o Lema 4.27 e a Proposição 4.28 garantem que tal múltiplo coincide com algum  $m_{ij}^k$ . □

**Proposição 4.30.** Fixado  $p$  primo e dados  $i, j \in \{1, 2, \dots, p\}$  e  $k \in \mathbb{N}$ , tem-se

$$E_p(m_{ij}^k!) = \sum_{l=1}^r E_p(lp),$$

sendo  $r$  o natural, tal que,  $m_{ij}^k = r \cdot p$ .

*Demonstração.* Tomando o item P1 da Proposição 2.13 note que

$$E_p(m_{ij}^k!) = E_p(1) + E_p(2) + \dots + E_p(m_{ij}^k).$$

Tomando o item i da Observação 2.4, concluímos que

$$E_p(m_{ij}^k!) = E_p(p) + E_p(2p) + \dots + E_p(m_{ij}^k) = \sum_{l=1}^r E_p(lp),$$

sendo  $r$  o natural, tal que,  $m_{ij}^k = r \cdot p$ . □



Vejamos agora um resultado que fornece os valores dos elementos da matriz  $L^k$ , mostrando que a lei de formação sempre terá um comportamento parecido com o visto no Exemplo 4.23.

**Proposição 4.31.** *Tomando  $p$  primo,  $i, j = 1, 2, \dots, p$  e  $k \in \mathbb{N}$ . Se  $m_{ij}^k = (i - 1)p^2 + jp + (k - 1)p^3$ , temos três possíveis casos:*

$$E_p(m_{ij}^k) = \begin{cases} 1, & \text{se } 1 \leq i \leq p \text{ e } 1 \leq j < p. \\ 2, & \text{se } 1 \leq i \leq p - 1 \text{ e } j = p. \\ 3 + E_p(k), & \text{se } i, j = p. \end{cases}$$

*Demonstração.* Seja o caso em que  $1 \leq i \leq p$  e  $1 \leq j < p$ . Como  $1 \leq j < p$  implica que  $p \nmid j$  e, por consequência,  $p \nmid (j + p(i - 1) + p^2(k - 1))$ .

Tomando o Lema 4.27  $m_{ij}^k = jp + p^2(i - 1) + p^3(k - 1) = p(j + p(i - 1) + p^2(k - 1))$ , então  $E_p(m_{ij}^k) = 1$ .

Os demais casos seguem de forma análoga ao descrito acima, no último caso é necessário escrever  $E_p(kp^3) = 3 + E_p(k)$ .  $\square$

Assim, caracterizamos qualquer matriz de ordem  $k$  como exposto abaixo.

**Corolário 4.32.** *Seja  $k \in \mathbb{N}$  e  $p$  primo, temos que*

$$L^k = \begin{pmatrix} 1 & 1 & \cdots & 1 & 2 \\ 1 & 1 & \cdots & 1 & 2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & 1 & \cdots & 1 & 2 \\ 1 & 1 & \cdots & 1 & 3 + E_p(k) \end{pmatrix}$$

**Definição 4.33.** Tome a matriz modelada no Corolário 4.32. Definimos  $\mathcal{L}_{ij}^k$  a soma dos elementos  $l_{ij'}^k$ , de forma que, sejam somados todos elementos das linhas  $i'$ , tal que,  $i' < i$  e todos elementos da linha  $i$  tal que  $j' \leq j$ .

**Exemplo 4.34.** Tomando  $p = 3$  e  $k = 6$ , temos que

$$L^6 = \begin{pmatrix} \mathbf{1} & \mathbf{1} & \mathbf{2} \\ 1 & 1 & 2 \\ 1 & 1 & 4 \end{pmatrix}.$$

Portanto,  $\mathcal{L}_{13}^6 = \mathbf{1} + \mathbf{1} + \mathbf{2} = 4$ . E, para a mesma matriz,

$$L^6 = \begin{pmatrix} \mathbf{1} & \mathbf{1} & \mathbf{2} \\ \mathbf{1} & \mathbf{1} & \mathbf{2} \\ \mathbf{1} & 1 & 4 \end{pmatrix}.$$

Logo,  $\mathcal{L}_{31}^6 = 2(\mathbf{1} + \mathbf{1} + \mathbf{2}) + \mathbf{1} = 9$ .

**Corolário 4.35.** *Tomando  $L^k$ , com  $k \in \mathbb{N}$ ,  $p$  primo e  $i, j = 1, 2, \dots, p$ , temos que*

i) Se  $i < p$ , desde que não ocorra, simultaneamente,  $i = p - 1$  e  $j = p$ , então  $\mathcal{L}_{(i+1)j}^k = \mathcal{L}_{ij}^k + (p + 1)$ .

ii) Se  $i = p - 1$  e  $j = p$ , então  $\mathcal{L}_{(i+1)j}^k = \mathcal{L}_{ij}^k + (p + 2) + E_p(k)$ .

*Demonstração.* Para o caso i), pelo Corolário 4.32, temos  $\mathcal{L}_{(i+1)j}^k - \mathcal{L}_{ij}^k = p + 1$ , portanto,  $\mathcal{L}_{(i+1)j}^k = \mathcal{L}_{ij}^k + (p + 1)$ . Para o caso ii) temos que  $\mathcal{L}_{pp}^k - \mathcal{L}_{(p-1)p}^k = p - 1 + 3 + E_p(k)$ , implicando que  $\mathcal{L}_{(i+1)j}^k = \mathcal{L}_{ij}^k + (p + 2) + E_p(k)$ .  $\square$

**Proposição 4.36.** Tomando  $p$  primo e  $i, j = 1, 2, \dots, p$ , desde que  $i$  e  $j$  não valham  $p$  simultaneamente, e  $k \in \mathbb{N}$ , temos que

$$\mathcal{L}_{ij}^k = i(p + 1) + (p + 1 - j) \left( \left[ \frac{j}{p} \right] - 1 \right)$$

*Demonstração.* Tomando  $i = 1$  e  $j < p$  temos que

$$\mathcal{L}_{ij}^k = j = 1(p + 1) + (p + 1 - j) \left( \left[ \frac{j}{p} \right] - 1 \right)$$

Suponhamos como hipótese de indução que para um  $i' < p$  temos  $\mathcal{L}_{i'j}^k = i'(p + 1) + (p + 1 - j) \left( \left[ \frac{j}{p} \right] - 1 \right)$  e usando o item i) do Corolário 4.35 temos que,

$$\mathcal{L}_{(i'+1)j}^k = \mathcal{L}_{i'j}^k + (p + 1) = i'(p + 1) + (p + 1 - j) \left( \left[ \frac{j}{p} \right] - 1 \right) + (p + 1) \Rightarrow$$

$$\mathcal{L}_{(i'+1)j}^k = (i' + 1)(p + 1) + (p + 1 - j) \left( \left[ \frac{j}{p} \right] - 1 \right).$$

Tomando  $j = p$ , dado  $i < p$  temos do Corolário 4.32 que

$$\mathcal{L}_{ip}^k = i(p + 1) = i(p + 1) + (p + 1 - p) \left( \left[ \frac{p}{p} \right] - 1 \right)$$

Agora, tomando  $i = p$  e  $j < p$ , temos do Corolário 4.32 que existem  $p - 1$  linhas com soma  $p + 1$  e na linha  $p$  temos  $j$  termos unitários.

$$\text{Logo, } \mathcal{L}_{pj}^k = (p - 1)(p + 1) + j = p(p + 1) + (p + 1 - j) \left( \left[ \frac{j}{p} \right] - 1 \right).$$

Assim, concluímos ser válido.  $\square$

Sobre a Proposição 4.36 notamos duas circunstâncias interessantes. Essas estão registradas e discutidas no que segue abaixo.

**Corolário 4.37.** Considerando as hipótese da Proposição 4.36, temos:

i) Se  $j = p$ , então  $\mathcal{L}_{ip}^k = i(p + 1)$ .

ii) Se  $j < p$ , então  $\mathcal{L}_{ij}^k = (i - 1)(p + 1) + j$ .

Quanto a soma de uma matriz, de forma integral  $L^k$ , fica evidente que o valor de  $k$  influenciará. Isso por que, temos que o elemento  $l_{pp}^k = 3 + E_p(k)$ . Assim, iniciaremos a discussão a cerca desses possíveis resultados.

**Definição 4.38.** Tomando  $k \in \mathbb{N}$  e a matriz caracterizada no Corolário 4.32, definimos  $S^k$  como a soma de todos os termos da matriz  $L^k$ . Particularmente, note que  $S^k = \mathcal{L}_{pp}^k$ .

**Proposição 4.39.** Tomando  $k \in \mathbb{N}$  e  $p$  primo, temos que  $S^k = p^2 + p + 1 + E_p(k)$ .

*Demonstração.* Tomando que  $S^k = \mathcal{L}_{p(p-1)} + l_{pp}^k$  e a Proposição 4.36 temos que,

$$\begin{aligned} S^k &= p(p+1) + (p+1 - p + 1) \left( \left[ \frac{p-1}{p} \right] - 1 \right) + E_p(k) + 3 \\ &= p(p+1) - 2 + E_p(k) + 3 \\ &= p^2 + p + 1 + E_p(k). \end{aligned}$$

□

**Lema 4.40.** Seja  $p$  primo,  $k \in \mathbb{N}$  e  $u \in \mathbb{N}$ , tais que,  $p^u \leq k < p^{u+1}$  temos,

$$\sum_{l=1}^k S^l = k(p^2 + p + 1) + E_p(k!),$$

ou

$$\sum_{l=1}^k S^l = k(p^2 + p + 1) + \left[ \frac{k}{p} \right] + \left[ \frac{k}{p^2} \right] + \dots + \left[ \frac{k}{p^u} \right].$$

*Demonstração.* Usando a Proposição 4.39, podemos escrever que:

$$\begin{aligned} S^1 &= p^2 + p + 1 + E_p(1) \\ S^2 &= p^2 + p + 1 + E_p(2) \\ &\vdots \\ S^k &= p^2 + p + 1 + E_p(k) \end{aligned}$$

Integrando os termos do membro da direita e os da esquerda, temos que

$$\sum_{l=1}^k S^l = k(p^2 + p + 1) + E_p(1) + E_p(2) + \dots + E_p(k).$$

Tomando o item P1, da Proposição 2.5, temos que:

$$\sum_{l=1}^k S^l = k(p^2 + p + 1) + E_p(k!),$$

ou, utilizando o Teorema de Legendre 4.5

$$\sum_{l=1}^k S^l = k(p^2 + p + 1) + \left[ \frac{k}{p} \right] + \left[ \frac{k}{p^2} \right] + \dots + \left[ \frac{k}{p^u} \right].$$

□

Após o desenvolvimento teórico, apresentamos os principais resultados desta seção.

**Proposição 4.41.** *Fixado  $p$  primo, temos*

$$E_p(m_{ij}^k!) = \sum_{l=1}^{k-1} S^l + \mathcal{L}_{ij}^k.$$

*Demonstração.* Com argumento análogo ao da demonstração da Proposição 4.30, podemos escrever

$$\begin{aligned} E_p(m_{ij}^k!) &= E_p(1) + E_p(2) + \dots + E_p(m_{ij}^k) \\ &= E_p(p) + E_p(2p) + \dots + E_p(m_{ij}^k) \\ &= E_p(m_{11}^1) + E_p(m_{12}^1) + \dots + E_p(m_{ij}^k) \\ &= S^1 + S^2 + \dots + S^{k-1} + \mathcal{L}_{ij}^k. \end{aligned}$$

□

**Corolário 4.42.** *A equação  $E_p(n!) = \alpha$ , tem solução se, e só se, existir  $i, j$  e  $k$ , tal que*

$$\alpha = \sum_{l=1}^{k-1} S^l + \mathcal{L}_{ij}^k = k(p^2 + p + 1) + E_p(k!) + i(p + 1) + (p + 1 - j) \left( \left[ \frac{j}{p} \right] - 1 \right).$$

*Demonstração.* Basta observar o Corolário 4.29, existindo solução, haverá uma solução escrita como múltiplo de  $p$ . □

**Corolário 4.43.** *Fixados  $p$  primo e  $\alpha \in \mathbb{N}$ , se  $\alpha = \sum_{l=1}^k S^l$ , para algum  $k \in \mathbb{N}$ , então  $E_p(m_{pp}^k!) = \alpha$ .*

**Lema 4.44.** *Fixados  $p$  primo e  $\alpha \in \mathbb{N}$  seja  $k \in \mathbb{N}$ , tal que,  $\sum_{l=1}^{k-1} S^l < \alpha < \sum_{l=1}^k S^l$ . Temos:*

- i) *Se  $\alpha - \sum_{l=1}^{k-1} S^l \geq p^2 + p - 1$ , então  $E_p(n!) = \alpha$  não possui solução.*
- ii) *Se  $\alpha - \sum_{l=1}^{k-1} S^l < p^2 + p - 1$  e  $\alpha - \sum_{l=1}^{k-1} S^l \equiv p \pmod{p+1}$ , então  $E_p(n!) = \alpha$  não possui solução.*
- iii) *Se  $\alpha - \sum_{l=1}^{k-1} S^l < p^2 + p$  e  $\alpha - \sum_{l=1}^{k-1} S^l \equiv 0 \pmod{p+1}$ , então  $E_p(m_{ij}^k!) = \alpha$ , com  $j = p$  e  $i = \frac{\alpha - \sum_{l=1}^{k-1} S^l}{p+1}$ .*
- iv) *Se  $\alpha - \sum_{l=1}^{k-1} S^l < p^2 + p$  e  $\alpha - \sum_{l=1}^{k-1} S^l \equiv j \pmod{p+1}$ , com  $j = \{1, 2, \dots, p-1\}$ , então  $E_p(m_{ij}^k!) = \alpha$ , com  $i = \frac{\alpha - \sum_{l=1}^{k-1} S^l - j}{p+1} + 1$ .*

*Demonstração.*

- i) Tomando o Corolário 4.32 temos  $\alpha - \sum_{l=1}^{k-1} S^l \leq (p-1)(p+1) + p - 1$ , implicando que  $\alpha - \sum_{l=1}^{k-1} S^l \leq p^2 + p - 2$ . Logo, se  $\alpha - \sum_{l=1}^{k-1} S^l \geq p^2 + p - 1$ , então  $E_p(n!) = \alpha$  não possui solução.
- ii) Dado  $i + j < 2p$  e o Corolário 4.37, note que não existe  $i$  e  $j$ , tal que,  $\mathcal{L}_{ij}^k \equiv p \pmod{p+1}$ , implicando que a equação  $E_p(m_{ij}^k!) = \alpha$  não tem solução.
- iii) Basta tomar o item i) do Corolário 4.37 garantindo que

$$\mathcal{L}_{ij}^k = i(p+1) = \frac{\left(\alpha - \sum_{l=1}^{k-1} S^l\right)(p+1)}{p+1} = \alpha - \sum_{l=1}^{k-1} S^l$$

e a Proposição 4.41 que garante  $E_p(m_{ip}^k!) = \alpha$ .

