Universidade Federal da Bahia - UFBA
Instituto de Matematica e Estatística - IME
Sociedade Brasileira de Matematica - SBM
Mestrado Profissional em Matemática em Rede Nacional - PROFMAT
Dissertação de Mestrado

# Aspectos Semióticos na Produção de Animações Matemáticas

João Carlos dos Anjos Cardoso

**Salvador - Bahia**
Maio de 2024

Ficha catalográfica elaborada pela Biblioteca Universitária de Ciências e Tecnologias Prof. Omar Catunda, SIBI – UFBA.

C268 Cardoso, João Carlos dos Anjos
Aspectos semióticos na produção de animações matemáticas / João Carlos dos Anjos Cardoso. – Salvador, 2023.
108 f.

Orientador: Prof. Dr. Vinícius Moreira Mello

Dissertação (Mestrado Profissional em Matemática em Rede Nacional – PROFMAT) – Universidade Federal da Bahia, Instituto de Matemática e Estatística, 2023.

1. Semiótica. 2. Recursos computacionais – Ensino e aprendizagem. 3. Matemática - Estudo e ensino. I. Mello, Vinícius Moreira. II. Universidade Federal da Bahia. III. Título.

CDU:51:004.438

# Aspectos Semióticos na Produção de Animações Matemáticas

João Carlos dos Anjos Cardoso

Dissertação de Mestrado apresentada à Comissão Acadêmica Institucional do PROFMAT-UFBA como requisito parcial para obtenção do título de Mestre em Matemática.

**Orientador**: Prof. Dr. Vinícius Moreira Mello.

**Salvador - Bahia**

Maio de 2024

“Aspectos Semióticos na Produção de Animações Matemáticas”

JOÃO CARLOS DOS ANJOS CARDOSO

Dissertação de Mestrado apresentada à comissão Acadêmica Institucional do PROFMAT-UFBA como requisito parcial para obtenção do título de Mestre em Matemática, aprovado em 15/05/2024.

**Banca Examinadora:**

Vinicius Moreira Mello

Prof. Dr. Vinícius Moreira Mello (orientador)
Instituto de Matemática e Estatística - UFBA

Kleyber Mota da Cunha

Profª. Dr. Kleyber Mota da Cunha
Instituto de Matemática e Estatística - UFBA

Maridete Brito Cunha Ferreira

Profª. Drª. Maridete Brito Cunha Ferreira
Universidade do Estado da Bahia (UNEB)

*À minha família*

# Agradecimentos

Agradeço ao Senhor Jesus Cristo e todos que contribuíram direta ou indiretamente para meu sucesso nessa etapa da minha vida acadêmica e profissional.

*"STANDING ON THE
SHOULDERS OF GIANTS"*
*Isaac Newton*

# Resumo

Este trabalho integra à temática ensino de matemática três temas que mantêm, entre si, um diálogo mais atual do que nunca: filosofia da matemática, semiótica e recursos computacionais no ensino. Isso se justifica pelo fato de, em todos os níveis de ensino, a matemática ser uma área na qual os estudantes ainda apresentam uma considerável dificuldade de aprendizagem e pelo fato de existirem muitos trabalhos em educação matemática apontando o uso de recursos computacionais como um auxiliar na resolução dessas dificuldade, embora isso, por si só, não seja suficiente para resolver o problema. O fato de as crianças, adolescentes e jovens já serem nativos digitais pode explicar essa importância. Mas compreender as dificuldades no aprendizado do conhecimento matemático implica, sem duvida, compreender o que é esse conhecimento matemático, isto é, sua natureza, que é abstrata. A Teoria dos Registros de Representação Semiótica discorre muito bem sobre isso e enfatiza sua relação com a aprendizagem em matemática. Então esses temas são, de modo lógico e coeso, abordados e integrados no trabalho, até serem aplicados à analise do produto educacional associado ao mesmo, que corresponde a animações envolvendo representações de alguns conceitos aritméticos e algébricos, desenvolvidas a partir de algoritmos computacionais em linguagem Python, com uso da biblioteca Manim. Neste trabalho, o novo recurso que possibilita usar códigos Python para fazer construções no GeoGebra também é usado para fazer animações da mesma natureza, de um modo de melhor usabilidade em sala de aula pelo professor leitor, encerrando o trabalho.

# Abstract

This work integrates three themes into the teaching of mathematics that maintain a more current dialogue with each other than ever before: philosophy of mathematics, semiotics and computational resources in teaching. This is justified by the fact that, at all levels of education, mathematics is an area in which students still have considerable learning difficulties and by the fact that there are many works in mathematics education pointing to the use of computational resources as an aid in resolution of these difficulties, although this, in itself, is not enough to solve the problem. The fact that children, teenagers and young people are already digital natives may explain this importance. But understanding the difficulties in learning mathematical knowledge undoubtedly implies understanding what this mathematical knowledge is, that is, its nature, which is abstract. The theory of semiotic representation registers discusses this very well and emphasizes its relationship with learning in mathematics. Then these themes are, in a logical and cohesive way, addressed and integrated into the work, until they are applied to the analysis of the educational product associated with it, which corresponds to animations involving representations of some arithmetic and algebraic concepts, developed from computational algorithms in language Python, using the Manim library.The new feature that allows the use of Python codes to make constructions in GeoGebra is also used to make animations of the same nature, in a way of better usability in the classroom by the reading teacher, ending the work.

# Sumário

**Introdução** **1**

**1 Semiótica e a Natureza do Conhecimento Matemático** **4**

1.1 Congruência e Correspondência Semântica . . . . . . . . . . . . . . . . . . . . 9

1.2 O GeoGebra e os registros de representação semiótica na matemática . . . 10

1.3 A necessidade semântica e didática de mais de um registro . . . . . . . . . 15

1.4 Metodologia Concreto–Pictórico–Abstrato . . . . . . . . . . . . . . . . . . . . 16

1.5 Tratamento como auxiliar à conversão . . . . . . . . . . . . . . . . . . . . . . 17

1.6 Objetos matemáticos como classes de objetos e os registros de representação semiótica . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 19

**2 Algoritmos Computacionais em Linguagem Python** **22**

2.1 Construindo o conceito de algoritmo computacional . . . . . . . . . . . . . 22

2.2 Paradigmas da programação e Linguagem Python . . . . . . . . . . . . . . 26

**3 Semiótica, a Aritmética dos Números Naturais e o Produto Educacional** **31**

3.1 A Biblioteca Manim . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 36

3.2 Descrição de uma Animação . . . . . . . . . . . . . . . . . . . . . . . . . . . 37

3.3 Um exemplo com expressão algébrica no Produto Educacional . . . . . . . 46

3.4 A estrutura geral dos códigos em Python por trás de cada animação . . . 51

3.5 Recursos do Python na versão GeoGebra5 (beta) . . . . . . . . . . . . . . 56

**4 Considerações Finais** **59**

**Referências Bibliográficas** **63**

# Introdução

A pergunta norteadora deste trabalho é se o uso de animações computacionais, simulando de modo dinâmico uma representação pictórica de objetos e conceitos matemáticos, permite uma melhor compreensão dos mesmos.

Não foi possível aplicar essa pesquisa no contexto empírico de sala de aula, porém algumas implicações vindas da análise de proposições da Teoria dos Registros de Representação Semiótica e o conceito de matemática dinâmica permitiram apontar uma resposta positiva para essa questão, sugerindo futuras pesquisas de campo, em sala de aula, na intenção de confirmar ou não essa conclusão, na instância estritamente teórica, embora parte das referências deste trabalho possa nos ter fornecido proposições vindas também de pesquisas empíricas feitas para confirmar tais proposições.

Assume-se aqui que a vantagem de se trabalhar com esse *software* de animação, em relação ao uso do material concreto, que já é e pode naturalmente ser feito, está no conceito de matemática dinâmica, o qual inspirou a construção dos códigos da animação, e que caracteriza, também, a natureza do *software* matemático GeoGebra, pois o aluno pode ver, de um modo instantâneo que alterações em uma representação automaticamente implicam mudança na outra representação, o que aponta para o fato de que objeto é o mesmo, a despeito da representação escolhida para representá-lo, ideia que está no cerne da Teoria dos Registros de Representação Semiótica. O produto educacional, embora não exiba uma mudança exatamente simultânea entre as representações, se inspirou nos *softwares* de matemática dinâmica para dar um sincronismo, com um *gap* de tempo suficientemente pequeno entre a mudança do registro do objeto de um sistema para outro (no caso do produto educacional, foram representação pictórica e representação simbólica).

Correspondentemente, uma mudança em uma representação sem uma mudança suficientemente simultânea (da unidade significativa correspondente) na representação em outro sistema causaria uma inconsistência no objeto matemático representado. Na realidade teríamos dois e não um objeto representado, cada um em um sistema semiótico diferente. Esse é um importante diferencial que os aplicativos de matemática dinâmica ou programas inspirados nesse paradigma, a exemplo das animações de nosso Produto Educacional, apresentam em relação ao uso de material concreto.

Além disso a última geração de pessoas já é considerada nativa digital, ou seja, já nasceram imersos em um mundo no qual a tecnologia permeia continuamente a vida cotidiana. Portanto, exigir uma menor necessidade de esforço cognitivo dos alunos para manter a atenção em uma mesma proposta educativa é mais uma vantagem do uso de um software educacional em relação ao uso de um material concreto.

O presente trabalho é desenvolvido ao longo de três capítulos. No primeiro capitulo, se discorre sobre a natureza abstrata do conhecimento matemático, fazendo uma breve reflexão filosófica sobre esse fato, e mostrando como isso se relaciona à aplicação da Teoria dos Registros de Representação Semiótica, de Duval, ao ensino de matemática. Nesse sentido, é feita uma conexão do que é dito na teoria com uma análise epistemológica e didática sobre exemplos de conteúdos matemáticos vistos no ensino fundamental e médio, como expressões algébricas, funções polinomiais do primeiro grau e estudo da reta e da circunferência em geometria analítica. Esse capítulo pode servir como uma introdução dos professores à Teoria dos Registros de Representação Semiótica.

Tudo isso é feito desenrolando-se importantes proposições da teoria de Duval de modo coeso e lógico, seguindo uma certa linearidade de raciocínio, no qual os exemplos aparecem para corroborar tais proposições expostas, através das citações feitas no capítulo.

No segundo capítulo, começamos o desenvolvimento de um conceito geral de algoritmo para, a partir daí, construir o caso particular de algoritmo computacional e os paradigmas para se construir um algoritmo computacional, desenvolvidos ao longo de todo século XX, que, finalmente, resultaram no paradigma mais atual e amplamente utilizado, e que também é a base de toda linguagem Python e de sua biblioteca Manim, utilizada na construção de nosso Produto Educacional (PE), que é a Programação Orientada a Objetos.

O terceiro capitulo traz uma descrição, juntamente com uma análise, do PE sob a ótica da Teoria dos Registros de Representação Semiótica, explanando o modo como essa teoria foi utilizada para fundamentar a construção das animações e explicando também os fundamentos computacionais, epistemológicos e didáticos que fundamentaram a construção dessas animações.

O trabalho é encerrado com animações construídas dentro da plataforma do GeoGebra, com o uso da linguagem Python, o que facilita fazer ajustes no código para atender a demandas educacionais específicas, com a intenção adicional de inspirar o leitor a fazer, também, novas construções com esse novo recurso, seja ele professor ou aluno.

Como apêndice, finalmente, é apresentado um plano de aplicação do PE em sala de aula, que idealmente é sugerido ser aplicado a turmas de 8° ano do ensino fundamental, no qual os alunos começam a generalizar as propriedades aritméticas dos números, sendo que a proposta original do PE é servir como um auxilio didático nesse processo.

# Capítulo 1

# Semiótica e a Natureza do Conhecimento Matemático

É usual do senso comum definir a matemática como "a ciência dos números", porém trata-se de uma definição simplista, que contempla apenas parte do que realmente estuda a matemática, no caso da definição, os números.

Os números, simplesmente, são os mais notáveis e quase sempre presentes, dos objetos de estudo dessa área do conhecimento. Mas será, agora, aqui usada, justamente, a ideia de número para uma melhor compreensão do que venha a ser a matemática.

Comecemos por uma definição mais completa da mesma: Matemática é a ciência que estuda entes abstratos e relações estabelecidas entre os mesmos. (pesquisar dicionário Houaiss).

Os números se ajustam perfeitamente como sendo um tipo desses entes abstratos aos quais a definição se refere, uma vez que os números não estão presentes na nossa realidade concreta a ponto de o percebermos com qualquer um dos nossos 5 sentidos, diferentemente de um objeto físico, como uma cadeira, mas representam uma abstração de algum aspecto da realidade, quantidade ou medida de algo dentro da mesma.

Por exemplo, ao vermos, ao redor de uma mesa, quatro cadeiras, além de vermos um objeto físico do tipo mesa e outros do tipo cadeira, podemos observar e abstrair (extrair) uma informação adicional dessa imagem que é a quantidade de cada tipo desses objetos, um e quatro, respectivamente, para mesa e cadeira. Essas quantidades tratam-se, portanto, de objetos ou entes abstratos, que, evidentemente, não têm forma física, diferente da cadeira ou da mesa em si. São objetos do tipo números naturais.

Essa é a natureza de tudo que estuda a matemática, sejam números, figuras geométricas, expressões algébricas, funções, matrizes, vetores, conjuntos, proposições etc. Portanto, por sua natureza abstrata, não temos acesso direto aos objetos matemáticos, exceto por sua representação.

Por exemplo, “4” não se trata do número quatro em si mas, sim, de uma das possíveis representações para este mesmo objeto. Outras representações possíveis são “$IV$”, pelo antigo sistema de numeração romano, ou “IIII”, traçado ou desenho que as crianças, nas fases escolares iniciais, costumam usar para visualizar e fazer operações com o número quatro (“ quatro palitinhos”). Todas são representações do número quatro e não o número em si.

Nessa perspectiva, a citação a seguir inclui uma definição para os objetos matemáticos e para a matemática:

> Os matemáticos platônicos definem os objetos matemáticos como entidades ideais que existiriam independentemente do espírito humano. Para os formalistas, a matemática é definida como a ciência da dedução formal, dos axiomas aos teoremas. Seus enunciados só tem conteúdo quando é fornecida uma interpretação. Para os mais radicais dentre eles, a matemática se resume em um jogo de linguagem sem relação com os “objetos” materiais.
>
> [11]

Por outro lado, defendendo a corrente de pensamento oposta, existem os matemáticos não-platônicos, que sustentam, portanto, que as ideias matemáticas surgem, necessariamente, como uma abstração da realidade concreta, a qual é feita pelo ser humano. Isto é, os objetos matemáticos não existem de modo independente da realidade concreta ou do ser humano.

Mas, qualquer que seja a corrente que se assuma, ambas pressupõe a natureza abstrata dos objetos matemáticos, logo a necessidade de representação para acessá-lo e operar cognitivamente sobre eles.