- iv) Análogo a iii), utilizando o item ii) do Corolário 4.37.

□

**Algoritmo 2.** Dada  $E_p(n!) = \alpha$ , com  $p$  primo e  $\alpha \in \mathbb{N} \cup \{0\}$  fixos, siga os passos que seguem, a fim de encontrar os valores de  $i, j$  e  $k$ , que com a aplicação do Lema 4.27, possibilita encontrar o menor natural  $n$  que satisfaz a equação.

1º Passo: Aplique o Lema 4.40.

Se  $\alpha = \sum_{l=1}^k S^l$  adote  $i = j = p$  e aplique o Lema 4.27 e vá para o 4º passo.

Se  $\sum_{l=1}^{k-1} S^l < \alpha < \sum_{l=1}^k S^l$ , então obtêm-se o valor de  $k$  e vá para o 2º passo.

2º Passo: Aplique o Lema 4.44.

Se respeitar os itens i) ou ii), então a equação não possui solução.

Se respeitar os itens iii) ou iv) prossiga para o 3º passo.

3º passo: Divida por  $p+1$ .

Se for divisível acuse  $j = p$  e  $i = \frac{\alpha - \sum_{l=1}^{k-1} S^l}{p+1}$  e vá para o 4º passo.

Se não for divisível acuse  $\alpha - \sum_{l=1}^{k-1} S^l \equiv j \pmod{p+1}$ , e  $i = \frac{\alpha - \sum_{l=1}^{k-1} S^l - j}{p+1} + 1$ .  
e vá para o 4º passo.

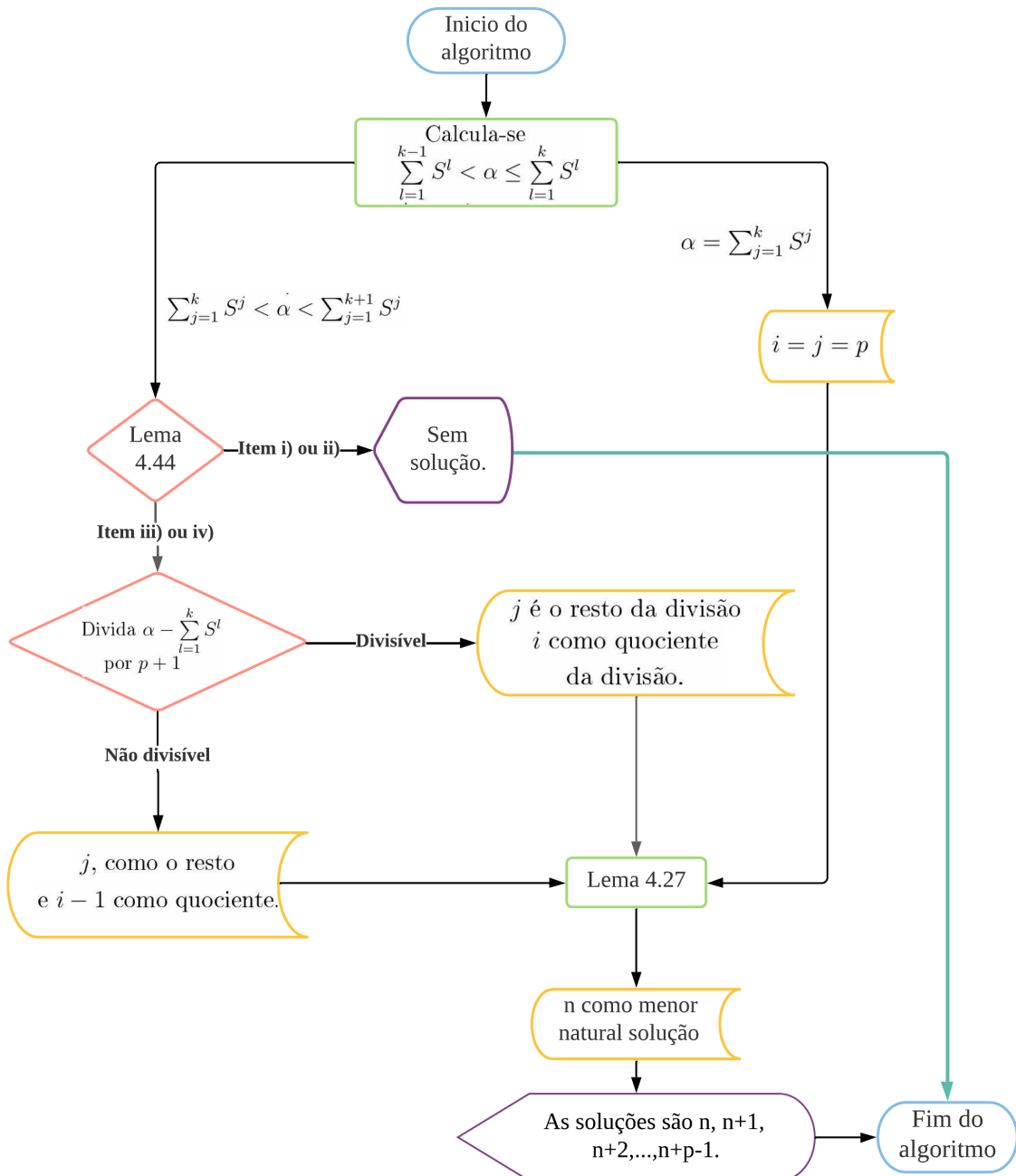
4º Passo: Aplique o Lema 4.27 a fim de obter o valor de  $n$

Indique que as soluções possíveis são dadas pelo conjunto abaixo:

$$\{n, n + 1, \dots, n + (p - 1)\}.$$

Simplificando, por meio de um diagrama temos que:

Figura 4.2: Fluxograma relacionado aos passos lógicos interpretados no algoritmo das matrizes, desenvolvido de forma inédita pelos autores.



Fonte: Autoria nossa

A seguir tomemos dois exemplos, mostrando a prática do algoritmo desenvolvido.

**Exemplo 4.45.** Encontre os valores de  $n$  que satisfaz  $E_7(n!) = 25$ .

*Solução.*

**1º Passo:** Condição do  $\alpha$

$$25 < 57 = \sum_{r=1}^1 S^r,$$

Assim, logo  $k = 1$ .

**2º Passo:** Condição de existência

Dado que,  $25 < 11^2 + 11$ , temos que a equação possui solução.

**3º Passo:** Divisibilidade

$25 = 3 \cdot (7 + 1) + 1$ , então  $i - 1 = 3$ , logo  $i = 4$  e  $j = 1$ .

**4º Passo:** Menor natural solução

Portanto,  $m_{41}^1 = 7 + 7^2(4 - 1) + 7^3(1 - 1) = 154$ .

**5º Passo:** Solução

Logo, o conjunto

$$\{154, 155, 156, 157, 158, 159, 160\}$$

representa todas soluções.

◇

**Exemplo 4.46.** Encontre os valores de  $n$  que satisfazem  $E_{11}(n!) = 855$ .

*Solução.*

**1º Passo:** Condição do  $\alpha$

$$\sum_{r=1}^6 S^r = 798 < 855 < 931 = \sum_{r=1}^7 S^r,$$

Assim, logo  $k = 7$ .

**2º Passo:** Condição de existência

Dado que,  $57 < 11^2 + 11$ , temos que a equação possui solução.

**3º Passo:** Divisibilidade

$57 = 4 \cdot (11 + 1) + 9$ , então  $i - 1 = 4$ , logo  $i = 5$  e  $j = 9$ .

**4º Passo:** Menor natural solução

Portanto,  $m_{59}^7 = 11 \cdot 9 + 11^2(5 - 1) + 11^3(7 - 1) = 8569$ .

**5º Passo:** Solução

Logo, o conjunto

$$\{8569, 8570, 8571, 8572, 8573, 8574, 8575, 8576, 8577, 8578, 8579\}$$

representa todas soluções.

◇

### 4.2.3 Duelo computacional: p-ádica X matrizes

Nesse momento estamos interessados em testar computacionalmente o Algoritmo da p-ádica, proposto por Abramo Hefez, e o das matrizes, desenvolvido de forma independente nesse trabalho.

**Observação 4.47.** A análise é baseada em um recorte de eventos.

**Observação 4.48.** Entendemos como mais eficiente, o algoritmo que desenvolve um tempo de processamento<sup>2</sup> inferior, em comparação ao outro, sobre as mesmas condições.

Para desenvolvimento dos testes foram confeccionadas duas sequências de linhas de programação<sup>3</sup> com auxílio do VisualG. As mesmas tem por objetivo, encontrar as soluções para a equação  $E_p(n!) = \alpha$ , sendo fixos diferentes valores de  $\alpha \in \mathbb{N} \cup \{0\}$  e  $p$  primo. Para cada variação marca-se o tempo (variável  $t$ ), em  $ms$  (milissegundos).

Marcado o tempo para os casos em que  $\alpha$  e  $p$  primo variam constituímos pontos com três coordenadas  $(\alpha, p, t)$ , assim gerando dois gráficos tridimensionais, exposto a seguir.

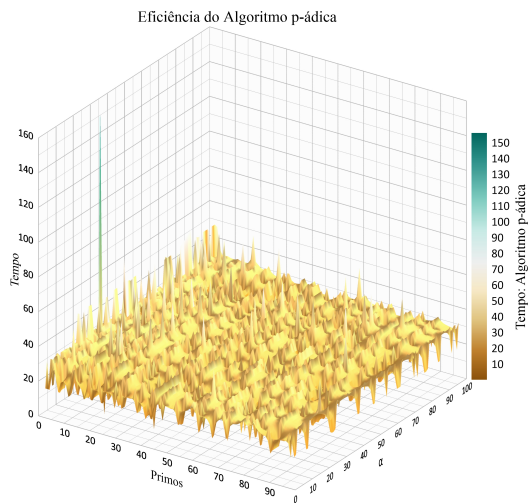
---

<sup>2</sup>Tempo em que o algoritmo levou, desde seu início até seu término, para exibir a resposta.

<sup>3</sup>Linhas disponíveis no Apêndice B

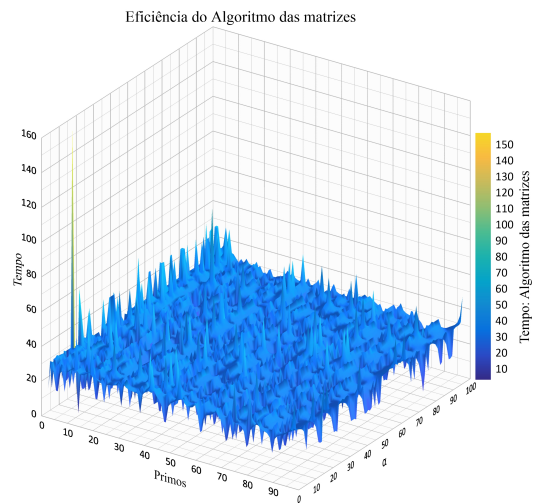


Figura 4.3: Variação  $\alpha \times p \times t$  para o Algoritmo p-ádica.



Fonte: Autoria nossa

Figura 4.4: Variação  $\alpha \times p \times t$  para o Algoritmo das matrizes.

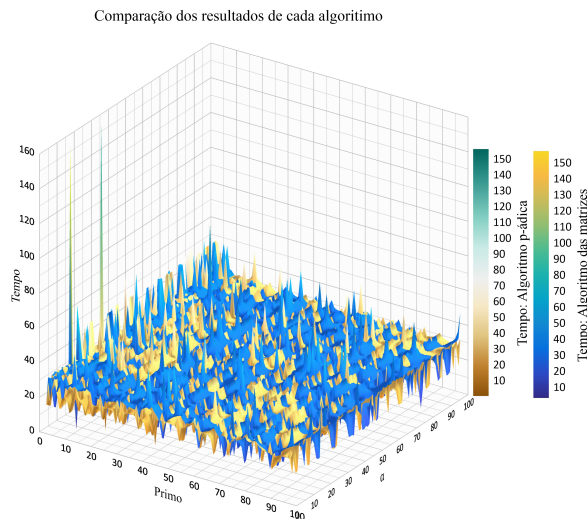


Fonte: Autoria nossa

Para os dados apresentados nas Figuras 4.3 e 4.4, foram tomados valores de  $\alpha$  entre 1 e 100 e  $p$  primos entre 2 e 100, uma quantidade de 25 primos. Por fim, para cada variação  $\alpha \times p$  foi registrado o tempo ( $t$ ) gasto para obtermos a solução. Logo, temos plotados 2500 pontos para cada gráfico.

Sobrepondo as superfícies apresentadas nas Figuras 4.3 e 4.4, passamos a perceber uma similaridade para os tempos de processamento.

Figura 4.5: Sobreposição dos dados apresentados pelas Figuras 4.3 e 4.4.



Fonte: Autoria nossa

Afim de entendermos os eventos dispostos no gráfico da Figura 4.5 dividiremos nossa análise em três momentos:

- i. Tempo médio desenvolvido em ambos os algoritmos para os pontos plotados.
- ii. Subtração entre o tempo do Algoritmo das matrizes pelo da p-ádica.
- iii. Para os casos em que cada algoritmo é mais eficiente analisamos a amplitude entre os tempos.

**Observação 4.49.** Nesse recorte não buscamos caracterizar as circunstâncias em que  $\alpha$  e  $p$  primo tornam um algoritmo ou outro mais eficiente.

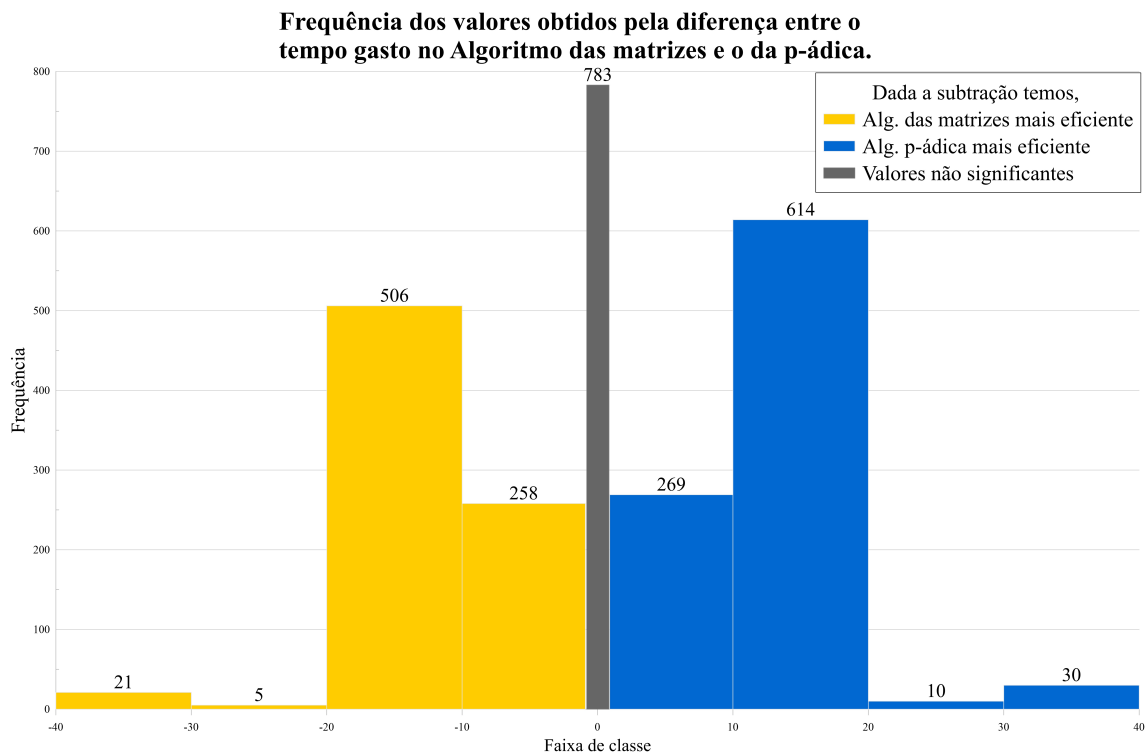
Dado os 2500 pontos, para cada algoritmo, conseguimos obter que o tempo médio do Algoritmo p-ádica é cerca de 26,86 *ms* e o Algoritmo das matrizes 27,76 *ms*. A partir desses dados, temos que o Algoritmo p-ádica tem uma média de tempo 3% menor quando comparado ao Algoritmo das matrizes.

Obtidos os tempos de processamento para cada algoritmo, os mesmos foram tabelados. Após isso, foi efetuada a subtração entre o tempo de processamento do Algoritmo das matrizes pelo tempo do Algoritmo p-ádica.

**Observação 4.50.** Tomando os valores da subtração entre o tempo de processamento do Algoritmo das matrizes pelo Algoritmo p-ádica, temos que

- i. **Positivos e maiores que 1:** Algoritmo da p-ádica mais eficiente. (924 casos)
- ii. **Valores entre -1 e 1:** a diferença entre os tempos de processamento não foram consideradas significativas<sup>4</sup>. (783 casos)
- iii. **Negativos e menores que -1:** algoritmo das matrizes mais eficiente. (793 casos)

Figura 4.6: Histograma de frequência baseado na Observação 4.50.



Fonte: Autoria nossa

Abaixo segue uma tabela que mostram 4 pontos que não foram abordados no Histograma disposto na Figura 4.6.

<sup>4</sup>Devido as limitações do software utilizado para 783 casos em que a diferença entre os valores é nula, não conseguimos concluir se os tempos realmente são iguais ou ficaram após serem arredondados.

Tabela 4.4: Tabela dos quatro valores retirados para a construção do Histograma representado na Figura 4.6, por conta de suas disparidades com os demais pontos. Da esquerda para direita temos na primeira coluna os valores de  $\alpha$  e  $p$  primo fixados, segunda coluna o tempo de processamento do Algoritmo das matrizes e na terceira o tempo para o Algoritmo p-ádica e na quarta coluna a diferença entre os tempos da segunda pela terceira coluna.

Casos $(\alpha, p)$	Algoritmo das matrizes	Algoritmo p-ádica	Diferença
(33, 2)	47	156	-109
(19, 2)	62	16	46
(80, 13)	74	16	58
(15, 2)	157	31	126

Fonte: Autoria nossa.

Os pontos citados acima foram retirados da análise, pois os mesmos produzem resultados, após a diferença, que destoam significadamente da grande maioria dos demais pontos.

Com base nos dados destacados, observamos que nenhum dos algoritmos se sobressai de forma significativa ao outro, nessa análise. Imaginamos que cada um deles deve funcionar melhor em uma determinada situação, embora essa seja uma especulação nossa, sem aprofundamento.

### 4.3 Estudo da solução fixando $n$ e $\alpha$ .

Tomemos  $E_p(n!) = \alpha$ , com  $\alpha$  e  $n$  previamente fixados. Portanto, temos os valores de  $p$  primo como variável livre e desejamos determinar tais valores que satisfazem a equação.

Visando uma melhor organização subdividiremos nossas abordagens como seguem abaixo:

Tabela 4.5: Indicativo e localização de cada caso abordado na Sub-seção 4.3.

Estudo	Resultado
Existência	Proposição 4.51 Proposição 4.52
Caracterização	Lema4.53 Lema4.55

Fonte: Autoria nossa.

Pelo TFA temos que  $n! = p_1^{\alpha_1} p_2^{\alpha_2} \dots p_s^{\alpha_s}$ . Logo, fica de fácil observação que para  $p$  ser solução da equação fica necessário que o mesmo seja menor ou igual a  $n!$ .

**Proposição 4.51.** *Fixados  $n \in \mathbb{N}$  e  $\alpha = 0$ , temos infinitas soluções para a equação  $E_p(n!) = \alpha$ .*

*Demonstração.* Se  $p$  é primo e  $p > n$ , então  $p$  não aparece na decomposição de  $n!$ . □

**Proposição 4.52.** *Fixados  $n, \alpha \in \mathbb{N}$  se  $p$  é, tal que,  $\left[ \frac{n}{p} \right] > \alpha$ , então  $p$  não satisfaz a equação  $E_p(n!) = \alpha$ .*

*Demonstração.* De fato, suponha que  $E_p(n!) = \alpha$ . Pelo Teorema de Legendre (4.5),  $\alpha = \left[ \frac{n}{p} \right] + \left[ \frac{n}{p^2} \right] + \dots + \left[ \frac{n}{p^u} \right]$ , com  $p^u \leq n < p^{u+1}$ , implicando que  $\left[ \frac{n}{p} \right] \leq \alpha$ . □

**Lema 4.53.** Fixados  $n, \alpha \in \mathbb{N}$ , se  $p$  satisfaz a equação  $E_p(n!) = \alpha$ , então

$$\frac{n+1}{\alpha+1} \leq p \leq n.$$

*Demonstração.* Tomando  $n = p \left\lfloor \frac{n}{p} \right\rfloor + r$ , com  $0 \leq r < p$  e a Proposição 4.52 temos,

$$\begin{aligned} \frac{n-r}{p} &= \left\lfloor \frac{n}{p} \right\rfloor \leq \alpha \Rightarrow \\ \frac{n-r}{p} &\leq \alpha \Rightarrow \\ -r &\leq p\alpha - n \Rightarrow \\ r &\geq n - p\alpha. \end{aligned}$$

Como  $r < p$ , temos

$$\begin{aligned} n - p\alpha &\leq p - 1 \Rightarrow \\ n + 1 &\leq p + p\alpha \Rightarrow \\ n + 1 &\leq p(1 + \alpha) \Rightarrow \\ \frac{n+1}{\alpha+1} &\leq p. \end{aligned}$$

Logo,  $\frac{n+1}{\alpha+1} \leq p \leq n$ . □

**Observação 4.54.** Com base no Lema 4.53 e Teorema de Legendre 4.5, deve-se testar os primos do intervalo  $\left[ \frac{n+1}{\alpha+1}, n \right]$ , um primo no referido intervalo será solução se, e só se, valer a igualdade  $\alpha = \left\lfloor \frac{n}{p} \right\rfloor + \left\lfloor \frac{n}{p^2} \right\rfloor + \dots + \left\lfloor \frac{n}{p^u} \right\rfloor$ , com  $p^u \leq n < p^{u+1}$ .

**Lema 4.55.** Fixados  $n, \alpha \in \mathbb{N}$ , para cada  $p$  primo no intervalo do Lema 4.53, tome  $u \in \mathbb{N}$  tal que  $p^u \leq n < p^{u+1}$ , então  $p$  é solução se, e só se,  $\alpha = \left\lfloor \frac{n}{p} \right\rfloor + \left\lfloor \frac{n}{p^2} \right\rfloor + \dots + \left\lfloor \frac{n}{p^u} \right\rfloor$ .

*Demonstração.* Se  $p$  é solução então ele respeita o Lema 4.53, como o Teorema de Legendre 4.5, implicando que  $\alpha = \left\lfloor \frac{n}{p} \right\rfloor + \left\lfloor \frac{n}{p^2} \right\rfloor + \dots + \left\lfloor \frac{n}{p^u} \right\rfloor$ , com  $p^u \leq n < p^{u+1}$ .

Se  $\alpha = \left\lfloor \frac{n}{p} \right\rfloor + \left\lfloor \frac{n}{p^2} \right\rfloor + \dots + \left\lfloor \frac{n}{p^u} \right\rfloor$ , com  $p^u \leq n < p^{u+1}$ , então  $E_p(n!) = \alpha$ , sendo  $p$  primo solução da equação e, portanto, respeitando o Lema 4.53. □

# 5 Uso de linguagem de programação no processo de ensino e aprendizagem

Nesse capítulo estudaremos a ideia de pensamento computacional, bem como, sua relação com o ensino da matemática, focando no ambiente da educação básica. Vale dizer que ao final do capítulo, descreveremos uma atividade elaborada pelos autores, que seja aplicável aos estudantes desse ciclo, que mescele a ideia de pensamento computacional, como linhas de programação e conceitos matemáticos.

## 5.1 O “Pensamento computacional” e o ensino da matemática

Como início de nossas abordagens investigaremos o que seria e como ocorre o pensamento computacional e qual sua associação com o ensino da matemática. Para isso tomaremos de pesquisadores referências nessas discussões, a fim de garantir um embasamento teórico a nossa atividade.

### 5.1.1 Pensamento computacional

Encontrar uma definição para pensamento computacional, que agrade a todos estudiosos da área vem sendo um processo complicado, como aponta [VALENTE \(2016\)](#).

Pensando sobre um ponto de vista cronológico, o termo (pensamento computacional) surge com Jannette M. Wing, que coloca-o como o processos de raciocínio que envolve a resolução de problemas, bem como sua formulação, onde um computador (humano ou máquina), possa resolve-lo ([WING \(2014\)](#)).

Esse processo de raciocínio pode ser entendido sobre qualquer ação, por exemplo, uma criança que deva escolher entre seus objetos aqueles que porá em sua bolsa (**ideia de escolhas e prioridades**). Ou o ato de tentar encontrar algo perdido recriando os passos anteriores ([WING \(2014\)](#)).

É fato que o termo pensamento computacional nos remete a ideia de tecnologias digitais, mas como aponta [WING \(2014\)](#) e [VALENTE \(2016\)](#) essa associação não é necessária. Todos desenvolvemos pensamento computacional assim que nos propomos a resolver ou gerar uma problemática.

[VALENTE \(2016\)](#) deixa claro que, apesar da introdução de tecnologias digitais, no processo de pensamento computacional, não ser pré requisito, nem determinante para um

sucesso ou não, a inserção de tais tecnologias promovem uma melhoria em todo processo.

Pensando para crianças, podemos entender a melhoria do pensamento computacional quando vinculado a tecnologia, sobre o que aponta VALENTE (2016) destacando que a presença das tecnologias digitais de informação e comunicação vem reestruturando o funcionamento de setores ligado a organização econômica, social e cultural, o que pode ser percebido em nossas ações rotineiras de intercomunicação e pela forma como atuamos em transações financeiras. É Papert, que destaca o estímulo a criatividade das crianças e, dessa forma, dos estudantes.

O processo do pensamento computacional fundamentasse na abstração. Este permite que criemos padrões, generalizações e passemos a entender as propriedades formadoras de um único objeto ou seu conjunto.

So, the most important and high-level thought process in computational thinking is the abstraction process. Abstraction is used in defining patterns, generalizing from specific instances, and parameterization. It is used to let one object stand for many. It is used to capture essential properties common to a set of objects while hiding irrelevant distinctions among them. For example, an algorithm is an abstraction of a process that takes inputs, executes a sequence of steps, and produces outputs to satisfy a desired goal. WING (2014)

Assim, pensar computacionalmente, vem de um estímulo da abstração e pela necessidade e/ou vontade de solucionar, bem como criar, problemas, sejam esses de qualquer natureza. Pensar nesse processo, significa entender como o computador (homem ou máquina) processa e organiza as informações para uma otimização das ações que podem ser tomadas frente a tais circunstâncias.

### 5.1.2 Ensino da matemática: pensamento computacional

Investigar a relação entre pensamento computacional e o ensino da matemática, passa pelos entendimentos do que seria o pensamento computacional e como o processo de ensino da matemática se desenvolveu nos últimos anos.