Uma vez frisado que um objeto matemático não é sua representação, é bem vinda uma definição do verbo “representar” no ensino de matemática:

> O verbo “representar”, quando usado no ensino de Matemática, indica a possibilidade de expor determinado objeto matemático de várias formas como escrita, notação, símbolo, traçado ou desenho. [14]

A citação acima entra em perfeita consonância com o que já foi afirmado a respeito da natureza abstrata dos objetos matemáticos e da necessidade de representá-los para torná-los acessíveis. Porém, como nenhuma representação é o objeto em si, sempre existiram várias representações possíveis do mesmo objeto.

Então, o único contato que professores e alunos têm com qualquer objeto matemático é, necessariamente, por meio dessas representações, o que pode nos levar a supor que estudar dificuldades na aprendizagem matemática implica estudar dificuldades na compreensão, interconversão e tratamento dessas representações. A citação abaixo aponta que esse é um dos pressupostos da Teoria de Registros de Representação Semiótica, no ensino de matemática, segundo Duval:

> O interesse de Duval está, principalmente, no funcionamento cognitivo do aluno. Para ele, o pensamento é ligado à operações semióticas e, consequentemente, não haverá compreensão possível sem recurso a essas representações. [4]

Nesse sentido, nesta dissertação, a proposta é trabalhar uma articulação entre as representações simbólicas usuais para os números naturais, para operações aritméticas de adição e multiplicação definidas sobre os mesmos, para as propriedades da primeira e uma representação num registro pictórico (figural) de cada um das mesmas, via uma animação construída em linguagem Python, com auxílio da biblioteca Manim. O mesmo é feito com a representação simbólica de algumas expressões algébricas simples na indeterminada $n$.

A ideia é permitir a construção de significado sobre esse tipo de expressão, fazendo uma conversão dinâmica para uma outra possível representação sua, no caso, uma representação pictórica.

Mas também podemos representar tais objetos (números, expressões algébricas ou operações) ou fatos matemáticos (propriedades) via seu enunciado em língua natural ou por um gráfico no plano cartesiano (como no caso de expressões algébricas, vistas como funções definidas sobre valores da indeterminada $n$). Essas diferentes formas de representar um mesmo objeto matemático são chamadas *sistemas semióticos de representação.*

Por exemplo, para ilustrar a conversão entre fórmula algébrica e a sua conversão em língua natural, consideremos o seguinte polinômio representado por sua fórmula algébrica: $2n+1$. Podemos convertê-la para sua representação em enunciado em língua natural como "o dobro de um número somado com um" ou "o sucessor do dobro de um número", ambas no mesmo sistema semiótico, podendo passar de um para o outro e vice-versa.

Conforme Duval, a passagem entre sistemas semióticos diferentes, como foi feito no Produto Educacional, entre representação pictórica e fórmula algébrica ou, como no último exemplo dado, entre fórmula algébrica e seu enunciado em língua natural é chamada *conversão*, enquanto a passagem dentro de um mesmo sistema semiótico, como entre "o dobro de um número mais um" e "o sucessor do dobro de um número", é chamada *tratamento.*

Do mesmo modo, dentro do sistema de fórmula algébrica, podemos fazer um tratamento na expressão $n+n+1$ para chegar à expressão $2n+1$. Portanto, tratam-se de duas representações de um mesmo objeto dentro de um mesmo sistema semiótico, conforme,a definição do próprio Duval:

> O tratamento de uma representação é a transformação dessa representação no mesmo registro onde ela foi formada. O tratamento é uma transformação interna a um registro. [8]

A seguir temos a definição de conversão de Duval:

> A conversão de uma representação é a transformação desta função em uma interpretação em outro registro, conservando a totalidade ou uma parte somente do conteúdo da representação inicial. [8]

l

Segundo Duval, existem vários tipos de tratamento e vários tipos de conversão. Entre os tipos de tratamento, temos a *paráfrase* e a *inferência*, no tratamento em linguagem natural. Um tipo de tratamento entre registros em linguagem natural é, justamente, o que se faz ao passar da representação "o dobro de um número somado com um" para "o sucessor do dobro de um número" e vice-versa.

O tipo de tratamento que fizermos ao passar de $n+n+1$ para $2n+1$ é chamado *cálculo*, especificamente nesse caso é o *cálculo algébrico*. Existem outros tipos de cálculo como o *cálculo numérico* e o *cálculo proposicional*, dentre outros. Existe também a *reconfiguração*, que é própria entre figuras geométricas, e *anamorfoses*, tratamento aplicado a toda representação figural.

Dentre os tipos de conversão podemos citar a *ilustração*, que é a representação em forma de figura de uma descrição em língua natural, e a *tradução* que é a passagem da informação codificada de um idioma para outro, dentre outras inúmeras formas de conversão, embora, de certo modo, possamos enxergar o último exemplo como tratamento, se considerarmos todos as línguas naturais produzidas pela humanidade como partes de um mesmo sistema semiótico.

A compreensão de um conceito ou objeto matemático requer o entendimento de mais de uma de suas representações, pois cada representação é apenas uma abstração, ou seja, um recorte do significado completo do objeto. Isto é, cada representação contempla apenas alguns de seus aspectos, outros não, os quais, por sua vez, são contemplados por uma representação em outro sistema semiótico.

Em uma função do 1° grau, por exemplo, sua fórmula algébrica é o que exibe a relação algébrica entre as coordenadas de seus pontos, e também o aspecto (valor) numérico de seu coeficiente angular e de seu coeficiente linear. Já uma certa reta (seu gráfico) é a figura geométrica que exibe esta relação expressa no plano cartesiano e o significado geométrico dos coeficientes, o angular e o linear,os quais só são contemplados pelo seu gráfico (representação gráfica no plano cartesiano), o que é justificado pela seguinte afirmação de Duval:

> ... toda representação é cognitivamente parcial em relação aquilo que ela representa, em registros diferentes não estão presentes os mesmos aspectos do conteúdo. [5]

Então, a importância da pluralidade de registros de representação semiótica não se dá apenas pelo fato da Teoria dos Registros de Representação Semiótica sustentar que essa é a única possibilidade que se dispõe para não confundir o conteúdo de uma representação com o objeto representado, como, também, pelo fato de que cada representação traz aspectos semânticos não tão evidenciados em outras representações, conforme se pode ler abaixo:

> Descartar a importância dos registros de representação leva a crer que todas as representações de um mesmo objeto matemático tem o mesmo conteúdo ou que seus conteúdos respectivos os deixam perceber um no outro com transparência. [7]

E tal crença não corresponde ao argumento desenvolvido acima (de que cada registro traz um aspecto semântico próprio de um objeto matemático).

Portanto, por exemplo, se pode notar que é impossível conhecer uma função do primeiro grau apenas por seus aspectos algébricos (sua fórmula) ou apenas pelo seu gráfico, isto é, por meio de um único sistema semiótico de representação. Tal raciocínio é estendido a qualquer objeto matemático, conforme disserta Duval:

> Se a conceitualização implica coordenação de registros de representação, o principal caminho das aprendizagens de base matemática não pode ser somente a automatização de certos tratamentos ou a compreensão de noções, mas deve ser a coordenação de diferentes registros de representação necessariamente mobilizados por estes tratamentos ou por esta compreensão. [8]

Representar está presente tanto para a construção conceitual de um objeto matemático, como através de raciocínios envolvendo objetos matemáticos, comunicação e aprendizagem em matemática. Portanto, ela faz parte do que podemos chamar de *pensamento matemático*, atividade de pensamento que envolve primordialmente conceitos e princípios matemáticos.

Assim, é imperativo que os alunos da educação básica, a fim de desenvolver essa requisitada competência, vindo seguir depois a carreira de matemático ou não (uma vez que a mesma é exigida em outras profissões e em outros aspectos da vida em sociedade), sejam estimulados a trabalhar, o tempo todo, com diversos tipos de representação de um objeto matemático, a começar por sua construção, que deve ser espontânea, mesmo que, a primeira vista, não seja a mais adequada.

Estimular essa construção a ser espontânea se justifica pelo fato de que é isso o que será encontrado pelos alunos na vida cotidiana, que é começar com esquemas imperfeitos da realidade e ir ajustando a algo mais convencionalmente aceito (no caso dos matemáticos, pela sua comunidade acadêmica) e fazer as adequadas transições entre diversos registros, seja por tratamento ou conversão, conforme é apontado a seguir:

> De acordo com o NCTM (2007), é importante desafiar os alunos a representar suas ideias matemáticas de maneiras que realmente façam sentido para eles, mesmo que essas representações sejam um pouco não convencionais no início. [16]

## 1.1 Congruência e Correspondência Semântica

Quando existe uma *congruência semântica* na conversão de uma representação para outra, qualquer alteração numa *unidade significativa* da representação em determinado *sistema semiótico* resulta na representação de outro objeto, de modo que essa alteração, também tem implicações na representação em outros *sistemas semióticos*, conforme a citação abaixo:

> O único modo de discriminar as *unidades significantes* de uma representação é realizar a observação, por um lado, variações de representação sistematicamente efetuadas em um registro e, por outro lado, as variações concomitantes de representação em outro registro. [8]

Esse fato pode ser observado tomando como exemplo, ainda, a classe de objetos matemáticos chamada função do primeiro grau, modificando ora seu coeficiente angular ou seu coeficiente linear na sua fórmula algébrica ou ambos e notar uma a concomitante variação no seu gráfico, produzindo portanto objetos matemáticos distintos, com representações próprias em cada um dos dois sistemas. Portanto, podemos afirmar que o coeficiente linear e o coeficiente angular são as unidades significativas de uma função do primeiro grau.

Podemos experienciar isso através do GeoGebra que é um *software* educacional de *matemática dinâmica* que dentre suas principais funcionalidades está relacionar aspectos algébricos e aspectos geométricos de um objeto matemático, a exemplo de funções reais a valores reais, onde estão inclusas as funções polinomiais do primeiro grau.

Veremos na próxima seção um exemplo inicial desse tipo de relação de uma função do primeiro grau e como, ao modificar a fórmula no programa, modifica-se algum aspecto da representação geométrica.

É aí que entra o conceito de discriminação visual, que permite ver as semelhanças entre as diferentes retas, mas também as diferenças que são, justamente, determinadas pelas suas unidades significativas, coeficiente angular e coeficiente linear, representando, portanto, objetos distintos. Este conceito é bem explicada por Rincón e Pabón:

> Discriminação visual é a habilidade que permite comparar dois ou mais objetos, identificando suas semelhanças e diferenças [13]

O GeoGebra e o Produto Eduacional deste trabalho, ambos, contemplam dois processos importantes, quais sejam, o processo de Interpretação da Informação Figural (IFI) e o Processamento Visual (PV). Vejamos como definir esses conceitos:

> ...Bishop(1983) refere os processos de Interpretação da Informação Figural e Processamento Visual. A Interpretação da Informação Figural (IFI) é o processo de interpretação da representação visual para extrair informações delas. Sendo o processamento visual (PV) o processo inverso. [13]

De fato, no GeoGebra, ao construirmos uma figura geométrica, no plano cartesiano, a calculadora gráfica do *software* a descreve por sua fórmula algébrica, que contém os seus parâmetros e que a definem como, exatamente, tal figura (coeficientes linear e angular, no caso das retas, e centro e raio, no caso das circunferências, por exemplo). Nesse sentido ocorre a IFI. Inversamente, podemos definir o tipo de figura que queremos (reta ou circunferência, por exemplo) e os parâmetros da figura, descrevendo-a, portanto, através de sua fórmula algébrica, para obter sua forma gráfica (visual), o que caracteriza o PV, conforme foi definido por Rincon e Pabón.

Falemos um pouco sobre o GeoGebra e sobre o conceito de matemática dinâmica no qual o mesmo está inserido.

## 1.2 O GeoGebra e os registros de representação semiótica na matemática

GeoGebra é um *software* de matemática dinâmica que combina vários aspectos matemáticos tais como geometria, álgebra, estatística e cálculo em uma simples plataforma. Trata-se de uma poderosa ferramenta utilizada por professores e estudantes de matemática, como uma forma de conduzir experimentos matemáticos e explorações. O GeoGebra pode ser utilizado tanto como aplicativo como na *web*, sendo acessível a uma ampla gama de usuários.

Programas de geometria dinâmica, dos quais o GeoGebra é um proeminente exemplo, referem-se a *softwares* que permitem ao usuário criar e manipular figuras geométricas em tempo real, explorando aspectos geométricos pela interação com tais figuras. Um excelente recurso que permite isso, e portanto verificar como mudanças nas unidades significativas do objeto algebricamente implicam, automaticamente, mudanças nas respectivas unidades significativas da sua representação gráfica, são os controles deslizantes.

Por exemplo, ainda falando de funções do primeiro grau, podemos definir parâmetros `a` e `b` e o atribuímos aos coeficientes angular e linear de uma função polinomial do primeiro grau genérica $f(x) = ax + b$. O gráfico que inicialmente vai aparecer na tela representa a função com os valores que iniciamos os parâmetros `a` e `b`.

Por exemplo, podemos definir na calculadora gráfica `a = 2` e `b = 1`. O gráfico da função $f$ será, portanto, o gráfico de $f(x) = 2x+1$. Mas, quando definimos os parâmetros `a` e `b` como esses valores, a calculadora nos fornece, também, um controle deslizante que permite ao usuário variar o valor de cada parâmetro, dentro de um intervalo que o contém, por exemplo, de $-5$ a 5, em ambos os parâmetros. Não necessariamente devemos atribuir os mesmos intervalos de variação para ambos os parâmetros. Toda variação dos valores `a` e `b` feita pelo usuário no controle deslizante causa uma mudança em sua fórmula algébrica e, em tempo real, a respectiva mudança ocorre no seu gráfico.

Podemos ver, nas figuras 1.1, 1.2, 1.3, os efeitos geométricos (inclinação da reta) produzidos por mudanças no parâmetro `a`. Inicialmente instanciamos `a = 1` e `b = 1`. E, como se pode ver, o coeficiente angular $a$ corresponde a inclinação da reta em sua representação geométrica, correspondendo a tangente do ângulo formado com o eixo das abcissas (eixo $x$), uma vez que a mudança nesse parâmetro $a$, gera, justamente, mudança nessa característica da reta representada.

Agora, podemos ver, nas figuras 1.1, 1.2, 1.3, os efeitos geométricos (inclinação da reta) produzidos por mudanças no parâmetro `a`, coeficiente angular.

Podemos também ver, nas figuras 1.4 e 1.5, os efeitos geométricos produzidos por mudanças no parâmetro `b`, coeficiente linear, que, pela observação em tais mudanças, nos permite notar que corresponde a ordenada do ponto de abcissa 0 (do ponto de interseção da reta com o eixo das ordenadas (eixo $y$)).

Entre as características dos softwares de geometria dinâmica estão geometria interativa, construções, medições e cálculos, exploração de conceitos matemáticos, visualização de funções matemáticas, integração entre diferentes assuntos matemáticos. A intenção é que esse processo realizado pelo *software* ajude o estudante a ter autonomia para fazer o mesmo.

Dada uma função $f$, $f(x) = y$, onde $y = ax + b$, $a$ e $b$, números reais fixados, além dessa forma padrão, na qual o $y$ está isolado em destaque, evidenciado que $f$ representa a variável $y$ em função da variável $x$, e não o contrário, podemos fazer tratamentos nessa fórmula algébrica e obter outras formas que representam, exatamente a mesma função $f$,

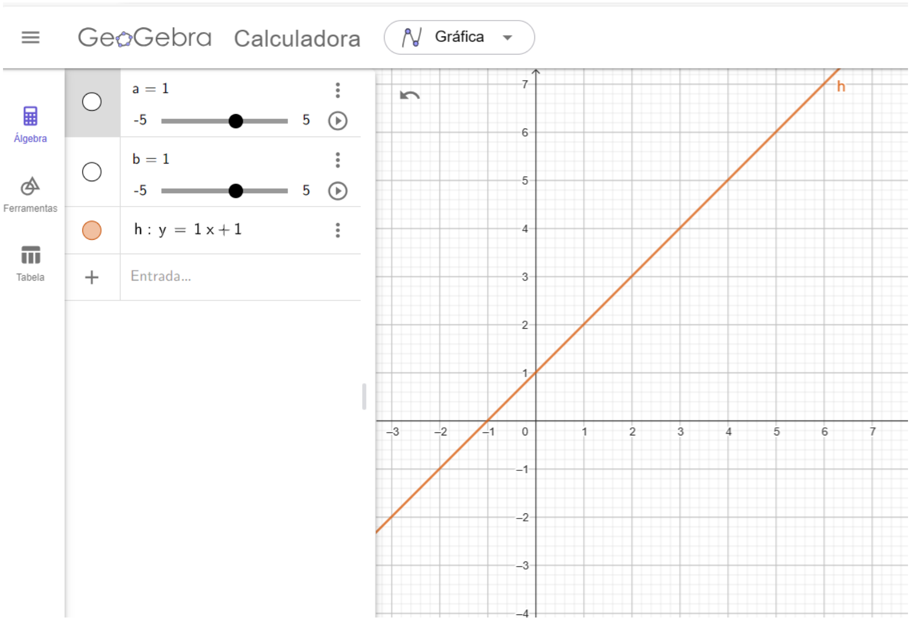


Figura 1.1

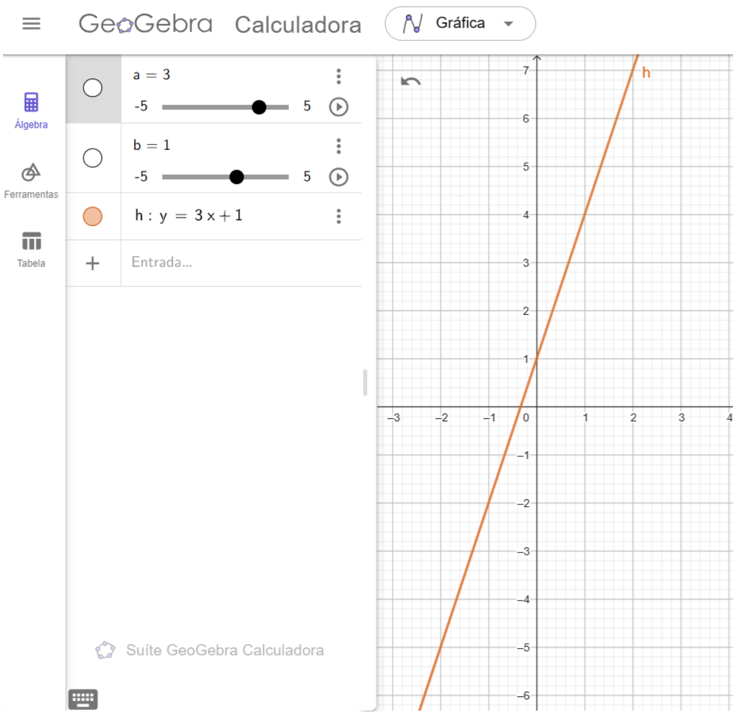


Figura 1.2

de $y$ em relação a $x$, e portanto, inserida na calculadora gráfica do GeoGebra produzirá, exatamente o mesmo gráfico, uma vez que mantivemos o objeto, a despeito do tratamento feito, porém diferentes fórmulas algébricas produzidas vão dar uma outra perspectiva do mesmo objeto.

Uma outra forma de representar essa função, seria $cy + mx + n = 0$, onde $a$ corresponde a $-m/c$ e $b$ corresponde a $-n/c$, e é ressaltado um aspecto mais geral dessa função,

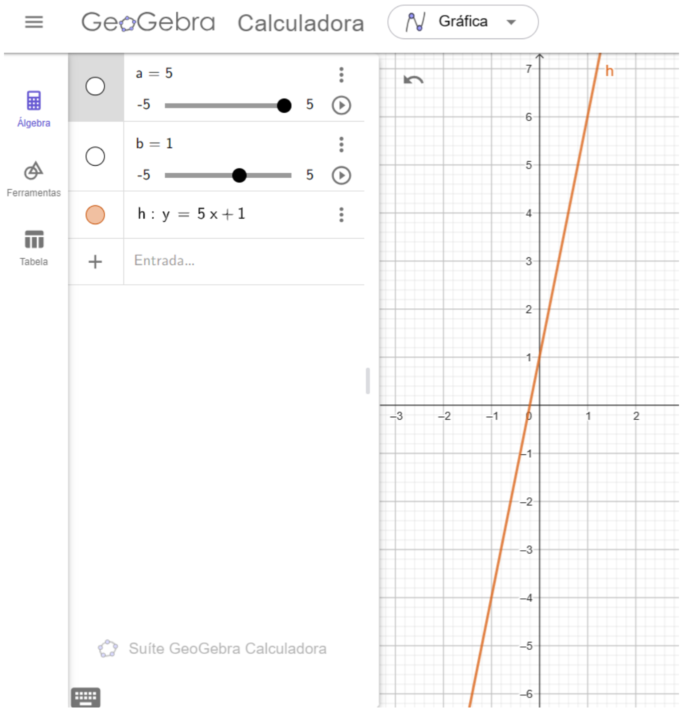


Figura 1.3

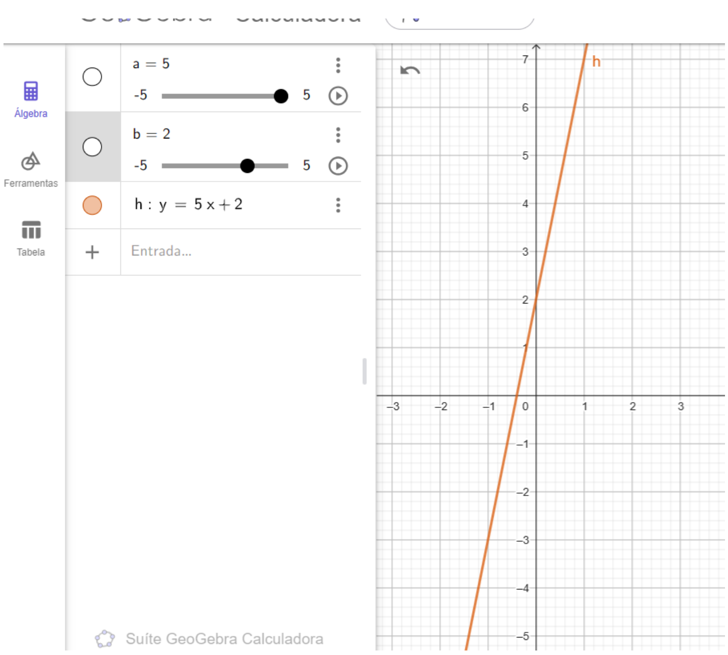


Figura 1.4

o fato de a mesma ser uma relação algébrica entre as variáveis $y$ e $x$, independentemente de se tratar, por definição, de uma função ou não, uma vez que essa forma de representação, diferentemente da anterior, não mostra uma evidência de quem seja a variável dependente ou a variável independente dessa função, até por que a função afim, como também, é conhecida, é bijetiva. Portanto, assim, como isolamos a variável $y$ para evidenciá-la como variável dependente, poderíamos, também, perfeitamente isolar a variável $x$.

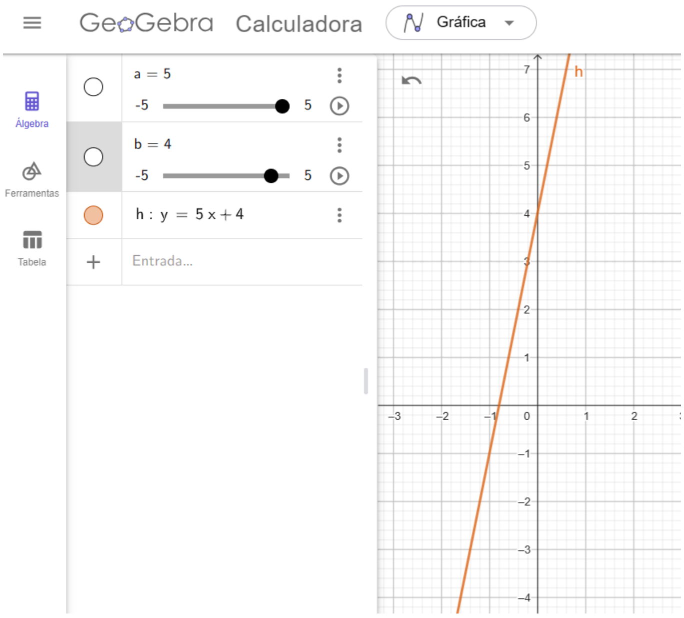


Figura 1.5

Esse fato de que até dentro de um mesmo sistema semiótico de representação podemos obter diferentes sentidos para um mesmo objeto matemático é reforçado pela seguinte afirmação:

> A representação 4/2, assim como (1+1), tem como referente o numeral 2, o que significa o indicativo do número dois. Por sua vez, este objeto (numeral dois) refere-se ao número dois, ou seja, a entidade abstrata que corresponde a uma quantidade, grandeza, intensidade. . . , portanto, à referência da representação 4/2 e também dá (1+1). Se há, então, no referente um substrato da referência, há também um sentido. No entanto, como foi dito, este sentido não será o mesmo para os dois modos de representação; ele vai depender da representação escolhida. [16]

E, conforme defende Duval (2009), se faz necessária a criação de condições que permitam ao aluno a visualização de um mesmo objeto matemático via várias representações no mesmo registro afim de criar uma significação mais consistente desse objeto.

Examinemos esse fenômeno se manifestando em outro exemplo trabalhado aqui: "O dobro de um número mais um" enfatiza o aspecto semântico que esse objeto, polinômio na indeterminada $n$, número natural, corresponde a soma entre dois números, porém um deles é o 1. Por outro lado, "o sucessor do dobro de um número natural" enfatiza que esse objeto é o elemento do conjunto imagem associado ao dobro de um número natural pela função sucessor definida sobre o conjunto dos números naturais a valores naturais.

## 1.3 A necessidade semântica e didática de mais de um registro

A depender da situação na qual se esteja trabalhando, seja num contexto em que se somar o dobro de um número com 1 é um caso especial da soma entre dois números naturais ou seja em um outro no qual se trata de um caso especial de aplicação da função sucessor sobre um número natural, uma outra forma de representação em língua natural de $2n + 1$ pode ser mais conveniente. Inclusive, haveria uma forma mais conveniente de registro algébrico para dar ênfase ao aspecto semântico de sucessor, que pode ser $s(2n)$, onde $s$ representa a função sucessor.

A conversão de um registro para outro não é algo cognitivamente imediato, sem uma análise minimamente detalhada das correspondentes unidades significativas em cada registro, como, mais uma vez, Duval afirma abaixo:

> Na realidade, a conversão entre gráficos e equações supõe que se consiga levar em conta , de um lado, as variáveis visuais próprias dos gráficos (inclinação, intersecção com os eixos etc) e, de outro, os valores escalares das equações (coeficientes positivos ou negativos, maior, menor ou igual a 1 etc.) [7]

E, de fato, por exemplo, perceber a correspondência dos coeficientes angulares e lineares de uma função polinomial do 1° grau com seus pares em seu registro geométrico ou gráfico (que é uma reta), que são a inclinação da reta e sua intersecção com o eixo das abcissas (eixo $y$), respectivamente, não é imediato.

Isso se evidencia com a prática comum de, dada a fórmula algébrica de uma função desse tipo, ser proposta, pelo professor, ao aluno, um procedimento análogo a um tratamento, que é a construção de uma tabela com duas colunas, cuja coluna da esquerda, tem-se, para a variável independente, $x$, incluindo o 0, valores inteiros que variam de 1 em 1 e, na coluna da direita, os respectivos valores de $f(x)$, como auxílio para um esboço do seu gráfico, o que é feito marcando os pontos

$$(x, f(x))$$

obtidos da tabela e ligando, com um linha cheia e reta, tais pontos.

Observando o gráfico construído, o aluno ao observar as variações de 1 unidade na horizontal, verifica a variação de $f(x)$ na vertical, justamente iguala a $a$ (coeficiente angular), mas essa variação é igual ao quociente entre essa variação e 1, que é a variação vertical sofrida, o que se nota, visualmente como a tangente do ângulo que é a inclinação de tal reta. Também, observa-se, na tabela que $f(0) = b$. Portanto, para $x = 0$, sobre o eixo das ordenadas (eixo $y$), $f$ vale $b$, isto é $f(x) = b$, ou seja, $b$, coeficiente linear, é o valor no qual a função intercepta o eixo das ordenadas.

De modo geral, Duval conclui que a conversão de representações, quaisquer que sejam os registros considerados, é redutível a um tratamento, expondo o seguinte argumento:

> Há, por trás da aplicação de uma regra de codificação para passar de uma equação a um gráfico cartesiano, a necessária articulação entre as variáveis cognitivas que são específicas do funcionamento cognitivo de cada registro. Pois são essas variáveis que permitem determinar quais unidades de significado pertinentes, que devem ser levadas em consideração em cada um dos registros [7]

## 1.4 Metodologia Concreto–Pictórico–Abstrato

A Teoria dos Registros de Representação Semiótica entra em consonância com uma das metodologias de ensino de matemática em Singapura, país asiático que tem tido excelentes resultados com seus estudantes em matemática.

Essa metodologia CPA (Concreto–Pictórico–Abstrato) leva em conta não só a natureza abstrata do conhecimento matemático e, portanto, a necessidade de representação para acessar seus objetos mas, também, que existem diversas representações cabíveis para um mesmo objeto matemático.