Inicialmente, é perceptível que o uso de computadores trouxe uma reflexão sobre a ação do homem para a resolução de problemas. Sendo válida a ideia de quando a utilidade do homem ou da máquina é conveniente e para quais problemas.

Apear do vínculo que os termos “computador” e “pensamento computacional” possuem, quase que de forma evidente, o pensamento computacional não se restringe a ideia de programar um computador para fazer tarefas, das simples as mais complexas. Este pelo simples ato de raciocinar a solução ou criação de um problema.

SOUTO (2017) destaca a colocação de Onuchic, aonde ressalta que na década de 1930 a matemática começava se desvincilhar da ideia de repetição e memorização passando a dar mais importância a forma com que o estudante aprende e se posta frente as dificuldades encontradas.

Assim concluí SOUTO (2017), que essa mudança de pensamento traz um questionamento sobre a prática de como resolver problemas, sendo esta de importância fundamental para o desenvolvimento do estudante como um todo, envolvendo as partes técnicas e criativas, como a elaboração de hipóteses e reflexões.

Tomando a necessidade de discutir o ensino da matemática e a crescente presença dos meios digitais em nossa sociedade, é sensível a necessidade de estudarmos como estender

e aplicar, de forma significativa, o pensamento computacional no ensino, particularizando nesse caso ao estudo da matemática.

Dessa forma, concluímos que a discussão entre pensamento computacional e ensino da matemática permeia uma necessidade atual que o processo de ensino e aprendizagem deve modelar-se. Mais que isso, pensar em como inserir a tecnologia digital (um dos meios pelo qual o pensamento computacional pode ser desenvolvido), de forma significativa, é discussão fundamental para alicerçar uma educação de qualidade e contextualizada com as recentes demandas sociais, econômicas.

## 5.2 Ação nacional de educação e o pensamento computacional

Como vimos, ter em mente o Pensamento computacional pode ser de grande valia para o sistema educacional. Isso, por que, como aponta Papert, este pode estimular a criatividade das crianças e, dessa forma, dos estudantes.

Além disso, vimos que os meios digitais estão inseridos em todas as ações sociais, econômicas e culturais, por isso, fica de extrema importância integrar tal realidade as vivências dos estudantes, assim produzindo um maior interesse por parte dos mesmos as práticas de ensino.

Ações de integração, como a descrita anteriormente, é algo que já vem sendo executado em alguns países. Exemplo disso, é o programa “*Computer Science is for Everone*”, lançado pelo governo dos Estados Unidos da América, que visa uma das maneiras de adaptar o uso de tecnologia aos diferentes contextos do nosso dia-a-dia (VALENTE (2016)).

Outro exemplo dessa prática esta no relatório “*European Schoolnet*”. Nele ficam evidentes que, dos 20 países pesquisados, 13 já alteraram suas diretrizes educacionais, sendo destes, a Inglaterra a instituir Ciências da Computação, Tecnologia da Informação e o Letramento digital, como obrigatórios no currículo da educação básica (VALENTE (2016)).

No Brasil, os avanços frente a essas tendências, ficam relativas a construção da BNCC (Base Nacional Comum Curricular). Neste documento, é possível notar a existência de habilidades que ressaltam o uso de ferramentas tecnológicas, bem como, aspectos do pensamento computacional, em assuntos antes trabalhados por meio repetição e memorização.

A primeira e segunda competências gerais, destacadas a seguir, mostram as preocupações destacadas anteriormente.

Reconhecer que a Matemática é uma ciência humana, fruto das necessidades e preocupações de diferentes culturas, em diferentes momentos históricos, e é uma ciência viva, que contribui para solucionar problemas científicos e tecnológicos e para alicerçar descobertas e construções, inclusive com impactos no mundo do trabalho. (BNCC (2018))

Utilizar processos e ferramentas matemáticas, inclusive tecnologias digitais disponíveis, para modelar e resolver problemas cotidianos, sociais e de outras áreas de conhecimento, validando estratégias e resultados. (BNCC (2018))

Apesar da BNCC trazer uma formalidade ao início desse processo, é necessário que o mesmo conte com a logística de formação do professor e o entendimento sobre os meios

avaliativos (VALENTE (2016)).

Quanto a capacitação do docente, é necessário que seja feita de forma inicial (durante os cursos de licenciatura) e continuada, visando manter o profissional atualizado para as novidades e tendências a serem aplicadas e, nas quais, seus estudantes estão em constante contato.

Quanto aos meios de avaliação VALENTE (2016) cita a pesquisa realizada por Brennan e Resnick (2012). Nesse trabalho os autores desenvolvem três fases de avaliação, que diferem entre observação e montagem do portfólio.

Portanto, passamos a entender que o processo de desenvolvimento - significativo - do pensamento computacional, esta sendo promovido como parte das políticas educacionais de alguns países, bem como no Brasil por intermédio da BNCC. Apesar disso, necessitamos estruturar a camada dos docentes, para que estejam qualificados às novas demandas que aparecerão, garantindo qualidade e contextualidade ao processo de ensino e aprendizagem nacional.

## 5.3 Proposta de atividade envolvendo conceitos de teoria dos números

Com os conhecimentos confrontados e as ideias estabelecidas anteriormente, descreveremos uma possível atividade que pode ser aplicada para estudantes da educação básica que busquem, visando o conceito de pensamento computacional, introduzir os conceitos de programação básica, fazendo uso de conteúdos tradicionais da matemática, recorrentemente desenvolvidos pelos currículos educacionais pelo Brasil.

### 5.3.1 *Software*

Para a execução das linhas de programação faremos uso do “VisualG”. A escolha do programa em questão, vêm dos pontos destacados a seguir:

- i. Uso do Portugol <sup>1</sup>, auxiliando em uma familiaridade maior com a língua matriz dos estudantes (Português).
- ii. “*Layout*” simples e amigável para um primeiro contato com a ideia de programação.
- iii. Blocos lógicos com palavras indicadores de abertura e fechamento.

Entendemos que todas as condições acima garante uma melhor adaptação a maioria dos estudantes da educação básica que não possuem um contato tão direto com formas de linguagem de programação. Assim, a escolha deste programa vem nesse sentido. Segundo o site do programa

Foi propositadamente criado para ajudar aos alunos iniciantes em programação, mas pode ser utilizado por professores e outros autodidatas.  
(VISUALG)

---

<sup>1</sup>O VisuAlg é um programa que permite criar, editar, interpretar e que também executa os algoritmos em português (estruturado português) como se fosse um “programa” normal de computador. É um programa de livre uso e distribuição GRÁTIS, e DOMÍNIO PÚBLICO, usado para o ensino de lógica de programação em várias escolas e universidades no Brasil e no exterior.



### 5.3.2 Condições iniciais

Para que uma melhor aplicação dessa atividade, fica necessário garantir sentenças prévias em três partes o docente, o estudante e a instituição.

Em carácter do docente, assim como o apontado por VALENTE (2016), fica necessário uma formação inicial, bem como, a continuada. É de fundamental importância que tenha domínio sobre o *software* e as técnicas utilizadas, bem como, mantenha-se em estudos constantes, afim de suprir quaisquer dúvidas que venham a surgir por parte dos estudantes.

Para os docentes que não estão familiarizados com tal linguagem, recomenda-se um estudo prévio visando cobrir essas falhas.

Apesar disso, vale ressaltar, que essa atividade pode ser aplicação sem o uso necessário da tecnologia, apesar da ideia ser o uso de programação básica, não existe uma necessidade de transpô-la ao computador, podendo assim, ser desenvolvida por meio de fluxogramas que busquem trabalhar a resolução de devidos problemas<sup>2</sup>.

Quanto aos estudantes, para garantir a boa prática, é necessário que se mostre ativo, quanto a prática de pesquisa e estudos. Uma vez assim, o mesmo se manterá familiarizado ao *software* a ser utilizado, gerando facilidades progressivas para as atividades.

E, por último, em termos de infra-estrutura, é necessário que as instituições tenham ou passem a ter equipamentos compatíveis e habilitados, desenvolver um meio que incentive as condições de aprendizagem e garantir o bem estar do estudante, como do docente, de forma geral.

## 5.4 Desenvolvimento de algoritmos auxiliares

Esses algoritmos devem ser estruturados para que possam ser utilizados no produto final da atividade. Fica claro que o uso desse algoritmos, bem como, suas aplicações fica reservada ao docente que ministrará as aulas. Podendo este, desenvolve-los de forma separada ou de forma conjunta.

Cada linha de programação contará com:

- Um bloco de **referencial teórico**: representa os conceitos que o docente deverá explorar com os estudantes de forma prévia garantindo o máximo de compreensão.
- Os **passos lógicos** que descrevem a evolução do raciocínio para a construção das linhas de programação.
- **Fluxograma**, que funciona como material auxiliar que pode ser usado pelo docente como uma etapa introdutória na construção final da linha de programação.
- **Linha de programação**, com comentários, desenvolvidas no ambiente do VisualG.

### 5.4.1 Divisores de um $n$ natural

Para a construção das linhas de comando, necessitamos entender os passos lógicos e os embasamentos teóricos que guiam a construção do algoritmo.

---

<sup>2</sup>Tais fluxogramas encontram-se anexados a seguir, na subseção referente a cada algoritmo.

## REFERENCIAL TEÓRICO

**Definição 5.1.** Sejam  $n, d, k \in \mathbb{N}$ ,  $d$  é divisor de  $k$  se, e só se,  $n = d \cdot k$ .

**Proposição 5.2.** Seja  $n \in \mathbb{N}$  e  $\mathcal{D} = \{1, d_2, \dots, d_i, n\}$  conjunto de todos os divisores de  $n$ , tais que,  $d_s > d_{s-1}$ , para  $s = 1, 2, \dots, i$ , então  $d_i \leq \frac{n}{2}$ .

*Demonstração.* Tomemos  $n = d_i \cdot k$ , com  $k \geq 2$ . Suponhamos  $d_i > \frac{n}{2}$ , então  $d_i \cdot k > \frac{n}{2} \cdot k$ , implicando que  $n > n$ , o que é um absurdo. □

## PASSOS LÓGICOS

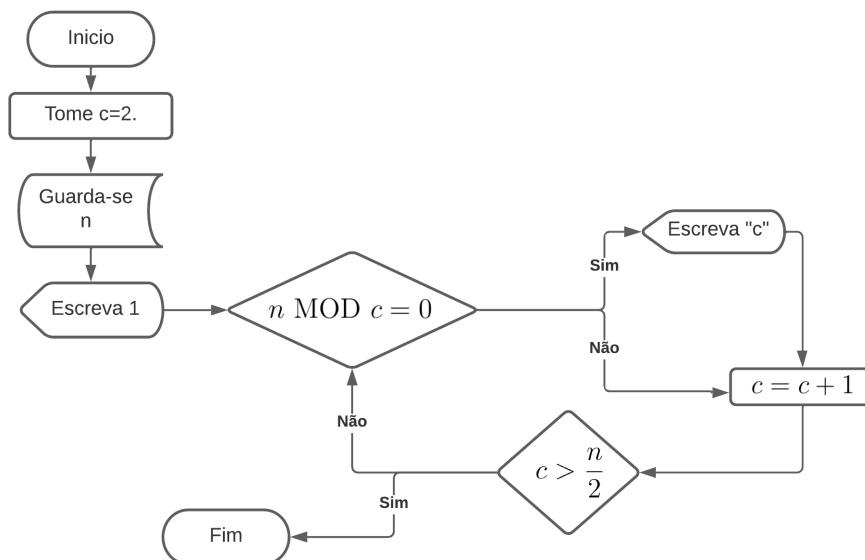
1º: Entender quem é o número a ser trabalhado.

2º: Testar desde o número 2 até a metade do número do 1º passo se algum é divisor.

3º: Registrar os que foram classificados como divisores no 2º passo.

Tomando o referencial e os passos lógicos podemos estruturar o fluxograma como segue.

Figura 5.1: Fluxograma dos divisores



Fonte: Autoria nossa

Por fim, propomos a construção do algoritmo conforme segue,

## LINHAS DE PROGRAMAÇÃO

```

Var
// ambiente variáveis.
n,c,j: inteiro
Inicio
// j recebe 1
j<-1
enquanto j = 1 faça
// Digita na tela os caracteres e pula linha.
    escreval("-----")
// Digita na tela os caracteres e pula linha.
    escreval("|          DIVISORES          |")
// Digita na tela os caracteres e pula linha.
    escreval("-----")
// Pede-se o número a ser trabalhado.
escreva("Digite o número desejado:")
// guarda a variável n.
    leia(n)
// digita na tela os caracteres e pula linha.
escreval("Os divisores de ",n,"são:")
// digita na tela o número 1 sem pular linha.
escreva("1,")
// Atribui a c valores de 2 até a metade de n.
para c de 2 ate n div 2 faça
    se (n MOD c = 0) entao
// Se c dividir n dita c na tela sem pular linha.
        escreva(c,")
fimse
fimpara
// Digita na tela a variável n e pula linha.
escreval(n, ".")
escreval("Digite 1 para recomeçar e 2 para finalizar o algoritmo:")
// Se j guarda 1, então recomeça o algoritmo, se não finaliza o algoritmo.
    leia(j)
fimenquanto
Fimalgoritmo

```

Tomando as linhas de programação vemos que os comandos são bem simples. Iniciamos definindo o número que estaremos analisando. Após isso, abrimos um ambiente “*para*”, que iniciará seus trabalhos com a variável *c* adquirindo valor 2 e finalizando quando adquirir a metade do valor analisado.

Esse *loop* serve para testarmos todos os possíveis candidatos a divisores. Quando a variável *c* recebe um dos possíveis candidatos este é testado dentro do ambiente “*se*”, que valida ele ser ou não um divisor.

Caso seja, o mesmo é escrito como uma das soluções e o algoritmo volta a funcionar para um novo valor de *c*. Isso se repetirá ate o ambiente “*para*” finalizar suas ações.

Note que o ambiente “*enquanto*” colocado no início do algoritmo garante a possibilidade de se realizar uma nova análise, digitando 1 para recomeçar ou 2 para finalizar o algoritmo.