Essa ideia é aplicada no sentido que os conceitos matemáticos são apresentados gradativamente aos estudantes, a partir de uma representação concreta, mais palpável e manipulável, a exemplo de uma balança de dois pratos para representar o conceito de equação.

Nessa primeira etapa, a ideia tem um referencial mais próximo ao que o aluno já tem familiaridade, do seu cotidiano. Isto é, essa experiência permite ao aluno construir uma sólida fundação, ao fazer uma conexão mais próxima entre a matemática e o mundo concreto, o que está alinhado com a seguinte citação do psicólogo Jerome Bruner, psicólogo idealizador desta metodologia, assumindo que a criatividade e auto-descoberta são partes do processo de aprendizagem um novo conhecimento pelo aluno.

> A essência da criatividade é usar o conhecimento que já temos para tentar dar um passo adiante. [1]

Em seguida, passa-se pela representação pictórica, que é uma representação mais próxima à formal, porém, na mesma ainda existe um referencial visual, até atingir uma representação abstrata, simbólica, a usualmente adotada no fazer matemático. Nessa etapa, aparecem os símbolos, como as letras do nosso alfabeto e os numerais. Justamente, nesse momento, os alunos apresentam as clássicas dificuldades em matemática, por tais representações não apresentarem um referencial direto com o que eles já conhecem. Por isso, essa passagem gradativa se apresenta como didaticamente importante

A ideia do Produto Educacional é, justamente, ajudar os alunos a compreenderem a representação simbólica usual de alguns conceitos matemáticos, associando-os a uma conveniente representação pictórica, que o aluno tenha suficiente facilidade em compreender e associar ao que poderia ser sua representação simbólica e vice-versa.

Então, pressupõe-se que os alunos aos quais será destinado o Produto Educacional já passaram pela fase de manipular e reconhecer objetos por sua representação concreta e já conseguem, portanto, compreender representações pictóricas como desenhos, traçados, diagramas etc.

Os conceitos de “noesis” e “semiosis” são dois conceitos fundamentais da filosofia e da semiótica, respectivamente, que permeiam toda teoria dos Registros de Representação Semiótica de Raymond Duval. Tais conceitos são cruciais no entendimento de como o ser humano comunica, interpreta e constrói significado.

Noesis refere-se ao ato do indivíduo em se engajar intelectualmente ou cognitivamente para compreender aspectos do que observa no mundo. Conforme diz o autor, o que permite o aumento da cognição é a diversidade de representações semióticas e, por consequência, surge o desenvolvimento das representações mentais.C

A afirmação acima indica, portanto, que não existe noesis sem semiosis, isto é Toda operação mental feita nestes objetos é apoiada por operações, no papel, quadro, lousa digital etc, sobre suas representações, o que está no campo da semiose. Reciprocamente, não existe semiose sem noesis, pois todo símbolo matemático, o que está no campo da semiótica, passa a representação de algo a ser percebido, processado e conhecido pelo aluno. Daí vem uma clássica frase dessa teoria.

> Não há noésis sem semiósis, é a semiósis que determina as condições de possibilidade e de exercício da noésis. [6]

## 1.5 Tratamento como auxiliar à conversão

Dada a representação de um objeto em certo registro, nem sempre se consegue identificar de modo imediato suas unidades significativas para fazer a correspondente conversão para sua representação em outro registro.

Daí, a necessidade de se fazer um tratamento a fim de se obter uma forma na qual se perceba mais facilmente tais unidades significativas. Ou seja, muitas vezes um pré-tratamento é fundamental auxiliar no processo de conversão. Por exemplo,

$$x^2 + y^2 + 2x + 10y - 10 = 0$$

é uma a forma no registro algébrico de uma cônica com os coeficientes dos termos quadráticos iguais, portanto uma circunferência. Mas, para representá-la por meio de seu gráfico no plano cartesiano, precisa-se obter seu centro e seu raio, o que não está em evidência nessa representação.

Mas, podemos torná-los evidentes, ao fazermos alguns tratamentos a partir dessa fórmula, a começar por completar quadrados, como se pode ver nos passos abaixo:

$$x^2 + y^2 + 2x + 10y - 10 = 0 \Leftrightarrow$$

$$x^2 + 2x + 1 + y^2 + 10y + 25 - 1 - 25 - 10 = 0 \Leftrightarrow$$

$$(x + 1)^2 + (y + 5)^2 = 36$$

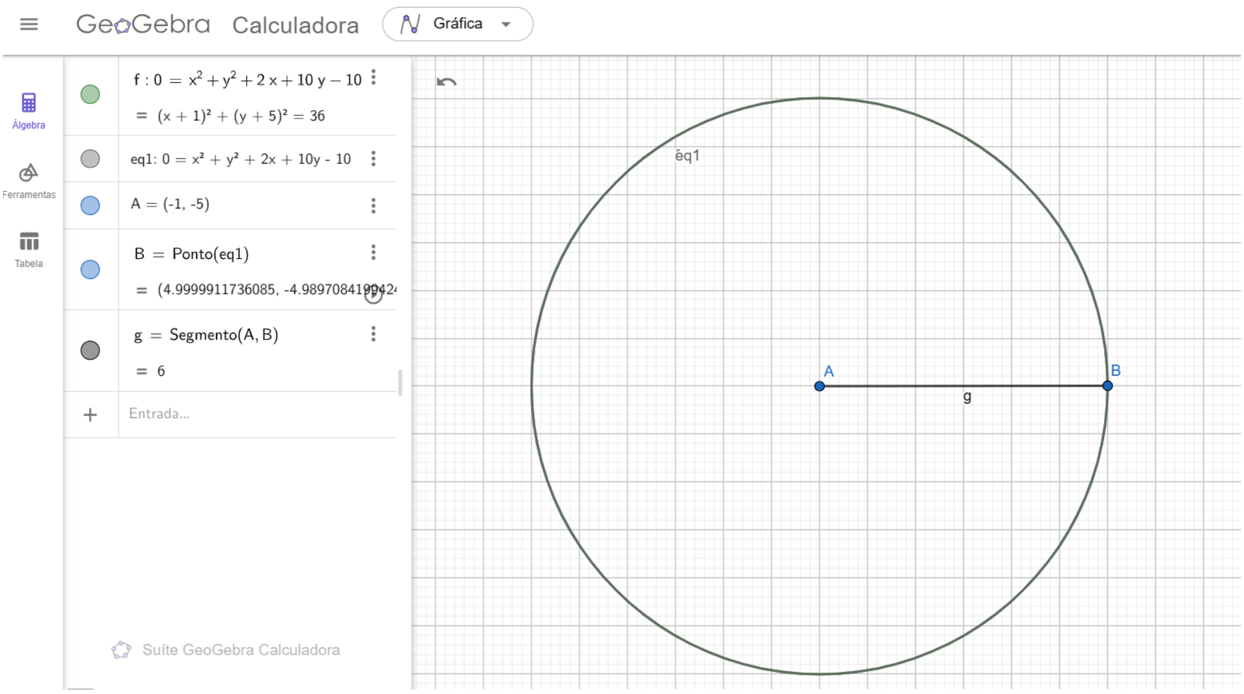


Figura 1.6

Portanto, chegou-se a uma forma na qual se nota que essa é a equação de uma circunferência de centro $(-1, -5)$ e raio 6, representação gráfica no plano cartesiano.

Inclusive, ao digitar, na janela de álgebra, a equação original, o GeoGebra faz, automaticamente, o tratamento para o último formato, conforme pode ser visto na figura abaixo, expressando algebricamente a circunferência da qual foi falada no parágrafo anterior e é exibida na tela do programa, ao lado, conforme é visto na figura 1.6. E pode-se observar, na calculadora gráfica que, de fato, as coordenadas de seu centro, A, são $(-1, -5)$ e o raio, que é equivalente a medida do segmento $AB$, vale 6.

Levando em conta o exemplo que já vem sendo trabalhado, da função polinomial do primeiro grau, o tratamento feito a partir de qualquer outra fórmula algébrica desse objeto para a forma $y = ax + b$, sendo $y = f(x)$, deixa evidente seu coeficiente linear (valor de $b$) e seu coeficiente angular (valor de $a$), que vai permitir melhor determinar, respectivamente, onde a reta intercepta o eixo das ordenadas (eixo $y$) e a tangente do ângulo de inclinação da mesma em relação ao eixo $x$.

Outro caso clássico no qual isso é relevante envolve a conversão de uma sentença em linguagem natural para sua expressão algébrica, processo recorrente na resolução de atividades ou exercícios matemáticos para alunos da educação básica e que é, comumente, fonte de dificuldades para esses alunos e objeto de estudos matemáticos nesse sentido, como os feitos pelo professor Dr. José Nilson Machado da USP. (citação)

Nesse contexto, consideremos um exemplo já visto, o de converter a expressão "o sucessor do dobro de um número" em sua expressão algébrica na indeterminada $n$. Essa conversão seria facilitada por um *pré-tratamento* da expressão original para a expressão "o dobro de um número mais um" ou "duas vezes um número mais um". Nesse caso, ficaria evidente não só o "$2n$", correspondendo ao "dobro de um número" ou a "duas vezes um número" como, também, o operador de adição(+) seguido do 1, o que indica a ideia de sucessor, quando segue o "$2n$".

Essa necessidade de se fazer um tratamento adequado para se chegar a uma forma do registro que apoie sua conversão para outro registro é comum entre objetos matemáticos de modo geral.

## 1.6 Objetos matemáticos como classes de objetos e os registros de representação semiótica

Muitas vezes, um objeto matemático não corresponde a um único objeto em si mas a uma classe (conjunto) de vários objetos que apresentam uma característica em comum, a despeito das demais diferenças entre os mesmos. Esse é um tipo de conjunto o qual chamamos de *classe de equivalência*, por existir uma relação de equivalência pela qual todos os elementos desse conjunto se relacionam entre si.

Por exemplo, um triângulo retângulo é um objeto matemático, porém não corresponde a um único triângulo, em particular, mas, sim, a um conjunto de vários triângulos, distintos entre si por outras propriedades, como ser isósceles ou não. Ainda assim todos apresentam um ângulo reto (que mede $90^o$).

Para representar geometricamente essa classe de triângulos, escolhemos um representante dentre os mesmos, desde que indiquemos que um dos seus ângulos mede $90^o$. Para fazer isso, convencionou-se, ao invés de usar um arco, como nos demais, representar este ângulo pela figura de um pequeno quadrado em um dos vértices, dois lados comuns entre o ângulo e o quadrado, e um ponto em seu interior.

Outro registro desse objeto se baseia no famoso Teorema de Pitágoras, o qual caracteriza essa classe de triângulos, uma vez que obedecer esse teorema é uma condição necessária e suficiente para um triângulo ser retângulo. Portanto, se alterarmos, significativamente, algo na sentença lógica que define os triângulos que satisfazem este teorema, estamos alterando o objeto que essa sentença representa, o qual, imediatamente, deixa de ser um triângulo retângulo e não é mais válida a indicação geométrica de que um de seus ângulos é reto e vice-versa.

Esse fato ocorre pois, conforme já visto, a alteração de unidades significativas de uma representação muda o objeto que está sendo representado, logo sua representação é mudada em qualquer outro registro. Uma vez que não temos mais o mesmo objeto inicial e, sim, outro, um registro diferente se faz necessário, mantendo uma consistência no sistema de representações semióticas de um determinado objeto.

Mas se fizermos qualquer outra alteração no triângulo, porém conservando as unidades significativas que o caracterizam como do tipo retângulo, obteremos outro elemento da classe mas, em termos de classe, ainda teremos o mesmo objeto, que é um triângulo retângulo, uma vez que não houve qualquer alteração na representação que o descaracteriza como tal.

Aí entra mais uma vez a habilidade de discriminação visual, citada pór Rincon e Pabón, a qual já foi tratada anteriormente aqui, que permite ao aluno, além de perceber as diferenças entre os diferentes triângulos dessa classe, notar a semelhança que os une, que é a propriedade de ser retângulo e todas as consequências ou equivalências lógicas disso, como o Teorema de Pitágoras.

Logo, por exemplo, sempre que um aluno reconhecer dentre as mais diversas formas de triângulo, um triângulo retângulo, ele vai saber que tem o Teorema de Pitágoras ou as relações métricas no triângulo retângulo como recurso para resolver problemas relacionados a esse triângulo.

A relação que divide (particiona) o conjunto dos triângulos em duas classes, a classe dos triângulos retângulos e a classe complementar dos triângulos não-retângulos é, de fato, uma relação de equivalência. O que justifica isso é que tal relação é definida do seguinte modo: Um triângulo se relaciona (está na mesma classe que) com um outro se e somente se ambos são triângulos retângulos ou ambos não são triângulos retângulos. Fazendo verificação lógica sobre essa proposição, chega-se facilmente à conclusão que estamos nos referindo a uma relação de equivalência entre triângulos.

Um outro exemplo de um objeto que é uma classe de equivalência é o vetor estudado em geometria analítica, que representa o conjunto de todos os segmentos orientados de mesmo comprimento (módulo), direção e sentido.

# Capítulo 2

# Algoritmos Computacionais em Linguagem Python

## 2.1 Construindo o conceito de algoritmo computacional

Programas de computador pessoal e aplicativos de *smartphone* estão, quase, a todo momento presentes em nosso cotidiano, para nos auxiliar em diversas atividades, como mobilidade, o caso do Waze, edição de texto, que é o caso do Microsoft Word ou do Google Docs. Podemos lançar mão do Microsoft Excel ou do Google Sheets para construir planilhas que nos auxiliem no nosso fluxo financeiro pessoal ou de uma empresa, podemos utilizar o WhatsApp ou o Telegram, por exemplo, para comunicação ou interações sociais a distância ou utilizar as animações interativas de sites para fins educacionais, como é o caso da plataforma PHET, da Universidade do Colorado.

O que todos estes programas, aplicativos ou sites têm em comum é que para exercer a tarefa para a qual os mesmos foram desenvolvidos, eles precisam seguir uma série de passos que resultam no efeito por nós desejado, em cada etapa da realização de tal tarefa.

Por exemplo, na construção de uma planilha financeira no Excel, se queremos acrescentar uma linha à tabela com a qual estamos trabalhando, executamos um determinado comando, que só pode ser efetivado por que o programador ou a equipe de programadores que desenvolveu o Excel pré-estabeleceu, na programação do mesmo, feito via uma linguagem de programação, uma instrução que equivaleria ao seguinte em português "se o usuário fizer o comando 'x', construa uma nova linha na tabela".

Isto é apenas um trecho do que se assemelha a uma receita de bolo, um manual de como funciona todo programa, nesse caso o Excel. Seria um manual de instrução dado pelo programador ao computador pessoal, *smartphone*, calculadora, *tablet* ou qualquer outro dispositivo computacional de como ele deve se comportar para que o *software* (programa ou aplicativo) instalado nele funcione da forma desejada, a fim de atender a demanda do usuário.

Este manual de instruções, nada mais nada menos, é o que o que conhecemos como *algoritmo*, que pode ser definido como uma sequência de passos para realizar uma tarefa. Isso ocorre a partir do processamento dos dados de entrada para produzir uma saída, que é o que queremos obter com a execução dessa tarefa.

Um algoritmo pode ser criado para ser executado por um dispositivo computacional, portanto, muitas vezes, de forma mais rápida e eficiente ou por nós, seres humanos mesmo, principalmente se a tarefa que queremos realizar é uma tarefa simples.

Por exemplo, somar 3 com 2 se mostra tão simples que nossa resposta é, praticamente, automática (quando ainda não decoramos o fato de que $3 + 2 = 5$. Nesse caso, não houve qualquer processamento mental para se dar a resposta e a resposta tende a ser literalmente automática, que é o comum quando já estamos na fase adulta, mas não quando estamos aprendendo a fazer contas de adição, uma vez aprendido a contar). Mas, ainda assim, trata-se de um algoritmo, no qual as entradas são os números 3 e 2 e o processamento para obter a saída desejada, que é a soma deles, é aplicar a função sucessor a partir da primeira entrada, que é 3, um número de vezes seguidas e igual a segunda entrada, que é 2, resultando em 5.

Porém existem algoritmos que são tão extensos, demorados e até praticamente inviáveis de serem executados por um ser humano de modo eficaz, que o ideal é recorrer aos dispositivos computacionais, como é o caso da conveniência de se usar uma calculadora científica quando queremos calcular seno de ângulos que não são notáveis ( os ângulos ditos notáveis são 30, 45 e 60) nem combinações aritméticas relativamente simples dos mesmos, o que poderia ser obtida mediante aplicação única ou combinada da fórmula de seno da soma ou seno da diferença de dois arcos.

Segue uma definição de algoritmos, que está em consonância com a conotação que foi dada aqui de uma receita de bolo ou manual de instruções escrito por um programador para desenvolver um *software* que vai nos ajudar na realização de alguma tarefa:

> Algoritmos são conjuntos de passos finitos e organizados que, quando executados, resolvem um determinado problema [12]

De fato, para fazer a soma de 3 com 2, da qual falamos, tivemos exatamente dois passos organizados a serem feitos. O primeiro passo foi tomar o sucessor de 3 e o segundo passo foi tomar o sucessor do número obtido (4), obtendo, no final 5.

A definição, também, diz que os passos, além de finitos, devem ser organizados e um modo de organização importante é a sequência em que esses passos ocorrem. Ter exatamente os mesmos passos não é suficiente para obter o resultado desejado do algoritmo. Organizá-los em determinada ordem específica é importante, senão necessário.

Um exemplo disso é o algoritmo usual de resolução de uma equação do $2^o$ grau, visto pelos alunos no $9^o$ ano do ensino fundamental. Os passos são calcular o valor do discriminante (*delta*) e aplicar a fórmula de Bhaskara, necessariamente nessa ordem. A ordem contrária, inclusive inviabilizaria qualquer saída pois o uso da fórmula de Bhaskara pressupõe já ter o valor do discriminante calculado, portanto ele deveria ser calculado em um passo anterior.

Esse detalhe é muito importante na programação de um dispositivo computacional uma vez que linguagens de programação como Java, C, C++ e Python, essa última a qual foi utilizada na codificação das animações que compõem o Produto Educacional, seguem uma abordagem sequencial, isto é, executam cada comando, um após o outro, exatamente, na ordem em que eles aparecem no código.

A única exceção a essa regra é na hora da aplicação das *estruturas de fluxo e controle*, como *loops* e *condicionais*, porém após terminado o que é feito dentro do *loop* ou *condicional*, a execução do programa passa a ser feita, na linha imediatamente abaixo, se o programa não finalizar ali.

Como já vimos, um algoritmo, para ser executado, não precisa de um dispositivo computacional. Por exemplo, uma receita de bolo, o manual de instruções para montar certo aparelho, a informação de como chegar a certo lugar a partir do ponto de partida, resolver uma equação polinomial do segundo grau (aprendemos a fazer na escola com papel e lápis ou caneta) se encaixam, perfeitamente, na definição de algoritmo dada acima e podemos nos valer deles sem o auxílio de um dispositivo computacional.

Porém, os algoritmos por trás de calculadoras, do cálculo de preço de viagens de aplicativos como UBER, de outras funções em outros tipos de aplicativos, programas de computador pessoal, sites, redes sociais etc., são rodados dentro de dispositivos computacionais como computador pessoal, celular, sistemas eletrônicos embarcados em aparelhos eletrodomésticos como as *smartTVs* e micro-ondas , ou industriais, computadores com muita memória de alto poder computacional das *bigtechs* (empresas poderosas do ramo de tecnologia), que podemos ter acesso por serviços de nuvem etc.

Esses algoritmos são chamados *algoritmos computacionais*, que será a classe de algoritmos à qual pertencem os algoritmos utilizados na construção do nosso Produto Educacional.

Dois conceitos fundamentais para a compreensão de algoritmos computacionais são os conceitos de *variável* e *memória*. A cada uso do programa, o usuário, a depender da sua necessidade, digita dados distintos a serem processados, afim de obter o resultado desejado. Ou, por outro exemplo, no caso de uma máquina de automação industrial, cada nova medida de temperatura, pressão, potencial elétrico etc registrada pelo seu sensor a respeito do ambiente sobre o qual a mesma está atuando resultará em entradas (de dados) distintas a serem processadas pelo algoritmo que nos darão saídas distintas, que são as informações que precisaremos pra fazer a devida atuação sobre tal ambiente por um robô ou um operador humano no intuito de obter sucesso no processo industrial no qual está inserido tal ambiente.

Daí faz sentido o uso do termo variável pois uma vez recebidos os dados digitados por um ser humano ou registrado por um sensor, em seus respectivos contextos (só para dar dois exemplos desse fato mais geral), tais dados são armazenados em um local do dispositivo físico (*hardware*) chamado memória, prontos para serem processados assim que a variável a qual o mesmo corresponde for solicitada dentro do código que expressa o algoritmo. Em um outro estado de condições ambientais ou em outras necessidades do usuário os valores antigos são substituídos pelos novos valores de entradas, em seus respectivos espaços de memória, prontos para serem processados durante a execução do algoritmo.

Mas para que isso ocorra, o programador deve alocar esse espaço de memória, ao declarar a variável no código, dizendo o tipo de dado que aquela variável representa, ou seja, que vai ser armazenado naquele espaço de memória, uma vez que diferentes tipos de dados (imagens, áudios, textos, números inteiros, números decimais etc.) ocupam mais ou menos memória. Então é necessário saber que dado será armazenado ali para até saber quanto será alocado de memória para isso. Porém o Python tem uma particularidade de não haver uma necessidade de pré-definir o tipo da variável, sendo portanto, comumente, categorizada como linguagem de tipagem fraca.

Para ilustrar melhor essa ideia, vamos trabalhar com um algoritmo computacional cuja tarefa é somar dois números. Naturalmente os dados de entrada são dois números, que podem ser inteiros ou racionais, mas como queremos um caso mais geral, vamos considerar que são dois números decimais (pontos flutuantes) e, como falamos isso, o tipo do dado, ponto flutuante ou *float*, na linguagem Python sob a qual foi construída o Produto Educacional, deve ser declarada junto com a variável, que pode ser representada por duas letras do nosso alfabeto, uma para cada variável, alocando um espaço de memória para quando o usuário for digitar os valores de $a$ e $b$, ou seja, dos números os quais ele quer somar entre si. Em seguida ocorre o processamento, onde dizemos o que deve ser feito digitando `s = a + b`, onde a variável `s` representa a soma que queremos e deve ser dada como saída no visor do dispositivo e o meio mais adequado para isso é usar o comando que imprime o valor de `s` obtido no processamento de somar o valor digitado para `a` com o valor digitado para `b`.

Por exemplo, se quisermos somar 3 com 2, ao digitar os dados no programa, `a` recebe `3` e `b` recebe `2`, portanto segue-se a instrução do processamento substituindo `a+b` por`3+2`, dando `s` igual a `5`, em seguida, imprimindo `5` na tela, que é o valor de `s`, ou seja, o valor que estávamos buscando, ao usar o programa. O mesmo processo ocorre ao pedir a soma de 598 com 217 e obter, na tela, 715 como resultado, por exemplo.

*Programar* ou construir um código é escrever um algoritmo que expressa o que queremos que seja feito como tarefa. Hoje, isto é feito por meio de linguagens de programação como as já citadas Java, C, C++ e Python. Cada uma dessas linguagens tem suas *sintaxes*, isto é, o modo ou a estrutura padrão de como os comandos devem ser escrito pelo programador, sob pena de não conseguir executá-lo, caso a escrita não esteja conforme.

Essa ideia de sintaxe para linguagens de programação é análoga à mesma ideia no caso das linguagens naturais. Cada idioma tem sua sintaxe ou modo de escrever as sentenças para expressar uma mesma informação. Por exemplo no português e espanhol, idiomas oriundos do latim, para cada pessoa e tempo, o verbo assume formas diferentes (conjugação verbal), já no caso do inglês, isso não acontece, isto é, a concordância verbal tem uma estrutra diferente.

Do mesmo modo, por exemplo, a estrutura da escrita de um código em Java difere da estrutura da escrita de um código para o mesmo algoritmo em Python, embora existam semelhanças que permeiam uma estrutura mais geral para a construção de um algoritmo em qualquer linguagem de programação, do mesmo modo que tanto o português, espanhol, inglês e outros idiomas como o *wolof* (idioma nativo do Senegal), preservam a estrutura geral sujeito/predicado para orações construídas em qualquer um desse idiomas.

Mas o que nos interessa, em particular, e é o que será feito aqui, é inicialmente uma breve explanação sobre a linguagem Python, pois, conforme já foi exposto, foi a linguagem usada, munida da biblioteca Manim, da qual falaremos mais adiante, para a construção das animações do nosso Produto Educacional.

## 2.2 Paradigmas da programação e Linguagem Python

A ideia do desenvolvimento da linguagem Python teve início em 1989, em Amsterdã, capital da Holanda, no Centro de Matemática e Ciência da Computação pelo programador Guido Van Hossun. Seu intuito foi desenvolver uma linguagem, ao mesmo tempo mais intuitiva, isto é, com comando mais simples de entender, e ao mesmo tempo poderosa, como ela de fato é hoje, uma linguagem de propósito geral, isto é, embora não possamos afirmar que ela é a melhor linguagem para tudo, ela tem aplicações nas mais diversas áreas, seja na construção de aplicativos, jogos, biotecnologia, inteligência artificial, sistemas eletrônicos embarcados, ciência de dados, animações educacionais, estatística etc.

Inúmeras empresas famosas e de grande influência no mercado a usam em seus sistemas como o Google, Globo.com, a NASA, a Microsoft etc. É uma linguagem totalmente *orientada a objetos*, isto é, cada dado na linguagem é, em última análise, um *objeto*.

O modo de programar passou por vários paradigmas ao longo do último século, quando a atividade se tornou cada vez mais comum. Então será exposta aqui essa evolução e usaremos o exemplo de um programa para somar 5 com 10 escritos dentro de cada um desses paradigmas e, para nos familiarizarmos com a sintaxe da linguagem Python, os exemplos códigos em alguns desses paradigmas são de códigos escritos na linguagem Python.

Essa evolução é feita de tal modo que chegamos ao paradigma mais recente, que permeia toda linguagem Python, e tornou programar mais fácil e intuitivo ao relacionar, fazer analogias, do ato de programar com aspectos de objetos do mundo real, sendo, portanto, mais fácil, aos aspirantes a programador, hoje se inserirem nesse mercado do que no passado, obtendo uma melhor curva de aprendizado.

Esses paradigmas são, nessa exata sequência, *programação de baixo nível*, *paradigma imperativo*, *programação estruturada*, *programação modular* e *programação orientada a objetos*.

A programação de baixo nível envolve a escrita do código diretamente no hardware, isto é, na parte física. Ler e interpretar um código escrito desse modo e fazer sua manutenção é razoavelmente difícil.

Esse paradigma é então mudado para o paradigma imperativo, no qual as instruções são executadas sequencialmente, isto é, escreve-se, primeiro, o que se quer que a máquina faça primeiro, em seguida, o que se quer que ela faça e seguida e assim sucessivamente pois a máquina, executando um comando em um trecho do código, nunca vai voltar para executar um comando escrito em uma linha anterior.

Vejamos uma estrutura extremamente simples de um código, seguindo esse paradigma, na linguagem Python.

```python
a=5
b=10
s=a+b
print(s)
```

Nas duas primeiras linhas, alocamos espaços de memória, criando, respectivamente, as variáveis a e b, para armazenar os valores dos dados de entrada, que são `5` e `10`. Na linha seguinte fazemos o devido processamento entre os valores assumidos por `a` e por `b`, que são respectivamente `5` e `10` e atribuímos esse valor a variável `s` cujo valor será mostrado na tela, conforme manda o último comando, dando a saída do algoritmo que é a soma entre`5`e `10` dando `15`. Como se pode perceber, pela descrição feita um comando do código é executado após outro, ou seja, terminando um comando, necessariamente é executado o comando da linha seguinte, até se atingir a última linha e se encerrar o programa.

Porém, algumas tarefas precisam repetir mais de uma vez um mesmo passo, por exemplo, e seria muito cansativo e computacionalmente não-conveniente (o código ficaria extenso etc.) digitar, praticamente, o mesmo comando diversas vezes um abaixo do outro.

Para resolver demandas como essa surge o paradigma da programação estruturada, a qual é caracterizada pelo uso de *estruturas de controle e de fluxo* como os *loops*, a exemplo de `while` e `for`, que significam "enquanto" e "para" em português), e condicionais, como `if`, que significa "se" em português, permitindo uma melhor organização ao código.

Nas linguagens de programação atuais, como Python, não é necessário escrever mais do que esse código simples para solicitar que a máquina some dois números, uma vez que o processo de como somar dois números já está embutido na programação da própria linguagem antes de ela ficar disponível para o programador que usá-la para algum fim. Mas só para ilustrar o uso de uma estrutura de controle como `for`, que será útil, em inúmeras outras aplicações vamos repetir o mesmo algoritmo como se precisássemos colocar em código como se soma dois números, partindo do pressuposto, que, ao menos foi embutido na linguagem como somar um número com 1, isto é, tomar o sucessor, uma vez que a soma de dois números naturais ou inteiros, também, é definida a partir da função tomar o sucessor de um número $a$, $b$ quantidades de vezes, se queremos somar $a$ com $b$. Esse é um meio, embora não o único nem, talvez, o mais eficiente de se fazer isso.