### 5.4.2 Mínimo Divisor Comum de dois naturais

Para a construção das linhas de comando, necessitamos entender os passos lógicos e os embasamentos teóricos que guiam a construção do algoritmo.

#### REFERENCIAL TEÓRICO

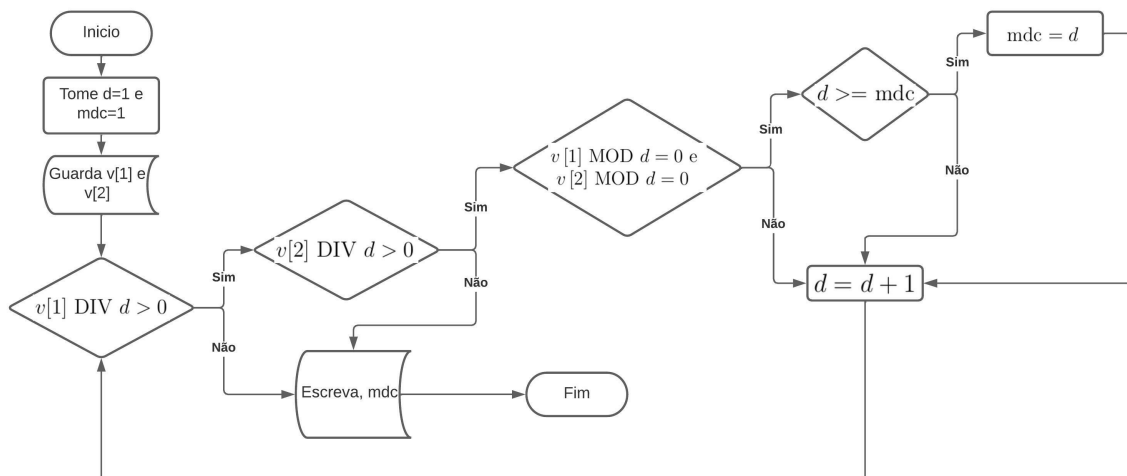
**Definição 5.3.** Sejam  $a, b, d \in \mathbb{N}$ . Se  $d$  é o maior divisor  $a$  e  $b$  ao mesmo tempo, então  $\text{mdc}(a, b) = d$ .

#### PASSOS LÓGICOS

- 1º: Entender quem são os números a serem trabalhados.
- 2º: Descobrir os números que são divisores de ambos.
- 3º: Dentre os números encontrados acima qual deles é o maior.
- 4º: Acusar o número encontrado no passo anterior como o mdc dos números selecionados.

Tomando o referencial e os passos lógicos podemos estruturar o fluxograma como segue.

Figura 5.2: Fluxograma do MDC



Fonte: Autoria nossa

Por fim, propomos a construção do algoritmo conforme segue,

### LINHAS DE PROGRAMAÇÃO

```

Var
v: vetor [1..2] de inteiro    // Variável vetor.
d,mdc,b,j, indice: inteiro    // Variável inteira.

Inicio
// j recebe 1
j<-1

enquanto j = 1 faça
// Digita na tela os caracteres sem pular linha.
    escreval("-----")
// Digita na tela os caracteres sem pular linha.
    escreval("|           MDC           |")
// Digita na tela os caracteres sem pular linha.
    escreval("-----")
// mdc recebe 1
mdc<-1
// Atribui valor 1 e 2 para variável indice.
para indice de 1 ate 2 faça
    escreval("Digite o ", indice, "º número:")
// Guarda a variável v [indice].
    leia(v[indice])
fimpara
// d recebe 1
d<-1
// Enquanto o quociente da divisão de v [1] por d e v [2] por d for maior que 0.
enquanto (v[1] div d > 0) e (v[2] div d > 0) faça
    se ((v[1] MOD d = 0) e (v[2] MOD d = 0)) entao
        se d>=mdc entao
// Se d é divisor de v [1] e v [2] e maior ou igual que mdc, então mdc atribui d.
            mdc<-d
        fimse
    fimse
// d recebe o valor de seu sucessor.
d<-d + 1
// Ao fechar o ambiente enquanto teremos o maior divisor comum de v [1] e v [2].
fimenquanto

```

```
// Digita na tela os caracteres e pula linha.
escreval("mdc(",v[1],"",v[2],") = ",mdc,".")

// Se j guarda 1, então recomeça o algoritmo, se não finaliza o algoritmo.
escreval("Digite 1 para recomeçar e 2 para finalizar o algoritmo:")
    leia(j)
fimenquanto
Fimalgoritmo
```

Tomando as Linhas de Programação acima, vemos algumas estruturas que caracterizam os quatro passos abordados antes da apresentação das linhas propriamente.

O algoritmo inicia com uma variável  $j$  inteira seguida por um ambiente “*enquanto*”, este funciona como um loop, onde o usuário pode escolher entre fechar o programa ou reaplicá-lo.

Logo após a apresentação o usuário deve escolher dois números, que serão os avaliados. Depois disso, temos um ambiente “*enquanto*”, este serve para efetivar um loop dos divisores que funciona até que encontremos um valor  $d$ , tal que, a divisão de  $v$  [índice], com índice= 1, 2, por  $d$  seja inferior a 0. Enquanto isso ocorrer, os valores de  $d$  são testados no ambiente “*se*”.

Assim, até que a condição dos ambientes “*se*” e “*enquanto*” não seja quebrada, o algoritmo continua rodando, até que todos os pontos sejam satisfeitos.

### 5.4.3 Mínimo Múltiplo Comum de dois naturais

Tomando a sequência abordada nas seções anteriores, focaremos em explicitar como se dá a construção das linhas de programação voltadas a resolução do MMC entre dois números.

#### REFERENCIAL TEÓRICO

**Definição 5.4.** Sejam  $a, b, m \in \mathbb{N}$ . Se  $m$  é o menor múltiplo de  $a$  e  $b$ , ao mesmo tempo, então  $\text{mmc}(a, b) = m$ .

**Proposição 5.5.** Seja  $a, b \in \mathbb{N}$ , então  $\text{mdc}(a, b) \cdot \text{mmc}(a, b) = ab$ .

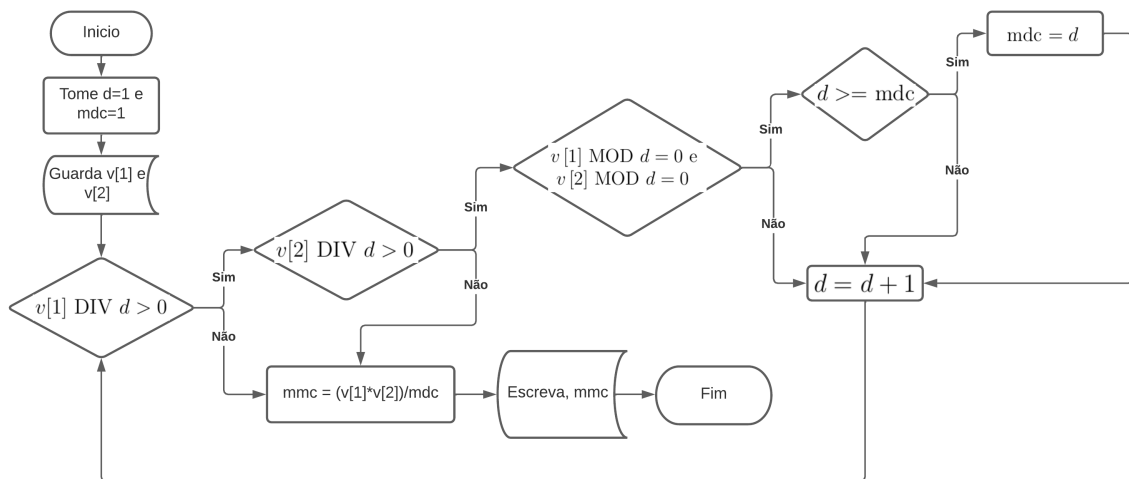
Note que tendo a construção do MDC podemos utilizar as mesmas linhas de programação porém com uma finalização visando a divisão entre o produto dos números apontados no início do algoritmo pelo MDC deles.

### PASSOS LÓGICOS

- 1º: Entender quem são os números a serem trabalhados.
- 2º: Efetuar as linhas do MDC.
- 3º: Efetuar o produto dos números apontados.
- 4º: Acusar o MMC como a divisão entre o valor do 3º passo e 2º passo.

Tomando o referencial e os passos lógicos podemos estruturar o fluxograma como segue.

Figura 5.3: Fluxograma do MMC



Fonte: Autoria nossa

Por fim, propomos a construção do algoritmo conforme segue,

### LINHAS DE PROGRAMAÇÃO

```

Var
v: vetor [1..2] de inteiro // Variável vetor.
mdc,d,j,l,indice: inteiro // Variável inteira.

Início
// j recebe 1
j<-1
  
```

```

enquanto j = 1 faça
// Digita na tela os caracteres e pula linha.
    escreval(" - - - - - ")
// Digita na tela os caracteres e pula linha.
    escreval("|          MMC          |")
// Digita na tela os caracteres e pula linha.
    escreval(" - - - - - ")
// mdc recebe 1
mdc<-1
// Atribui valor 1 e 2 para variável indice.
para indice de 1 ate 2 faça
    escreval("Digite o ", indice, "º número:")
// Guarda a variável v[indice].
    leia(v[indice])
fimpara
// d recebe 1
d<-1

// Enquanto o quociente da divisão de v[1] por d e v[2] por d for maior que 0.
enquanto (v[1] div d > 0) e (v[2] div d > 0) faça
    se ((v[1] MOD d = 0) e (v[2] MOD d = 0)) entao
        se d>=mdc entao
// Se d é divisor de v[1] e v[2] e maior ou igual que mdc, então mdc atribui d.
            mdc<-d
        fimse
    fimse
// d recebe o valor de seu sucessor.
d<-d + 1
// Ao fechar o ambiente enquanto teremos o maior divisor comum de v[1] e v[2].
fimenquanto
// A variável l recebe o valor do produto entre v[1] e v[2].
l<- v[1]*v[2]
// Digita na tela o valor do mmc calculado.
escreval("mmc( ",v[1]," ", v[2],") = ", l div mdc)
escreval("Digite 1 para recomeçar e 2 para finalizar o algoritmo:")
// Se j guarda 1, então recomeça o algoritmo, se não finaliza o algoritmo.
    leia(j)
fimenquanto
Fimalgoritmo

```

Do mesmo jeito que apontando nas Linhas de Programação do MDC note que o primeiro ambiente “*enquanto*” garante que o programa será executado quantas vezes for conveniente ao usuário.

Logo após a abertura do algoritmo, temos o entendimento dos números que serão analisados notados por  $v[1]$  e  $v[2]$ . Após isso, temos o ambiente “*enquanto*” que encontra o valor do MDC dos números analisados.

Por fim, tomando o Referencial teórico que concluímos, com simples operações o valor



do MMC.

#### 5.4.4 Classificando os naturais entre primos e compostos

Temos como objeto descrever a construção, desde os passos lógicos até as efetivas linhas de programação, de um algoritmo responsável pela discriminação de números naturais, indicando ser primo ou composto.

Para a construção do algoritmo tomemos a proposição abaixo, bem como a observação na sequência, que sintetizam a ideia dos passos lógicos usados nas linhas de programação.

##### REFERENCIAL TEÓRICO

**Proposição 5.6.** *Sejam  $a \in \mathbb{N}$  e  $\mathcal{D} = \{1, d_2, \dots, d_n\}$  o conjunto de todos os divisores de  $a$ , tal que,  $d_2 > 1$  e  $d_j > d_{j-1}$ , com  $j = 3, 4, \dots, n$ . Se  $d_2 = a$ , então  $a$  é primo.*

*Demonstração.* Se  $d_2 = n$ , então este é o último divisor de  $n$ . Portanto  $n$  tem apenas dois divisores, logo é primo. □

**Observação 5.7.** Tomando as condições da Proposição 5.6, se  $d_2 \neq a$ , então  $a$  é composto.

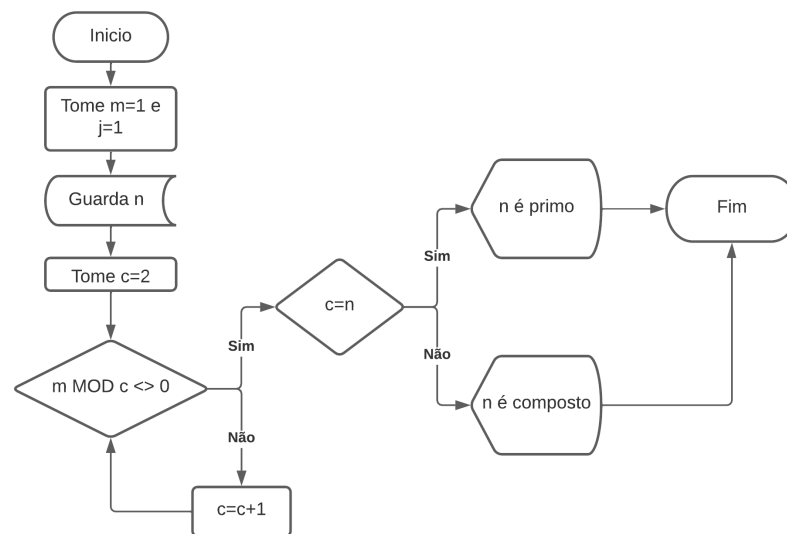
Note que as consequências do exposto acima descrevem uma maneira de testarmos se um número é primo ou composto. Vale ressaltar que existem mais formas de se pensar em primos, mas para o que buscamos nesse momento essa será suficiente.

##### PASSOS LÓGICOS

- 1º: Entender quem é o número a ser trabalhado.
- 2º: Sabendo que em ordem crescente o 1 é o primeiro divisor, descobrir quem é o próximo divisor.
- 3º: Ver se o número obtido no 2º passo é igual ou diferente do 1º passo.
- 4º: Se igual então o número é primo e se é diferente é composto.

Tomando o referencial e os passos lógicos podemos estruturar o fluxograma como segue.

Figura 5.4: Fluxograma da classificação entre primos e compostos



Fonte: Autoria nossa

Por fim, propomos a construção do algoritmo conforme segue,

### LINHAS DE PROGRAMAÇÃO

```

Var
n,c,j: inteiro    // Variável inteira.

Início
// j recebe 1
j<-1
enquanto j=1 faça
// Digita na tela os caracteres e pula linha.
escreval(-----")
// Digita na tela os caracteres e pula linha.
escreval("|      É PRIMO OU COMPOSTO?      |")
// Digita na tela os caracteres e pula linha.
escreval(-----")
// Digita na tela os caracteres sem pular linha.
escreva("Digite o número:")
// Guarda a variável n.
leia(n)
// c atribui 2.
c<-2
enquanto (n MOD c <> 0) faça
// Enquanto c não for divisor de n, c recebe o valor de seu sucessor.
    c<-c+1
fimenquanto
  
```

```

    se c=n entao
// Se c é igual a n, então digita “é primo”.
        escreval(n,"é primo.")
    senao
// Se c é diferente de n, então digita “é composto”.
        escreval(n,"é composto.")
    fimse
escreval("Digite 1 para recomeçar e 2 para finalizar o algoritmo:")
// Se j guarda 1, então recomeça o algoritmo, se não finaliza o algoritmo.
leia(j)
fimenquanto
Fimalgoritmo

```

Tomando as linhas de programação, notamos uma sentença que se repete desde o primeiro algoritmo desenvolvido, onde o ambiente “*enquanto*” funciona para reiniciar o algoritmo após a verificação inicial dada.

A partir desse momento, todas as linhas de programação seguem a Proposição 5.6 e a Observação 5.7. Para iniciar apresentamos a primeira variável  $c$ , na qual atribuímos o valor inicial de 2. Ela entra em um *loop*.

Este ambiente é repetido até que encontremos um valor de  $c$  que divida o número acusado no início. Seguimos para um ambiente “*se*”, que avalia o divisor  $c$  encontrado sendo que se este for igual ao número iniciado temos um valor primo do contrário este é composto.

### 5.4.5 Teorema Fundamental da Aritmética

Temos como objetivo apresentar o algoritmo que fornece a decomposição em fatores primos expressas pelo TFA (Teorema Fundamental da Aritmética), muito comumente utilizado e desenvolvido em sala de aula, por professores e estudantes.

Este, além de ser amplamente citado ao decorrer do trabalho, também é de extrema importância para o decorrer do processo de ensino.

#### REFERENCIAL TEÓRICO

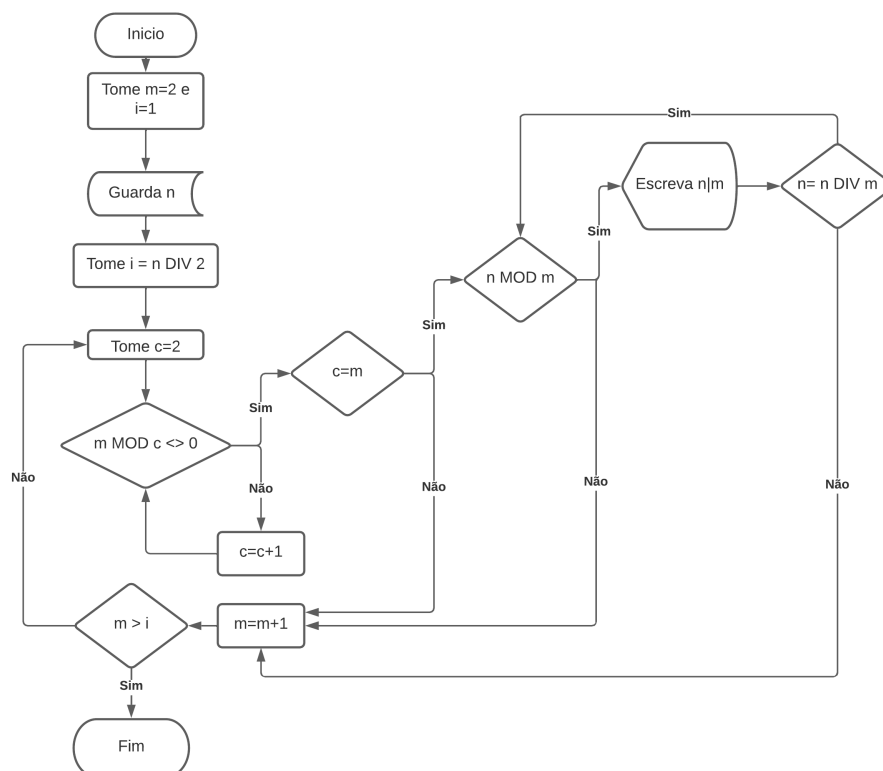
**Teorema 5.8.** (Teorema Fundamental da Aritmética - TFA) *Todo número natural  $n \geq 2$  possui uma decomposição única por fatores primos, isto é, dado  $n \geq 2$  existem únicos primos distintos  $p_1, p_2, \dots, p_r$  e naturais  $\alpha_1, \alpha_2, \dots, \alpha_r$  tais que  $n = p_1^{\alpha_1} p_2^{\alpha_2} \dots p_r^{\alpha_r}$ .*

### PASSOS LÓGICOS

- 1º: Entender quem é o número a ser trabalhado.
- 2º: Entender quem são os possíveis divisores primos.
- 3º: Ver se o número obtido no 2º passo divide o número do 1º passo. Se este não dividir testa-se outro número.
- 4º: Se o número do 1º passo for divisível pelo primo divisor encontrado no 2º passo, repetimos o algoritmo para o quociente da divisão.

Tomando o referencial e os passos lógicos podemos estruturar o fluxograma como segue.

Figura 5.5: Fluxograma do TFA



Fonte: Autoria nossa

Por fim, propomos a construção do algoritmo conforme segue,

## LINHAS DE PROGRAMAÇÃO

```

Var
n,c,m,j: inteiro    // Variável inteira.

Inicio
// j recebe 1
j<-1
enquanto j = 1 faça
// Digita na tela os caracteres e pula linha.
    escreval("-----")
// Digita na tela os caracteres e pula linha.
    escreval("|          TFA          |")
// Digita na tela os caracteres e pula linha.
    escreval("-----")
// Digita na tela os caracteres sem pular linha.
escreva("Digite o número a ser fatorado:")
// Guarda a variável n.
    leia(n)
para m de 2 ate n div 2 faça
// c recebe 2
    c<-2
    enquanto (m MOD c <> 0) faça
// Enquanto c não for divisor de m, c recebe o valor de seu sucessor.
        c<-c + 1
    fimenquanto
    se c = m entao
// Repita os comandos abaixo até m não for divisor de n.
        repita
            se (n MOD m = 0) entao
// Se m divide n, digite na tela os caracteres.
                escreval(n,"| ",m)
// n atribui o quociente da divisão de n por m.
                n<-n div m
            fimse
        ate n MOD m <> 0
    fimse
fimpara
escreval("_|")
escreval(1)
escreval("Digite 1 para recomeçar e 2 para finalizar o algoritmo:")
// Se j guarda 1, então recomeça o algoritmo, se não finaliza o algoritmo.
    leia(j)
    fimenquanto
Fimalgoritmo

```

Iniciamos o algoritmo com o ambiente “*enquanto*” para efeitos de reiniciar-lo, caso haja necessidade de avaliarmos outro número. Seguimos com o ambiente “*para*” e o ambiente “*se*” que avaliam quais número são primos. Após isso, entramos no ambiente “*se*” que avalia se o primo é divisor do número atribuído inicialmente. Se este for divisível ele fica registrado como divisor do número.

Após isso, o algoritmo reinicia tomando o quociente da divisão efetuada anteriormente. Isso se repete até o momento em que o primo obtido for igual ao quociente da última divisão.

## 5.5 Atividade final

Com os algoritmos desenvolvidos anteriormente, construiremos uma atividade cujo o produto final seja a criação de linhas de programação que disponham de todas as características dos diferentes números naturais.

---

### ATIVIDADE: "RG DOS NÚMEROS"

---

**Título do projeto:** RG dos números naturais.

**Publico Alvo:** Alunos da educação básica.

**Tempo:** Depende da turma e suas experiências anteriores com linhas de programação. Apesar disso, damos uma projeção mínima de 1 ano.

**Avaliação:** A avaliação partirá de duas frentes:

- Análise do portfólio;
- Análise das linhas de programação desenvolvidas; e,
- Envolvimento nas atividades.

**Justificativa:** Tomando o que foi discutido, anteriormente, entendemos que a atividade segue em proporcionar um desenvolvimento de conceitos matemáticos, por meio da integração entre tecnologia digital e pensamento computacional. Uma vez que o ensino da matemática encontra-se em estágio de mudanças e com índices bem abaixo do médio (segundo avaliações governamentais), essa atividade vem para resgatar e ensinar conceitos fundamentais da matemática, visando um bom seguimento no curso da educação básica. Capacitando o estudante há evoluir seu potencial de abstração para qualquer problemática.

**Objetivos:**

- **Geral:** Realizar com o estudante a criação de linhas de programações capazes de fornecer características dos números escolhidos como, por exemplo, se é primo ou composto, quais são seus divisores, condições especiais, escrita por extenso e outras mais.

**– Específicos:**

- i. Entendimento de MDC e MMC.
- ii. Conceito de divisibilidade.
- iii. Classificação dos Números Naturais.
- iv. Posição relativa a reta numérica.
- v. Conhecimento básico em programação<sup>3</sup>.

**Recurso:**

- i. Computadores com o VisualG instalado.
- ii. Lousa.
- iii. Datashow<sup>4</sup>.

**Habilidade:**

- i. (EF06MA04) Construir algoritmo em linguagem natural e representá-lo por fluxograma que indique a resolução de um problema simples (por exemplo, se um número natural qualquer é par).
- ii. (EF06MA05) Classificar números naturais em primos e compostos, estabelecer relações entre números, expressas pelos termos "é múltiplo de", "é divisor de", "é fator de", e estabelecer, por meio de investigações, critérios de divisibilidade por 2, 3, 4, 5, 6, 8, 9, 10, 100 e 1000.
- iii. (EF06MA06) Resolver e elaborar problemas que envolvam as ideias de múltiplo e de divisor.
- iv. (EM13MAT405) Utilizar conceitos iniciais de uma linguagem de programação na implementação de algoritmos escritos em linguagem corrente e/ou matemática.

**Metodologia:**

ETAPA	DESCRIÇÃO	TEMPO
1ª Etapa	Apresentação do Projeto e seu seguimento	2 aula
2ª Etapa	Desenvolvimento das ferramentas básicas relacionadas ao programa VisualG. Aconselha-se o uso de exercícios básicos que desenvolvam essas ferramentas	1 mês
3ª Etapa	Explicação teórica e aplicada sobre o conceito dos divisores de um dado número $n$ . Focando em exemplos que levem a discussão na quantidade dos divisores de cada número.	0,5 mês

<sup>3</sup>Aconselha-se que o professor explique conceitos básicos de programação e teste alguns exemplos simples antes de abordar a atividade em sua completude.

<sup>4</sup>Esse item é opcional, mas entendeu-se que com ele o professor possa auxiliar de forma geral a sala, fazendo com que todos caminhem juntos e suas dúvidas sejam devidamente sanadas.

4 <sup>a</sup> Etapa	Desenvolvimento do fluxograma, por parte dos estudantes, capazes de descrever as linhas de programação a serem construídas.	0,5 mês
5 <sup>a</sup> Etapa	Desenvolvimento da linha de programação responsável por encontrar os devidos divisores.	1 mês
6 <sup>a</sup> Etapa	Construção teórica e prática relacionada a temática de MMC e MDC.	0,5 mês
7 <sup>a</sup> Etapa	Desenvolvimento do fluxograma, por parte dos estudantes, capazes de descrever as linhas de programação a serem construídas.	0,5 mês
8 <sup>a</sup> Etapa	Desenvolvimento da linha de programação responsável por encontrar os devidos o MMC e MDC de dois diferentes valores naturais.	1 mês
9 <sup>a</sup> Etapa	Evidenciar a classificação entre primos e compostos que existe para qualquer número natural. Formalizar a existência do TFA.	1 mês
10 <sup>a</sup> Etapa	Desenvolvimento do fluxograma, por parte dos estudantes, capazes de descrever as linhas de programação a serem construídas.	0,5 mês
11 <sup>a</sup> Etapa	Desenvolvimento da linha de programação responsável por classificar números naturais maiores que 1 em primos ou compostos.	1 mês
12 <sup>a</sup> Etapa	Explicação do produto final.	2 aulas
13 <sup>a</sup> Etapa	Agrupamento das linhas de programação construídas anteriormente para geração do produto final.	1 mês
14 <sup>a</sup> Etapa	Discussão de fechamento	4 aulas

### 5.5.1 1<sup>a</sup> etapa

Nesse momento o docente deve expor para os estudantes como a atividade funcionará. Será importante que o mesmo descreva com detalhe o tempo de cada tarefa, bem como, o que será produzido em cada uma delas. Isso, mostrará ao mesmo [estudante] que o conhecimento será desenvolvido com o passar do tempo, de forma que todos, mesmos aqueles sem prática ou contato com programação, possam seguir e desenvolver a atividade.

### 5.5.2 2<sup>a</sup> etapa

Durante o período programado, é fundamental que a primeira ação do docente seja inserir o estudante frente ao que seria essa linguagem de programação e para o que é usada.

Evidenciar partes do cotidiano e propor discussões acessíveis desencadeará um processo de interesse nos estudantes. Vale ressaltar que o docente deve fornecer acesso a literaturas exteriores, de forma, a dar suporte aos estudantes que queiram desenvolver essa prática fora do ambiente escolar.



Visando essa ideia pode ser aconselhado como literatura externa

- [http://www.inf.ufsc.br/~bosco.sobral/ensino/ine5201/Visualg2\\_manual.pdf](http://www.inf.ufsc.br/~bosco.sobral/ensino/ine5201/Visualg2_manual.pdf)
- [https://nte.ufsm.br/images/identidade\\_visual/introducaoalgoritmos.pdf](https://nte.ufsm.br/images/identidade_visual/introducaoalgoritmos.pdf)

Além do mais, o docente deve exercitar a ideia de variável, função, operadores e comandos de repetição, dando ênfase aos que serão utilizados na construção das linhas de programação.