Daí viria a necessidade de se usar o comando `for`, para evitar a repetição de praticamente a mesma linha de código $b$ vezes. Toma-se o sucessor e volta-se para linha de cima para tomar o sucessor novamente até encerrar o laço de repetição `for`, que termina na $b$-ésima repetição. Numa aplicação qualquer em linguagem Python, após um acontecimento dessa natureza, o código volta a ser executado linearmente a partir da linha seguinte se não encontrar mas outro laço de repetição ou estrutura de controle, até o código encerrar. Vejamos o código em Python:

```
1 a = 5
2 b = 10
3 s = a
4 for i in range(1, b+1):
5   s=s+1
6 print(s)
```

Lembrando que no código `=` não indica uma igualdade, a qual seria indicada por `==`. Nesse caso `=` indica que a variável à esquerda recebe o valor a direita e o valor recebido pode ser atualizado durante o programa. Na linha 3 `s = a` está escrito que a variável recebe o valor da variável `a`, que é `5`, e dentro do laço de repetição o valor da variável `s` é atualizado `b` vezes, isto é, `10` vezes. Conforme a linha `s = s+1`, cada atualização substitui o valor anterior de `s` pelo seu sucessor, até, no presente caso, se tomar o sucessor `10` vezes a partir de `5`, obtendo, como saída, o valor procurado, que é `15`. O intervalo varia de 1 a $b+1$ pois esse é um aspecto da linguagem Python, que é o fato de que, se queremos fazer uma coisa $b$ vezes a partir da primeira, devemos escrever que o intervalo termina no seu sucessor, no caso $b+1$.

O paradigma seguinte é a programação modular que permite a reutilização de código através da criação de módulos ou bibliotecas. Por exemplo, como já falei, para construir um algoritmo para somar dois números usando Python, eu não preciso reconstruir todo passo a passo do início.

Quem desenvolveu a linguagem Python já fez todo processo de como se soma dois números, basta ao programador, se precisar, para algum fim, dar o comando de soma entre dois valores, conforme fizemos no exemplo da programação linear, que nesse comando já está embutido todos os procedimentos anteriores e, assim, naturalmente, o programa retorna a soma desejada.

É muito comum a existência de uma comunidade de programadores de uma linguagem que criam módulos para uma necessidade específica sua que pode, futuramente, ser utilizado, novamente, pelos mesmos ou outros programadores da linguagem economizando tempo e linhas e mais linhas de código.

Finalmente, temos a *Programação Orientada a Objetos*, na qual o código é organizado em *classes* e *objetos*, permitindo uma modelagem mais efetiva do mundo real. Este paradigma de programação foi idealizado pelo matemático e biólogo Alan Kay, que buscou reunir ideias vindas da biologia com sua abstração matemática a um modo inovador de programar , conforme se pode ver na seguinte frase sua:

> O computador ideal deve funcionar como um organismo vivo, isso é, cada célula se relaciona com outras a fim de alcançar um objetivo, mas cada uma funciona de forma autônoma. As células poderiam também reagrupar-se para resolver um outro problema ou desempenhar outras funções [3]

Em consonância com a analogia feita por Kay acima entre computador e organismos vivos, as variáveis são *objetos* independentes com seus *atributos* ( características do *objeto*) e *métodos* (o que se pode fazer com esse *objetos*), analogamente a objetos físicos, mas que compartilham embora tais *atributos* e *métodos* com outros objetos (embora nem sempre os mesmos valores assumidos por tais atributos e métodos), reunidos em uma *classe*, que é o tipo do *objeto* ou dado. Outras *classes* tem *objetos* que, também, podem se relacionar pelos *métodos* e *atributos* que a definem com os *métodos* da primeira *classe* e vice-versa, como as células de um órgão ou os órgãos de um sistema que, embora independentes entre si, necessariamente, se comunicam entre si para o funcionamento de todo sistema ou organismo vivo. Analogamente é o comportamento das *classes* e seus respectivos*objetos* na contribuição do funcionamento de um *sistema* ou *algoritmo computacional* escrito conforme o paradigma de *Programação Orientada a Objetos*.

# Capítulo 3

# Semiótica, a Aritmética dos Números Naturais e o Produto Educacional

A Teoria dos Registros de Representação Semiótica é profundamente relevante no ensino de aritmética na educação básica pois existem diversas maneiras de se trabalhar esses conceitos, além da representação simbólico e da língua natural, como diagramas e outras, muitas delas pictóricas como o famoso uso de palitinhos e as operações que podemos fazer a eles como adicionar, retirar e agrupá-los.

As animações do Produto Educacional sobre aritmética, que irão ilustrar a definição de adição e suas propriedades, operam com uma ideia bastante similar. Uma abordagem que leve isso em conta pode beneficiar significativamente os alunos, ao permitir que eles compreendam os conceitos aritméticos sob várias perspectivas.

Trabalhar com varias representações semióticas dos objeto e conceitos aritméticos tende a promover nos alunos um entendimento mais flexível e profundo deles uma vez que, como já foi dito no capitulo 1, cada representação de um objeto traz à tona diferentes aspectos deste ente que não são bem contemplados em outra, com uma completando as deficiências de representação da outra e, de fato, é assim no estudo da aritmética.

De fato, não haveria como ser diferente pois nenhuma dessas representações é o objeto em si mas simplesmente uma abstração mais ou menos aproximada do mesmo, portanto, não tem a possibilidade de contemplar todos os seus aspectos.

A comutatividade e a associatividade são propriedades da operação de adição definida sobre o conjunto dos números complexos. Sendo cada um dos demais conjuntos numéricos, algebricamente, uma cópia de um subconjunto dos complexos mas, podemos considerar, a grosso modo, um subconjunto dele , e essas mesmas propriedades valem para a adição feita entre números naturais, conforme veremos abaixo.

Porém, a ordem usual da construção dos conjuntos numéricos nas estruturas algébricas é ver essas e todas as demais propriedades operatórias (de início pelas operações de adição e multiplicação) dos números naturais, como consequência lógica dos axiomas de Peano que caracterizam e definem uma classe de conjuntos da qual os números naturais é o menor relativamente à relação de inclusão de conjuntos.

Vamos, então, enunciar esse sistema de axiomas, definir o que é adição e o que é multiplicação nesse conjunto e exibir as propriedades da adição e multiplicação desse conjunto, além de demonstrar as propriedades da comutatividade e da associatividade da adição, que serão ilustradas entre as animações, e a distributividade da multiplicação em relação adição, também, o que pode ser mostrado na quarta animação, a depender do olhar do aluno e do professor mas, por outro lado, essa animação pode mostrar a própria definição do que é a multiplicação no conjunto dos números naturais.

Observação importante é que, nos dois casos, essa ilustração é feita somente nas passagens em que há adição e, portanto, acréscimo de fileiras, conforme veremos, quando falarmos da animação, pois a subtração não é uma operação sobre o conjunto dos números naturais. Mas como a proposta é mais geral, o propósito é ajudar o aluno a entender a propriedade distributiva no conjunto dos números reais, onde faz sentido falar sobre a subtração e defini-la como uma operação. O que interessa aqui não é o número em si mas a forma, isto é, a estrutura algébrica que permeia relações operacionais entre números reais quaisquer.

Embora estejamos mostrando propriedades numéricas, utilizando como exemplos números naturais, aprender estes exemplos permite ao aluno estender seu aprendizado para manipular números reais, de modo geral, pois a comutatividade e associatividade (da adição e também da multiplicação) e a distributividade da multiplicação em relação a adição são conservadas em qualquer estrutura algébrica de um subconjunto clássico dos números reais (naturais, inteiros, racionais e os próprios reais) com suas respectivas adição e multiplicação usuais, que são as estruturas que os alunos terão contato, em um contexto ou outro, durante toda a educação básica e superior, principalmente para quem cursar áreas com intenso uso da matemática (tradicionalmente chamadas de ciências exatas).

E se considerarmos os números complexos, que, de modo geral, são introduzidos na 3° série do ensino médio, ocorre a mesma preservação dessas propriedades em sua estrutura algébrica com as operações de adição e multiplicação, que foram introduzidas nesta série.

Vejamos, agora, os axiomas de Peano, que caracterizam formalmente uma classe conjuntos da qual o conjunto $\mathbb{N}$ dos números naturais é o menor, em termos da relação de inclusão de conjuntos:

### Axiomas de Peano

1. Existe uma função injetiva $s : \mathbb{N} \to \mathbb{N}$, dado $n$, $s(n)$ é chamado *sucessor* de $n$.

2. Existe um único número 1, que não é sucessor de nenhum outro.

3. Se $X \subset \mathbb{N}$ é tal que 1 pertence a $X$ e $s(X) \subset X$, então $X = \mathbb{N}$.

Esse último axioma é chamado Princípio de Indução e podemos nos valer dele quando queremos mostrar que um fato vale dentro do conjunto dos números naturais e será o método usado aqui nas demonstrações das propriedades comutativa e associativa da adição e distributiva, podendo também ser naturalmente usado para demonstrar as demais propriedades.

Agora, serão definidas, respectivamente, adição e multiplicação no conjunto dos números naturais. Além de ser usado como método de demonstração no conjunto $\mathbb{N}$, o princípio de indução, também, pode ser utilizado para definir recursivamente tais operações e é o que será feito a seguir.

A operação de adição é denotada pelo símbolo $+$. Fixado um número natural $m$ é um número $n$ natural qualquer,

1. $m + 1 := s(m)$

2. $m + s(n) := s(m + n)$

Essa definição, do tipo recursiva, está bem definida, uma vez que se comporta como um algoritmo (ou um procedimento) que permite calcular a soma de dois números naturais quaisquer. Ela define, para cada número natural $m$ fixado, a soma do mesmo com todos os números naturais. Isso pode ser garantido pelo princípio de indução, pois, primeiramente, ele define a soma de $m$ com 1 e, em seguida, partindo da hipótese que sabemos somar $m$ com um dado $n \in \mathbb{N}$, saberemos também somar $m$ com seu sucessor, que é o mesmo que tomar o sucessor de $m + n$, soma que supomos já conhecer, ao colocar seu sucessor após a igualdade. Cumprindo as duas condições, pelo princípio de indução, o subconjunto de $\mathbb{N}$ ao qual pertencem os valores $n$ cuja soma de $m$ com o mesmo foi definida é o próprio conjunto dos números naturais, isto é, foi ensinado a somar cada número natural com cada outro número natural ou consigo mesmo.

Está é uma definição do tipo recursiva, pois ela instrui que para calcular a soma de $m$ com certo número natural $n$ recorre-se a soma do mesmo com o seu antecessor, já determinada. E do mesmo modo, podemos definir a multiplicação no conjunto dos números naturais, o que pressupõe que se sabe somar dois números naturais, uma vez que tal operação usa o conceito de adição em sua definição, em sua sentença recursiva, como pode-se ver abaixo.

Fixado um número natural $m$ é um número natural $n$ qualquer

1. $m \times 1 := m$

2. $m \times s(n) := m \times n + m$

Do mesmo modo da adição, pelo Princípio de Indução, essa operação está bem definida, uma vez que é ensinado a multiplicar $m$ por 1 e se sabemos multiplicar $m$ por um número qualquer, sabemos multiplicar $m$ pelo sucessor de tal número. Portanto tal procedimento orienta a fazer a multiplicação entre dois números naturais.

Com a aplicação do Princípio de Indução como método de demonstração, chega-se às propriedades abaixo, das quais, conforme já foi prometido, serão feitas as demonstrações de três destas.

Seguem as propriedades: Dados $m$, $n$, $p \in \mathbb{N}$, temos

**Comutatividade da adição**

$$m + n = n + m$$

**Associatividade da adição**

$$(m + n) + p = m + (n + p)$$

**Comutatividade da multiplicação**

$$m \times n = n \times m$$

**Associatividade da multiplicação**

$$m \times (n \times p) = (m \times n) \times p$$

**Distributividade da multiplicação em relação à adição**

$$m \times (n + p) = m \times n + m \times p$$

Demonstremos a propriedade comutativa da adição:

Antes disso provemos dois resultados importantes, que saem como consequência da definição da operação de adição sobre os números naturais, os quais serão utilizados como argumentos na demonstração dessas propriedades. Seguem os resultados abaixo: $(a+b)+1 = a+(b+1)$, para $a$, $b$, sendo $a$ e $b$ números naturais e, além disso, $a+1 = 1+a$, para todo número natural $a$.

Provemos o primeiro dos resultados acima:

i Se $b = 1$, então

$$(a + b) + 1 = (a + 1) + 1 = s(a + 1) = a + s(1) = a + (1 + 1) = a + (b + 1).$$

ii Dado um número natural $b$, suponhamos que $(a+b)+1 = a+(b+1)$, então:

$$(a+s(b))+1 = s(a+b)+1 = s(s(a+b)) = s(a+s(b)) = a+s(s(b)) = a+(s(b)+1).$$

Portanto, de (i) e (ii), pelo Princípio de Indução, vale $(a+b)+1 = a+(b+1)$, para todo par a, b de números naturais.

Provemos agora o segundo resultado:

i Se $a = 1$ , então $a+1 = 1+1 = 1+a$.

ii Dado um número natural $a$, suponhamos que $a+1 = 1+a$, então

$$s(a)+1 = (a+1)+1 = (1+a)+1 = s(1+a) = 1+s(a).$$

Portanto, pelo Princípio de indução $a+1 = 1+a$, para todo número natural $a$.

Munidos desses dois resultados, agora, podemos provar as propriedades comutativa e associativa da adição no conjunto dos números naturais.

Provemos, primeiro, a comutatividade, isto é, que $a+b = b+a$, para todo par de números naturais $a$ e $b$.

Demonstração:

Fixado um número natural $a$, dado um número natural $b$, temos

i Se $b = 1$, então $a+b = a+1 = 1+a = b+a$.

ii Dado um par de números naturais, $a$ e $b$, suponhamos que $a+b = b+a$, então

$$\begin{aligned} a+s(b) &= s(a+b) \\ &= s(b+a) \\ &= (b+a)+1 \\ &= (1+b)+a \\ &= (b+1)+a \\ &= s(b)+a. \end{aligned}$$

Portanto, de (i) e (ii), pelo Princípio de Indução, vale a comutatividade na adição, conforme foi definida, sobre o conjunto dos números naturais, isto é, $a+b = b+a$, para todo par $a$ e $b$ de números naturais.

Provemos, agora, a associatividade da adição, isto é,

$$(a+b)+c = a+(b+c),$$

para toda tripla $a$, $b$ e $c$ de números naturais.

i Se $c = 1$, então $(a + b) + c = (a + b) + 1 = a + (b + 1) = a + (b + c)$.

ii Dado um número natural $c$, suponhamos que $(a + b) + c = a + (b + c)$, então

$$\begin{aligned}(a + b) + s(c) &= s((a + b) + c)\\ &= s(a + (b + c))\\ &= a + s(b + c)\\ &= a + ((b + c) + 1)\\ &= a + ((b + c) + 1)\\ &= a + (b + (c + 1))\\ &= a + (b + s(c)).\end{aligned}$$

Portanto, de (i) e (ii), pelo Princípio de Indução, vale a propriedade associativa dos números naturais, isto é,

$$(a + b) + c = a + (b + c),$$

para toda tripla $a$, $b$ e $c$ de números naturais.