### Exercícios: exemplos de abordagens

Abaixo seguem os exercícios e suas soluções que estarão presentes na 3ª etapa da atividade.

---

#### Construa linhas de comando que operem dois números com operações diversas.

---

#### Proposta de resolução:

---

```

Var
n,m: real    // Variável inteira.
o: caracter  // Variável carcter.
Inicio
escreval("Digite o 1º número:") // Digita na tela os caracteres e pula linha.

    leia(n)    // Guarda a variável n.
escreval("Digite o 2º número:") // Digita na tela os caracteres e pula linha.
    leia(m)    // Guarda a variável m.
escreval("Digite a operação desejada, sendo:") //Pede a operação desejada.
    escreval("+: soma;") // Lista a operação de soma.
    escreval("-: subtração;") // Lista a operação de subtração.
    escreval("/: divisão;") // Lista a operação de divisão.
    escreval("*: multiplicação;") // Lista a operação de multiplicação.
    leia(o)    // Guarda a operação desejada.
escolha o
    caso "+" // Caso a operação seja soma.
        escreval(n,"+",m,"=",n + m) // Armasse a operação e digita-se o resul-
tado.
    caso "-" // Caso a operação seja subtração.
        escreval(n,"-",m,"=",n - m) // Armasse a operação e digita-se o resul-
tado.
    caso "*" // Caso a operação seja multiplicação.
        escreval(n,"*",m,"=",n * m) // Armasse a operação e digita-se o resul-
tado.
    caso "/" // Caso a operação seja divisão.

```

```

        escreval(n,"/",m,"=",n/m)    // Armasse a operação e digita-se o resultado.
fimescolha
Fimalgoritmo

```

---

**Descreva a criação de linhas de programação que descrevam uma dada tabuada.**

**Proposta de resolução:**

```

Var
n,l: real    // Variável real.
Inicio
    escreval("Digite o valor da tabuada à ser calculada:")    // Pede-se o valor da tabuada
de multiplicação a ser calculada.
        leia(n)    // Guarda a variável n.
    l<-1    // l recebe 1.
    escreval(n,"x0 = 0")    // Escreve a notação de tabuada e o resultado para o caso em
que a multiplicação é feita por 0.
    enquanto l<11 faça
        escreval(n,"x",l,"=",l*n)    // Enquanto l for menor que 11, escreve-se a notação
de multiplicação e seus respectivos resultados.
        l<-l+1    // l recebe o valor de seu sucessor.
    fimenquanto
Fimalgoritmo

```

---

**Construa linhas de comando que calculem a média de um estudante indicando se o mesmo está aprovado ou não, tomando que o peso da primeira avaliação é 3 e as demais peso 1, existem 4 avaliações e a média para aprovação é 6 ou superior.**

**Proposta de resolução:**

```

Var
a,b,c,d,m,s: real    // Variável real.
Inicio
    escreval("Digite sua primeira nota:")    // Pede-se a primeira nota.

```

```

    leia(a)    // Guarda-se o valor da primeira nota.
    escreval("Digite sua segunda nota:") // Pedese a segunda nota.
    leia(b)    // Guarda-se o valor da segunda nota.
    escreval("Digite sua terceira nota:") // Pedese a terceira nota.
    leia(c)    // Guarda-se o valor da terceira nota.
    escreval("Digite sua quarta nota:") // Pedese a quarta nota.
    leia(d)    // Guarda-se o valor da quarta nota.
    m<-3 * b   // m recebe o triplo do valor de b.
    s<- a + m + c + d // s recebe o valor da soma das notas.
    escreval("Sua média é", s/6) // A média ponderada é calculada e digitada.
    se s/6>=6 entao
        escreval("Portanto, está aprovado.") // Se a média é igual ou superior a 6
        digita-se aprovado.
    senao
        escreval("Portanto, está reprovado.") // Se a média é inferior a 6
        digita-se reprovado.
    fimse
Fimalgoritmo

```

---

**Construa linhas de comando que indique qual a relação de maior ou menor entre dois números.**

---

### Proposta de resolução:

```

Var
v: vetor [1..2] de real // Variável vetor.
i: inteiro // Variável inteiro.
Inicio
i<-1 // i recebe 1.
repita
    escreval(i,"º número:")
    leia(v[i]) // Defini-se os dois valores (v [1] e v [2]) a serem ordenados.
    i<-i+1
ate i = 3
se v[1]>v[2] entao
    escreval(v[1],">",v[2]) // Digita-se caso v [1] > v [2].
senao
    escreval(v[1],"<",v[2]) // Digita-se caso v [1] < v [2].
fimse
Fimalgoritmo

```

---

---

**Em um mercado o quilo da batata custa 4,00, a alface  
custa 1,60 a unidade e o quilo da carne 30,00.  
Construa uma linha de programação que calcule o  
quanto seria gasto em uma compra desses itens em  
quantidades variadas.**

---

**Proposta de resolução:**

---

```

Var
b,c,r,s,t: real    // Variável real.
a: inteiro        // Variável inteiro.
Inicio
  escreval("Quantos quilos de batatas foram comprados?") // Pede-se a quantidade
a ser comprada.
  leia(b)          // Guarda-se a quantidade de Kg de batatas.
  escreval("Quantos maços de alface foram comprados?") // Pede-se a quantidade a
ser comprada.
  leia(a)          // Guarda-se a quantidade maços de alface.
  escreval("Quantos quilos de carne foram comprados?") // Pede-se a quantidade a
ser comprada.
  leia(c)          // Guarda-se a quantidade de Kg de carne.
  r<-4 * b        // Custo da batata.
  s<-a * 1.6      // Custo da alface.
  t<-c * 30       // Custo da carne.
  escreval("Será gasto",r + s + t,"no mercado.") // Escreve-se o valor total da compra.
Fimalgoritmo

```

---



---

**Construa as linhas de programação capazes de  
classificar em par ou impar qualquer número natural.**

---

**Proposta de resolução:**

---

```

Var
n: inteiro        // Variável inteira.
Inicio
  escreval("Digite o valor a ser analisado:") // Pede-se o número a ser analisado.
  leia(n)          // Guarda-se o valor de n.
se n MOD 2 = 0 entao
  escreval(n, "é par.") // Se 2 divide n, então digita "é par".
senao

```

---

```

    escreval(n, "é impar.") // Se 2 não divide n, então digita "é impar".
fimse
Fimalgoritmo

```

---

### 5.5.3 3<sup>a</sup>, 4<sup>a</sup> e 5<sup>a</sup> etapas

Nessas etapas o docente tem que tanto atacar os conceitos teóricos que envolvem a ideia de divisores como algumas aplicações do cotidiano do estudante que envolvam esse conceito.

É fundamental decorrer da relação entre teoria e prática, pois dessa forma se cria um elo de significância maior, estabelecida entre o estudante e o conteúdo a ser desenvolvido.

Logo após o docente deve discutir com os alunos características dos divisores. Para isso, aconselha-se estimular que os estudantes busquem e deixem registrados os divisores de alguns números da escola do professor.

Depois disso, cabe ao professor chamar atenção para algumas propriedade e características de evidência como, por exemplo,

- O fato de alguns números só possuírem dois divisores distintos; e,
- Números escritos na potência dois terem certo padrão na quantidade de divisores.

Feito estas análises o professor deve pedir aos estudantes que desenvolvam um fluxograma que descreva o processo de obtenção dos divisores de um número natural qualquer.

Como esse fluxograma, o aluno deve ser questionado sobre quais funções, operadores e comandos de repetição deve dispor para trazer o produto desenvolvido na 4<sup>a</sup> etapa para a linguagem de programação do VisualG.

### 5.5.4 6<sup>a</sup>, 7<sup>a</sup> e 8<sup>a</sup> etapas

As etapas que seguem tem desenvolvimento muito semelhante ao empregado na Subseção 5.5.3.

Como temas de discussão, o docente pode lançar temas como os que decorrem abaixo para aguçar o estudante e dar uma dimensão de como as ferramentas de MMC e MDC podem e são utilizadas em variadas tarefas.

Assim, aconselhamos discussões como, por exemplo,

- **Movimento de astros:** Os astros como, por exemplo, os que compõe o Sistema Solar, se alinham em determinado ponto sobre a descrição de períodos específicos. A fim de descobrir tais momentos podemos nos valer dos conceitos relativos a MMC.
- **Otimização:** muitos exemplos de otimização como, por exemplo, o corte de tecidos ou peças em partes iguais sobre as maiores dimensões, são solucionados com técnicas simples envolvendo MDC.
- **Sistemas de criptografias:** sistemas de que exigem alto nível de segurança exigem senhas e padrões de difícil acesso, a não ser, que tenhamos a chave do código. Devido as dificuldades de se obter e encontrar os números primos, estes se mostram bem interessantes para compor tais sistemas de segurança.

### 5.5.5 9<sup>a</sup>, 10<sup>a</sup> e 11<sup>a</sup> etapas

Para desenvolvimento do Teorema Fundamental da Aritmética, aconselhamos um procedimento análogo ao até agora empregado nas Subseções 5.5.3 e 5.5.4.

Chamamos a atenção para as discussões teóricas envolvendo esse teorema. Vale ressaltar ao estudante os casos em que  $n = 0$  e  $n = 1$ , sendo  $n \in \mathbb{N}$  o número decomposto pelo TFA.

Essas discussões auxiliaram no desenvolvimento, tanto do fluxograma como da própria linguagem de programação.

### 5.5.6 12<sup>a</sup> e 13<sup>a</sup> etapas

Nessa parte final, o docente deverá auxiliar os estudantes a agrupar as linhas de programação construídas no decorrer do trabalho. A seguir deixamos uma possível forma de agrega-las, tomando as linhas de programação sugeridas anteriormente.

#### LINHAS DE PROGRAMAÇÃO

```

Var
n,j,c,m,a,s,mdc,mocr,d,b,t,r,q: inteiro
Início
j<-1
enquanto j = 1 faça
    escreval("Digite o número que deseja conhecer:")
        leia(n)
    escreval("-----")
    escreval("                RG DOS NÚMEROS                ")
    escreval("-----")

    escreval("Nome:", n)

se n=1 entao
    escreval("( ) Par (X) Impar")
    escreval("Sucessor: 2")
    escreval("Antecessor: 0")
    escreval("mdc(1,2) = 1.")
    escreval("Decomposição: Não existe decomposição.")
    escreval("1: recomeçar/2: finalizar")
        leia(j)
fimse

se n=0 entao
    escreval("(X) Par ( ) Impar")
    escreval("Sucessor: 1")
    escreval("Antecessor: -")
    escreval("Decomposição: Não existe decomposição.")
    escreval("1: recomeçar/2: finalizar")

```

```
        leia(j)
fimse

se n >= 2 entao
    c <- -2
    enquanto (n MOD c <> 0) faca
        c <- -c + 1
    fimenquanto
    se c = n entao
        escreval("(X) Primo ( ) Composto")
    senao
        escreval("( ) Primo (X) Composto")
    fimse

se n MOD 2 = 0 entao
    escreval("(X) Par ( ) Impar")
senao
    escreval("( ) Par (X) Impar")
fimse

se n = 2 entao
    escreval("2 é o único par primo existente.")
fimse
escreval("Antecessor:", n-1)
escreval("Sucessor:", n+1)
escreva("Divisores: 1,")
    para m de 2 ate n div 2 faca
        se (n MOD m = 0) entao
            escreva(m, ",")
        fimse
    fimpara
escreval(n, ".")

a <- n-1
s <- n+1
mdc <- 1
d <- 1

enquanto (a div d > 0) e (n div d > 0) faca
    se ((a MOD d = 0) e (n MOD d = 0)) entao
        se d >= mdc entao
            mdc <- d
        fimse
    fimse
d <- -d + 1
fimenquanto
```

```
escreval("mdc(",a,",",n,") = ",mdc, ".")
b<-1
mdcr<-1

enquanto (s div b>0) e (n div b>0) faca
    se ((s MOD b=0) e (n MOD b=0)) entao
        se b>=mdc entao
            mdc<-b
        fimse
    fimse
b<-b+1
fimenquanto

escreval("mdc(",s,",",n,") = ",mdcr, ".")
escreva("Decomposição:")
q<-2
    enquanto (n MOD q <> 0) faca
        q<-q+1
    fimenquanto
se q=n entao
    escreval("Sem decomposição")
senao
    para t de 2 ate n div 2 faca
        r<-2
        enquanto (t MOD r <> 0) faca
            r<-r+1
        fimenquanto

se r=t entao
    repita
        se (n MOD t = 0) entao
            escreva(t, ".")
            n<-n div t
        fimse
    ate n MOD t <> 0
fimse
    fimpara
    fimse
escreval()

escreval("1: recomeçar/2: finalizar")
leia(j)

fimse
fimenquanto
Fimalgoritmo
```



# A Função Gama

Iniciamos tomando a definição dessa função .

**Definição A.1.** A função  $\Gamma : (0, \infty) \rightarrow \mathbb{R}$  é definida para cada  $x$ , por  $\Gamma(x) = \int_0^\infty t^{x-1} e^{-t} dt$ .

A fim de estabelecer demonstrações plausíveis e completas tomemos dois resultados importantes.

**Propriedade A.2.** A função  $\Gamma$  é bem definida.

*Demonstração.* Tomando  $f(x) = t^{x-1} e^{-t}$ , temos que ela é contínua e o Teorema de Weierstrass garante que é limitada no intervalo  $[0, L]$ , portanto  $\int_L^\infty t^{x-1} e^{-t} dt$  existe.