Antes de discorrer sobre as animações do Produto Educacional, será feita uma breve descrição da biblioteca Manim, usada na construção de tais animações.

## 3.1 A Biblioteca Manim

Manim é uma biblioteca da linguagem Python que oferece recursos de programação para fazer animações de objetos e fatos matemáticos ou científicos, com base em conceitos fundamentais como *cena*, *objeto* e *animação*. Portanto, ele depende fortemente do paradigma de *Programação Orientada a Objetos* para suas construções.

A *cena* é onde a animação vai acontecer. É em si um *objeto* no paradigma de *Programação Orientada a Objetos*. A *classe* na qual ele está inserido tem os seguintes *métodos*: adicionar objetos, ativar animações, pausar, adicionar objetos etc. Ou seja, são coisas que podem ser feitas com a cena. Para compor uma cena, precisamos adicionar objetos a ela. Existem vários tipos de objetos, como quadrados e círculos, textos. E as *cenas* são o resultado da aplicação desses métodos a tais objetos.

Existem métodos de execução aplicados à *cena* e outros aplicados a objetos contidos na cena. Seguem alguns dos métodos aplicados a estes últimos objetos, junto com o que cada um faz.

`shift()`: Move o objeto para uma nova posição no espaço.

`rotate()`: Gira o objeto em torno do seu próprio centro ou de um ponto específico.

`scale()`: Escala o objeto, aumentando ou diminuindo seu tamanho.

`to_corner()` e `to_edge()`: Posicionam o objeto nos cantos ou bordas da tela.

`next_to()`: Posiciona o objeto próximo a outro objeto.

`align_to()`: Alinha o objeto com outro objeto em relação a um eixo específico.

`set_color()`: Define a cor do objeto.

`set_opacity()`: Define a opacidade do objeto.

`add()`: Adiciona um objeto à cena.

`remove()`: Remove um objeto da cena.

`animate()`: Inicia uma animação para o objeto. Pode ser usado para animar propriedades do objeto, como posição, escala, cor etc.

`clear_updaters()`: Remove todas as animações associadas a um objeto.

`add_updater()`: Adiciona uma função de atualização personalizada que modifica as propriedades do objeto ao longo do tempo.

`shift()`: Move o objeto para uma nova posição no espaço.

`rotate()`: Gira o objeto em torno do seu próprio centro ou de um ponto específico.

`scale()`: Escala o objeto, aumentando ou diminuindo seu tamanho.

`flip()`: Inverte o objeto horizontal ou verticalmente.

`next_to()`: Posiciona o objeto próximo a outro objeto.

`align_to()`: Alinha o objeto com outro objeto em relação a um eixo específico.

`set_stroke()`: Define as características do traço do objeto, como cor, largura e tipo de linha.

`set_fill()`: Define as características de preenchimento do objeto, como cor e opacidade.

`shift_and_rotate()`: Move e gira o objeto simultaneamente.

## 3.2 Descrição de uma Animação

Será tratada aqui, primeiro, a animação construída para fazer uma representação pictórica da propriedade comutativa.

No início animação, temos, ao lado esquerdo uma fileira horizontal de $m$ quadradinhos de uma determinada cor do lado esquerdo e uma fileira horizontal de $n$ quadradinhos de uma outra cor do lado direito (`m` e `n` são escolhidos aleatoriamente a cada execução do programa). Vamos supor que foi sorteado `m = 3` e `n = 2`. Os $m = 3$ quadradinhos se movem, então, em direção aos $n = 2$ quadradinhos, atravessando-os e passando para o outro lado. Agora, temos $n = 2$ quadradinhos no lado esquerdo e $m = 3$ quadradinhos do lado direito e, mesmo com essa mudança de configuração, ainda temos, juntando os quadradinhos, 5 quadradinhos mostrando que $3 + 2 = 2 + 3$, isto é, $m + n = n + m$, um fato mais geral, que pode se tornar cada vez mais evidente para o aluno a o executar mais vezes o programa, com outros valores.

Vê-se nas figuras 3.1, 3.2, 3.3, 3.4 uma sequência de cenas da animação feita.

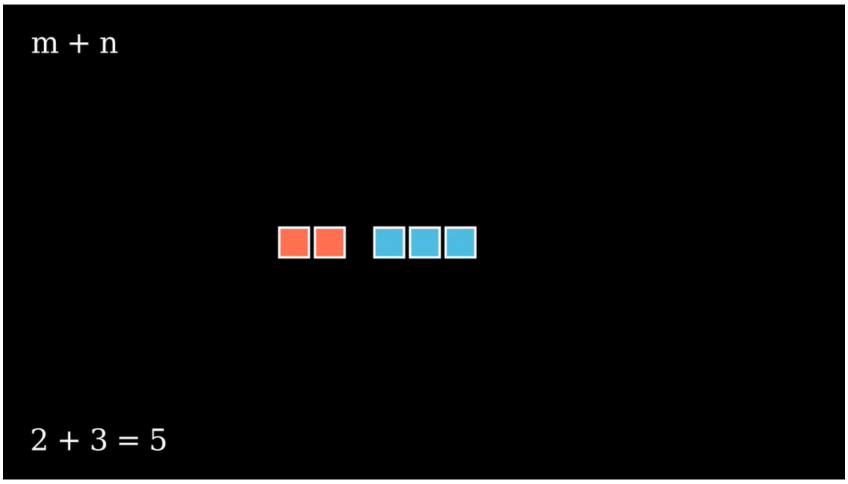


Figura 3.1

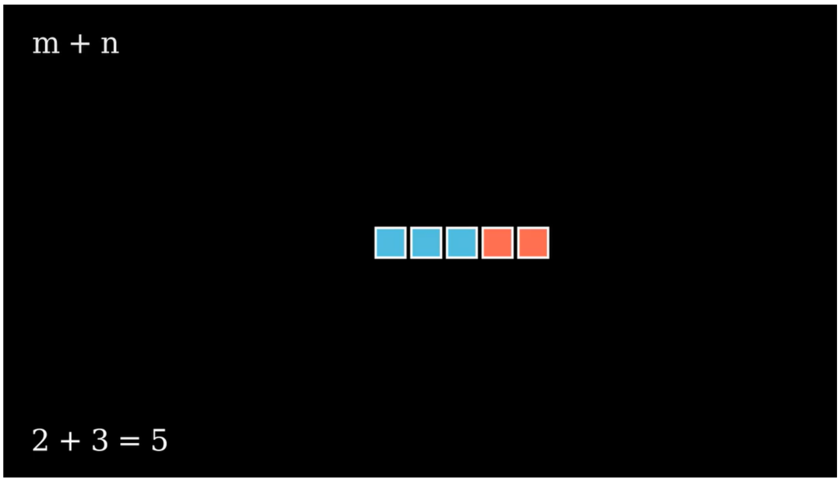


Figura 3.2

Vamos analisar a segunda animação, na qual é ilustrada a multiplicação entre dois números naturais à luz da Teoria de Registros de Representação Semiótica, de Raymond Duval.

No canto superior esquerdo da cena, temos o produto entre 9 e 7 em diferentes formas dentro do registro simbólico, sendo que a igualdade entre uma forma e a seguinte indica o tratamento feito para que, de uma, se obtenha a outra e, como já foi citado no capítulo 1, diferentes formas dentro de um mesmo registro sugerem diferentes ênfases semânticas dada a tal objeto. Por exemplo, `9*7`, indica que o produto entre 9 e 7 é a quantidade de unidades de 7 parcelas de 9 unidades cada o que, no mesmo momento, é mostrado em sua representação pictórica, dentro da cena, como 7 fileiras (correspondentes

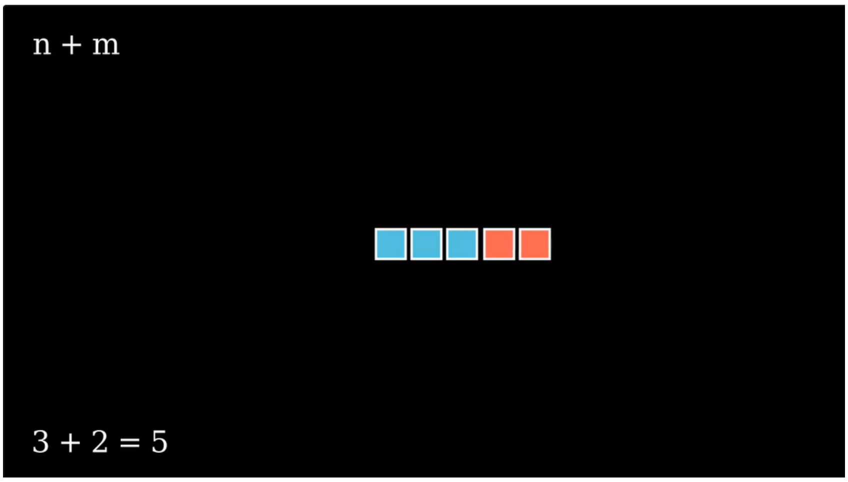


Figura 3.3

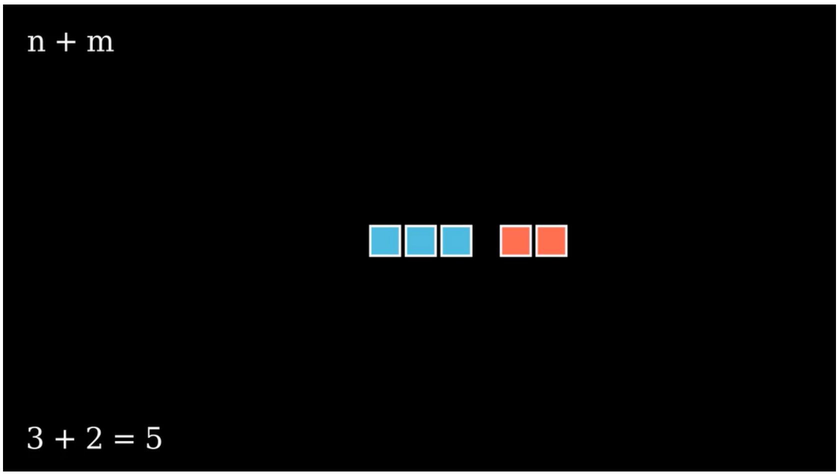


Figura 3.4

às parcelas) de 9 quadradinhos (correspondentes às unidades). E fazendo o tratamento para `9*6+9` que corresponde, também a $54 + 9$, dá se ênfase ao fato que a quantidade que é esse produto é obtida recursivamente a partir do produto da quantidade anterior de parcelas/fileiras (6) com a quantidade de unidades/quadradinhos por fileira (9) acrescido de mais uma parcela/fileira com 9 unidades/quadradinhos e `9*7`, por sua vez, foi obtida do mesmo modo. Isso é ilustrado, dinamicamente, na animação, pois ela faz um acréscimo gradativo de fileiras sobre a anterior que é acompanhado com a indicação da ação correspondente em sua representação simbólica.

Essa animação ilustra bem a lógica por trás do que é fazer uma adição, baseada na sua definição, que foi feita anteriormente. O objetivo é que observando e entendendo o padrão descrito na animação, o aluno não simplesmente saiba os resultados da tabuada, de modo literal, mas faça uma operação mental baseada na lógica por trás da operação para obter a soma, o que está de acordo com a citação abaixo que aponta a natureza do conhecimento aritmético. A citação abaixo define o que seria um conhecimento de tal natureza.

> Já o conhecimento lógico-matemático consiste em relações criadas individualmente. Ao contrário do conhecimento físico, sua origem está na mente de cada um. De acordo com Kamii (1995), as crianças elaboram esse conhecimento à medida que constroem relações mais complexas sobre outras mais simples que elas mesmas criaram. [15]

Reforçando essa ideia está a explanação de Jordan Ellenberg, em seu livro *O poder do pensamento matemático: a ciência de como não está errado*, que mostra como é fácil ter uma razoável intuição da propriedade comutativa da multiplicação, por exemplo, por observações no nível concreto ou até mesmo pictórico percebendo uma relação lógica, mesmo que ainda não num nível suficientemente formal, que o aluno só vai começar a ter no $7^o$ ano do ensino fundamental.

Nesse sentido, ele conta no livro sua experiência de que, deitado no chão da sala de seus pais ainda criança, ele notou que poderia notar o arranjo retangular de furos no de painel de madeira no qual estava embutido o aparelho estéreo, tanto como 8 fileiras (verticais) de 6 furos ou 6 fileiras (horizontais) de 8 furos. Considerando-a 6 por 8 ou 8 por 6, obteria os mesmos 48 furos que formam o único arranjo que foi analisado, embora de dois modos distintos, ilustrando, portanto, o fato mais geral que subjaz isso, que é a comutatividade da multiplicação no conjunto dos números naturais.

Nesse sentido, é válido citar o seguinte trecho:

> Temos uma tendência de ensinar matemática como uma longa lista de regras. Você as aprende numa ordem e deve obedecê-las, caso contrário tira nota baixa. Isso não é matemática. Matemática é o estudo de coisas que aparecem de certo modo porque não poderiam ser de modo diferente. [9]

Podemos colocar como essas listas de regras como um estudo da tabuada, baseada em memorização, do resultado de cada operação entre dois números a despeito de uma condução a aquele resultado por algum mecanismo que evidencie que aquele resultado não poderia ser, logicamente, diferente, como ele mesmo (o autor) pode notar que $8 \times 6$ não poderia ser diferente de $6 \times 8$, pela análise lógica possível ao seu desenvolvimento cognitivo (na época ele ainda era uma criança de 6 anos) de uma situação do mundo concreto.

Porém, ele faz uma ressalva de que fazer isso nem sempre é algo suficientemente espontâneo e imediato mas, mesmo assim, partiu de outras coisas mais imediatas ao senso comum que foi chegando a esse nível de complexidade, como veremos abaixo:

> Agora, sejamos justos. Nem tudo na matemática pode ser perfeitamente transparente para nossa intuição como a adição e a multiplicação. Você não pode

> fazer cálculo diferencial e integral pelo senso comum. Mas ainda assim o cálculo derivou do nosso senso comum — Newton pegou nossa intuição física sobre objetos que se movem em linha reta, formalizou-a e construiu sobre essa estrutura formal uma descrição matemática universal do movimento. [9]

A ideia é que o aluno, ao pegar a lógica por trás da soma de dois números naturais pequenos, no caso da animação, de 1 a até 10, possa estender o raciocínio para números na casa das centenas, daí para casa dos milhares, até generalizar a ideia completamente em sua cognição, isto é, saber aplicá-la em qualquer situação, para pares de números de quaisquer magnitudes.