Abordando  $\int_L^\infty t^{x-1} e^{-t} dt$ .

$$\begin{aligned}
 e^{\frac{t}{2}} &> t^{x-1} \Rightarrow \\
 e^{-\frac{t}{2}} &> e^{-t} t^{x-1} \geq 0 \Rightarrow \\
 \int_L^\infty e^{-\frac{t}{2}} dt &> \int_L^\infty e^{-t} t^{x-1} dt \geq 0 \Rightarrow \\
 \lim_{b \rightarrow \infty} \left( \int_L^b e^{-\frac{t}{2}} dt \right) &> \int_L^\infty e^{-t} t^{x-1} dt \geq 0 \Rightarrow \\
 -2 \lim_{b \rightarrow \infty} \left( e^{-\frac{b}{2}} - e^{-\frac{L}{2}} \right) &> \int_1^\infty e^{-t} t^{x-1} dt \geq 0 \Rightarrow \\
 2e^{-\frac{L}{2}} &> \int_1^\infty e^{-t} t^{x-1} dt \geq 0.
 \end{aligned}$$

Como, para  $x > 0$ , temos  $\int_0^L t^{x-1} e^{-t} dt$  e  $\int_L^\infty t^{x-1} e^{-t} dt$  convergem,  $\Gamma(x)$  também converge. □

O Lema a seguir será usado posteriormente, nas propriedades de  $\Gamma$ .

**Lema A.3.**  $\int_{-\infty}^\infty e^{-x^2} dx = \sqrt{\pi}$ .

*Demonstração.* Tomando  $A = \int_{-\infty}^\infty e^{-x^2} dx = \int_{-\infty}^\infty e^{-y^2} dy$  e que  $A$  converge. Assim,

$$\begin{aligned}
 A^2 &= \left( \int_{-\infty}^\infty e^{-x^2} dx \right) \left( \int_{-\infty}^\infty e^{-y^2} dy \right) \\
 &= \int_{-\infty}^\infty \int_{-\infty}^\infty e^{-(x^2+y^2)} dx dy.
 \end{aligned}$$

Adotando, sobre coordenadas polares que  $x = r \cos \theta$ ,  $y = r \sin \theta$ , temos que:

$$A^2 = \int_0^{2\pi} \int_{-\infty}^{\infty} e^{-r^2} r dr d\theta.$$

Fazendo a mudança de variável  $u = r^2$ , que implica  $\frac{du}{2} = r dr$  temos,:

$$\begin{aligned} A^2 &= \int_0^{2\pi} \int_0^{\infty} \frac{e^{-u}}{2} du d\theta \\ &= \pi \int_0^{\infty} e^{-u} du \\ &= \pi \lim_{b \rightarrow \infty} \left( \int_0^b e^{-u} du \right) = \pi. \end{aligned}$$

Portanto,

$$A = \int_{-\infty}^{\infty} e^{-x^2} dx = \sqrt{\pi}.$$

□

A função possui as quatro propriedades abaixo.

**Proposição A.4.** *A função Gama, expressa por  $\Gamma$ , possui como propriedades:*

P1.  $\Gamma(1) = 1$ .

P2.  $\Gamma(x + 1) = x\Gamma(x)$ .

P3.  $\Gamma\left(\frac{1}{2}\right) = \sqrt{\pi}$ .

P4.  $\Gamma(n + 1) = n!$ .

*Demonstração.*

P1. Tomemos  $\Gamma(x) = \int_0^{\infty} t^{x-1} e^{-t} dt$ , com  $x = 1$ .

$$\Gamma(1) = \int_0^{\infty} t^{1-1} e^{-t} dt \Rightarrow$$

$$\Gamma(1) = \int_0^{\infty} e^{-t} dt \Rightarrow$$

$$\Gamma(1) = \lim_{b \rightarrow \infty} \left( \int_0^b e^{-t} dt \right) \Rightarrow$$

$$\Gamma(1) = \lim_{b \rightarrow \infty} \left( -e^{-t} \Big|_0^b \right) \Rightarrow$$

$$\Gamma(1) = \lim_{b \rightarrow \infty} \left( -e^{-b} - (-e^0) \right) \Rightarrow$$

$$\Gamma(1) = \lim_{b \rightarrow \infty} \left( -\frac{1}{e^b} + 1 \right) \Rightarrow$$

$$\Gamma(1) = 1.$$

P2.

$$\Gamma(x+1) = \int_0^{\infty} t^x e^{-t} dt \Rightarrow$$

$$\Gamma(x+1) = \lim_{b \rightarrow \infty} \left( \int_0^b t^x e^{-t} dt \right).$$

Usando integral por partes e adotando,

$$u = t^x \rightarrow du = xt^{x-1} dt$$

$$dv = e^{-t} dt \rightarrow v = -e^{-t},$$

temos,

$$\Gamma(x+1) = \lim_{b \rightarrow \infty} \left( \int_0^b t^x e^{-t} dt \right) \Rightarrow$$

$$\Gamma(x+1) = \lim_{b \rightarrow \infty} \left( -t^x e^{-t} \Big|_0^b - \int_0^b -e^{-t} x t^{x-1} dt \right) \Rightarrow$$

$$\Gamma(x+1) = \lim_{b \rightarrow \infty} \left( \frac{-b^x}{e^b} + \int_0^b e^{-t} x t^{x-1} dt \right) \Rightarrow$$

$$\Gamma(x+1) = \lim_{b \rightarrow \infty} \left( \frac{-b^x}{e^b} \right) + \lim_{b \rightarrow \infty} \left( \int_0^b e^{-t} x t^{x-1} dt \right) \Rightarrow$$

$$\Gamma(x+1) = \lim_{b \rightarrow \infty} \left( x \int_0^b e^{-t} t^{x-1} dt \right) \Rightarrow$$

$$\Gamma(x+1) = x \int_0^{\infty} e^{-t} t^{x-1} dt \Rightarrow$$

$$\Gamma(n+1) = x \Gamma(x).$$

P3. Para essa demonstração tomemos o Lema [A.3](#).

$$\int_{-\infty}^{\infty} e^{-x^2} dx = \sqrt{\pi}.$$

Observando que a função  $e^{-x^2}$  é par, podemos reescrever:

$$2 \int_0^{\infty} e^{-x^2} dx = \sqrt{\pi}.$$

Substituindo  $x^2 = u$ , implicando que  $du = \frac{dx}{2\sqrt{u}}$ , temos que:

$$2 \int_0^{\infty} \frac{e^{-u}}{2\sqrt{u}} du = \sqrt{\pi} \Rightarrow$$

$$\int_0^{\infty} u^{\frac{1}{2}-1} e^{-u} du = \sqrt{\pi} \Rightarrow$$

$$\Gamma\left(\frac{1}{2}\right) = \sqrt{\pi}.$$

P4. Provemos por indução finita.

Tomando para  $n = 1$ , fica trivial utilizando o item P1.

Por hipótese de indução tomemos ser válida para um  $k$ , sendo  $\Gamma(k) = (k-1)!$ .  
Provemos implicar para o caso  $k+1$ , utilizando o item P2.

$$\begin{aligned}\Gamma(k+1) &= k \cdot \Gamma(k) \Rightarrow \\ \Gamma(k+1) &= k \cdot (k-1)! = k!.\end{aligned}$$

□

# B Linhas de programação - Algoritmo p-ádica e algoritmo das matrizes

## LINHAS DE PROGRAMAÇÃO: ALGORÍTIMO P-ÁDICA

```
Var
a, p, m, c, s, n, v, mult, guardar, soma fim, t, q, k, u, quociente, f, soma, r, alpha: inteiro

Inicio
n <- 2
enquanto n <= 100 faça
    c <- 2
    enquanto (n MOD c <> 0) faça
        c <- c + 1
    fimenquanto
    se c = n então
        p <- n
        alpha <- 1
        enquanto alpha <= 100 faça
            cronometro on
            a <- alpha
            soma fim <- 0
            repita
                s <- a
                m <- 0
                repita
                    m <- m + 1
                    a <- a div p
                ate a = 0
            t <- m
            u <- m - 1
            se u = 0 então
                guardar <- s
                soma fim <- soma fim + guardar
                a <- 0
            fimse
            se u <> 0 então
```

```

k<-1
soma<-0
  repita
    k<-p*k
    soma<-k + soma
    u<-u - 1
  ate u = 0
r<-soma
  se s = soma entao
    escreval(p,,alpha)
    a<-0
  fimse
se s <> soma entao
  repita
    m<-t
    v<-p - 1
    k<-1
    repita
      k<-p*k
      m<-m - 1
    ate m = 0
    q<-k - 1
    v<-v*s
    quociente<- v div q
    t<-t - 1
  ate quociente > 0
f<-t
k<-1
  repita
    k<-k*p
    t<-t - 1
  ate t = 0
guardar<-quociente*k
k<-1
soma<-0
t<-f
  repita
    k<-k*p
    soma<-k + soma
    t<-t - 1
  ate t = 0
soma<-soma + 1
mult<-quociente*soma
a<-s - mult
soma fim<-soma fim + guardar
fimse
fimse
ate a=0

```

```

                se s<>r entao
                    escreval(p,,alpha)
                fimse
                alpha<-alpha+1
            cronometro off
        fimenquanto          fimse
    n<-n + 1
fimenquanto
Fimalgoritmo

```

A seguir temos as linhas de programação do algoritmo das matrizes.

### LINHAS DE PROGRAMAÇÃO: ALGORÍTIMO DAS MATRIZES

```

Var
p, a, t, k, d, l, soma, somaa, r, b, s, m, n, solucao, i, j, h, f, g, q, z, repetir, c, num: inteiro
Inicio
num<-2
    enquanto num <= 10 faca
        c<-2
        enquanto (num MOD c <> 0) faca
            c<-c + 1
        fimenquanto
        se c = num entao
            a<-1
            enquanto a<=10 faca
                cronometro on
                p<-num
                t<- p*p + p + 1
                g<-p + 1
                b<-p*p + p - 1
                se a < t entao
                    k<-1
                    se a<=b entao
                        se a MOD g <> 0 entao
                            se a MOD g = p entao
                                escreval(p," ",a)
                            senao
                                h<-a div g
                                j<-a MOD g
                                i<-h + 1
                                solucao<- p * j + p * p*(i - 1)+p *
                                p * p*(k - 1)
                                escreval(p," ",a)
                            fimse
                        fimse
                    se a MOD g = 0 entao
                        j<-p
                        i<- a div g
            fimse
    fimse

```

```

p * p*(k - 1)
solucao<- p * j + p*p*(i - 1)+p*
    escreval(p," ",a)
    fimse
fimse
    se a > b entao
        escreval(p," ",a)
    fimse
fimse
se a = t entao
    k<-1
    j<-p
    i<-p
    solucao<- p*j+p*p*(i-1)+p*p*p*(k-1)
    escreval(p," ",a)
fimse
se a > t entao

1
    repita
        d<-0
        l<-0
        n<-t * k
        repita
            l<-k div p
            d<-d + l
            k<-k div p
        ate l<=0
        k<-n div t
        soma<- n + d
        k<-k + 1
    ate a<=soma
k<-k - 1
se soma = a entao
    j<-p
    i<-p
    solucao<- p * j + p*p*(i - 1)+p*

p * p*(k - 1)
    escreval(p," ",a)
fimse
se soma<>a entao
    k<-k - 1
    m<-t * k
    r<-0
    repita
        s<-k div p
        r<-r + s
        k<-k div p
    ate s<=0

```



```

soma ← -m + r
k ← -m div t
f ← a - soma
se f ≤ b então
  se f MOD g ≠ 0 então
    se f MOD g = p então
      escreva(p, " ", a)
    senão
      k ← -k + 1
      h ← -f div g
      j ← -f MOD g
      i ← -h + 1
      solucao ← p *
j + p * p*(i - 1) + p * p * p*(k - 1)
      escreva(p, " ", a)
fimse
fimse
se f MOD g = 0 então
  k ← -k + 1
  j ← -p
  i ← -f div g
  solucao ← p*j + p*p*(i - 1) + p*
  escreva(p, " ", a)
fimse
fimse
se f > b então
  escreva(p, " ", a)
fimse
fimse
a ← -a + 1
cronometro off
fimenquanto
fimse
num ← -num + 1
fimenquanto
Fimalgoritmo

```

# Referências Bibliográficas

- [LDB 1996] LDB: *Leis de diretrizes básicas*. 1996
- [BNCC 2018] BNCC: *Base Nacional Comum Curricular*. Bd. 3º versao. 2018. – 598 S
- [Instituto Federal de educação 2022] AO, ciência e t. Instituto Federal de educação  
*MANUAL DO VISUALG*. 2022. – URL [http://www.inf.ufsc.br/~bosco.sobral/ensino/ine5201/Visualg2\\_manual.pdf](http://www.inf.ufsc.br/~bosco.sobral/ensino/ine5201/Visualg2_manual.pdf)
- [HEFEZ 2016] HEFEZ, Abramo: *Aritmética*. 2016
- [Olimpédia ] OLIMPÉDIA: *Formula de Polignac*
- [Parreira 2017] PARREIRA, Fábio: *INTRODUÇÃO A ALGORITMOS*. 2017. – URL [http://www.inf.ufsc.br/~bosco.sobral/ensino/ine5201/Visualg2\\_manual.pdf](http://www.inf.ufsc.br/~bosco.sobral/ensino/ine5201/Visualg2_manual.pdf)
- [RAMIREZ 2005] RAMIREZ, Julieth Paola S.: FUNCAO GAMA. In: *UNICAMP/IMECC* (2005)
- [Sensagent ] SENSAGENT: *definition – de polignac s formula*
- [SOUTO 2017] SOUTO, Flavia Cristine F.: O ensino de matemática e a resolução de problemas contextualizados nos anos iniciais do ensino fundamental. In: *Encontro Paranaense de Educacao Matemática* (2017)
- [VALENTE 2016] VALENTE, José A.: Integração do pensamento computacional no currículo da educação básica: diferentes estratégias usadas e questões de formação de professor e avaliação do aluno. In: *e-Curriculum* (2016)
- [VISUALG ] VISUALG: *VisualG*
- [Wikipédia 2021] WIKIPÉDIA: *Formula de Legendre*. 2021. – URL [https://pt.abcdef.wiki/wiki/Legendre's\\_formula](https://pt.abcdef.wiki/wiki/Legendre's_formula)
- [WING 2014] WING, Jeannette M.: Computational Thinking Benefits Society. In: *Microsoft Research* (2014)