A situação acima é análoga ao que ocorre na primeira animação, que ilustra como se faz adição. Mas no caso da animação da adição, conforme vimos em exemplos e na definição de adição , a ideia de somar $m$ com $n$ é aplicar $n$ vezes seguida a função sucessor a partir de $m$. Por exemplo, a soma ente 5 e 3 pode ser interpretada como acrescentar 3 unidades a 5 unidades ou aplicar 3 vezes seguidas, começando pelo 5 a função sucessor a um número, o que é indicado pelas duas distintas representações simbólicas mostradas em cada etapa da cena.

A mesma correspondência entre os elementos significativos da representação simbólica e a representação pictórica da qual foi explanada a existência no parágrafo anterior a respeito da segunda animação também está presente nesse. No caso $m$ é indicada pela fileira, que aparece inicialmente à esquerda com $m$ quadradinhos o sinal de adição é indicada pelos acréscimos de $n$ quadradinhos, que são feitos um a um gradativamente, sendo que o acréscimo de cada quadradinho representa aplicação uma vez da função sucessor, correspondendo,de acordo com a definição, da soma representada simbolicamente por $m + n$.

Este conceito de correspondência é de notável importância na teoria de Duval, e está inserido num conceito mais geral que é o conceito de congruência, conforme pode ser notado na citação abaixo:

> São três os critérios para indicar se duas expressões são semanticamente congruentes (DUVAL, 2004b, p. 79 - 93): CRITÉRIO 1: Correspondência entre as unidades significativas própria a cada registro; CRITÉRIO 2: Univocidade para cada unidade significativa a ser convertida. Para uma unidade de partida, não há mais de uma unidade (significativa) possível no registro de chegada; CRITÉRIO 3: A ordem na organização das unidades significativas de partida é conservada na representação de chegada. [2]

Além da *correspondência semântica*, a *univocidade semântica* está evidentemente presente uma vez que não há qualquer ambiguidade, dúvida, de se certa representação simbólica corresponde a essa ou aquela soma ou produto e vice-versa. As duas representações foram construídas de modo a representar, exatamente, um único produto ou única soma, a partir de uma organização rigorosa dos seus elementos significativos o que, também, é reflexo da própria ordem de organização das unidades compondo cada uma das duas representações.

Por exemplo, pra somar 5 com 3, colocou-se os números, nessa exata ordem em uma de suas representações simbólicas e, de modo correspondente, em sua representação pictórica a fileira de $m$ quadradinhos está à esquerda, o acréscimo gradativos de $n$ quadradinhos à esquerda da a a noção de uma soma de $m$ com $n$ e não de $n$ com $m$ ou de qualquer outra soma.

Perceber essa congruência entre as duas representações permite ao aluno estar de posse do essencial na atividade matemática, que não está nas representações em si mas nas transformações de uma para a outra, conforme a citação abaixo.

> Segundo Duval (2011, p.68), "o que é matematicamente essencial em uma representação semiótica são as transformações que se podem fazer, e não a própria representação. Essas representações também precisam ser decodificadas, conforme o conhecimento prévio que se tem delas ou de processos análogos, como os de transformações de registros." [14]

As figuras 3.5, 3.6, 3.7, 3.8 mostram, nessa ordem, uma sequência de cenas da animação feita, baseada na teoria dos registros de representação semiótica e na definição de adição de número naturais para ilustrar esta ideia (adição sobre números naturais).

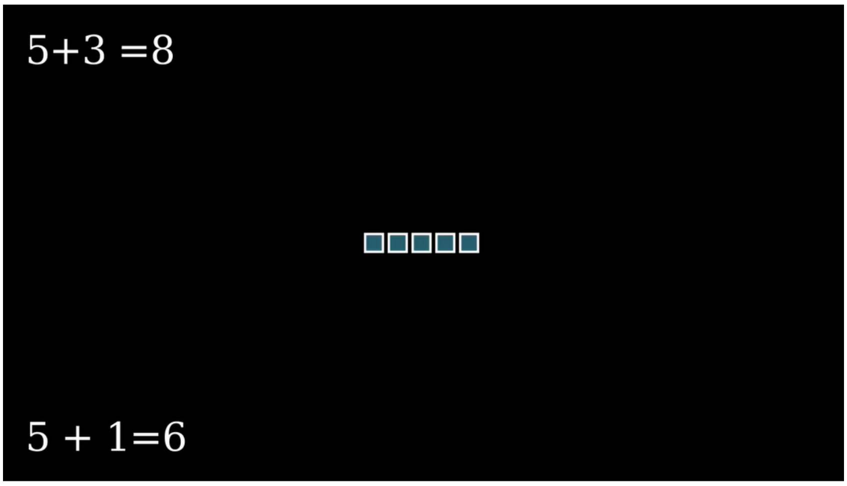


Figura 3.5

Já as figuras 3.9, 3.10, 3.11, 3.12, 3.13, 3.14 e 3.15 formam, nessa ordem, cenas do Produto Educacional que ilustram a operação de multiplicação sobre o conjunto dos números naturais.

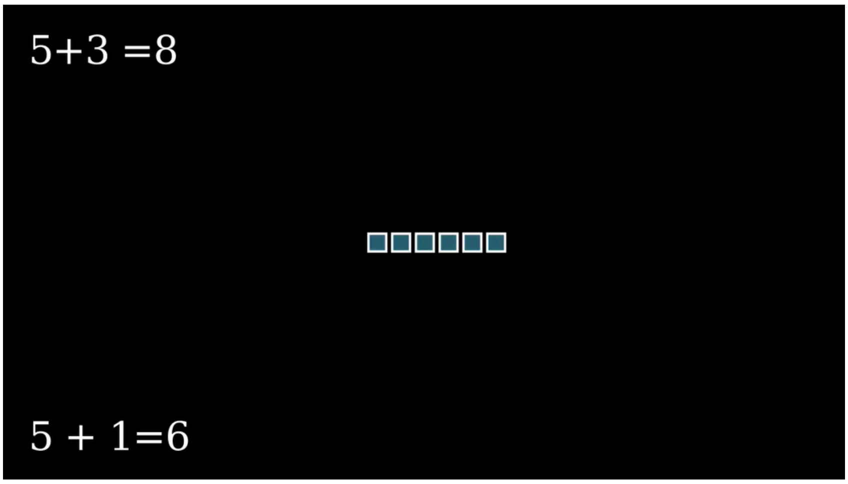


Figura 3.6

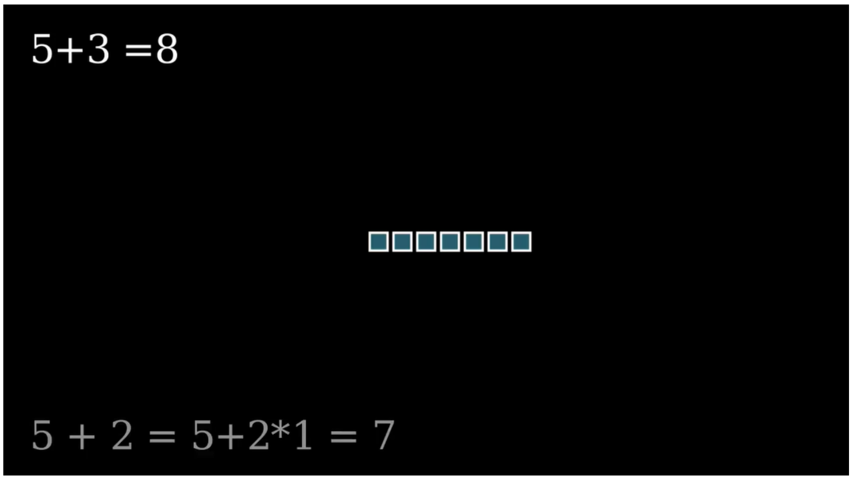


Figura 3.7

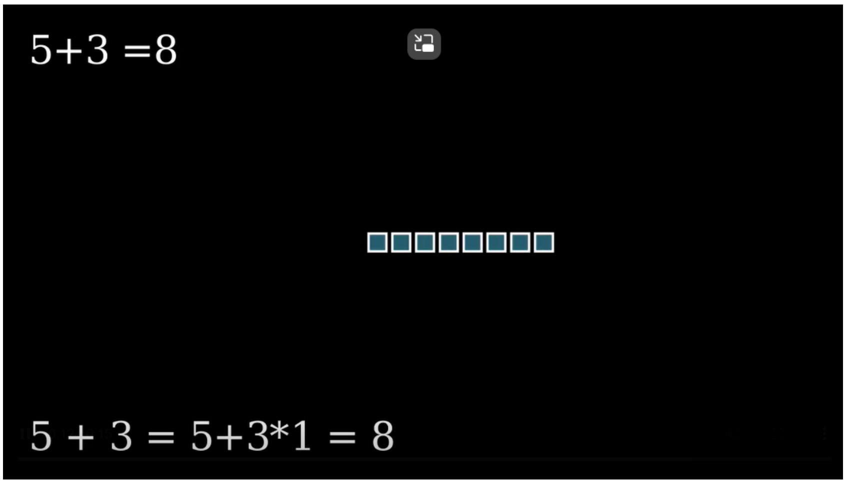


Figura 3.8

Conforme já foi demonstrado, a adição tem a propriedade comutativa. Vejamos, na sessão seguinte, uma sequência de cenas da animação que ilustra essa ideia, também

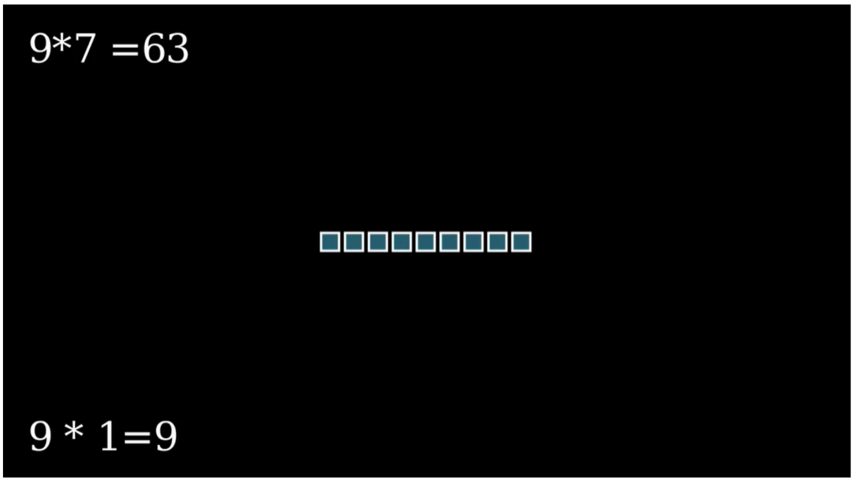


Figura 3.9

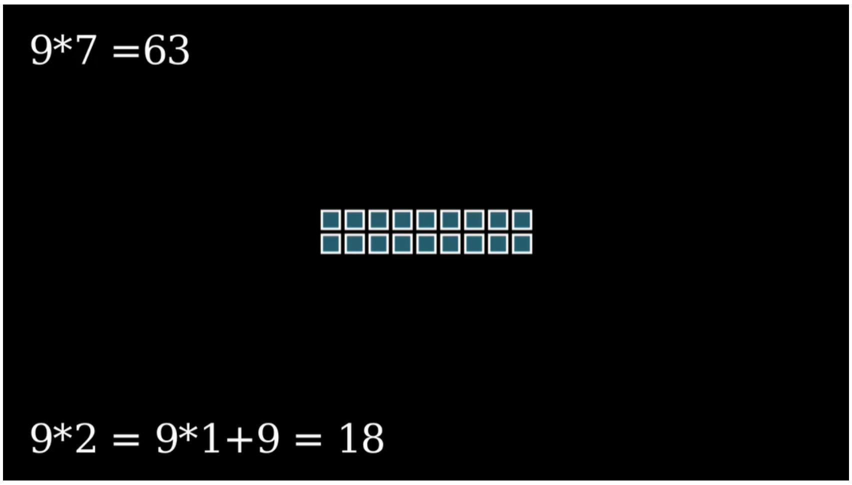


Figura 3.10

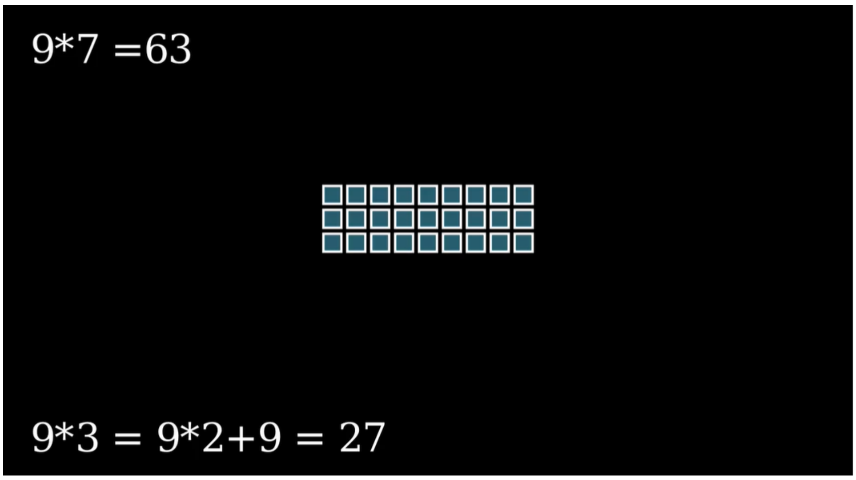


Figura 3.11

baseada na teoria dos registros de representação semiótica:

As figuras 3.1, 3.2, 3.3, 3.4 formam uma sequência de cenas da animação do Produto

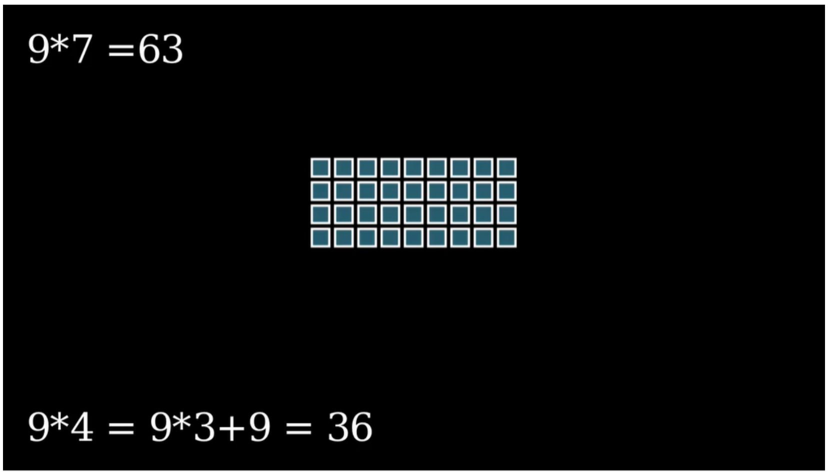


Figura 3.12

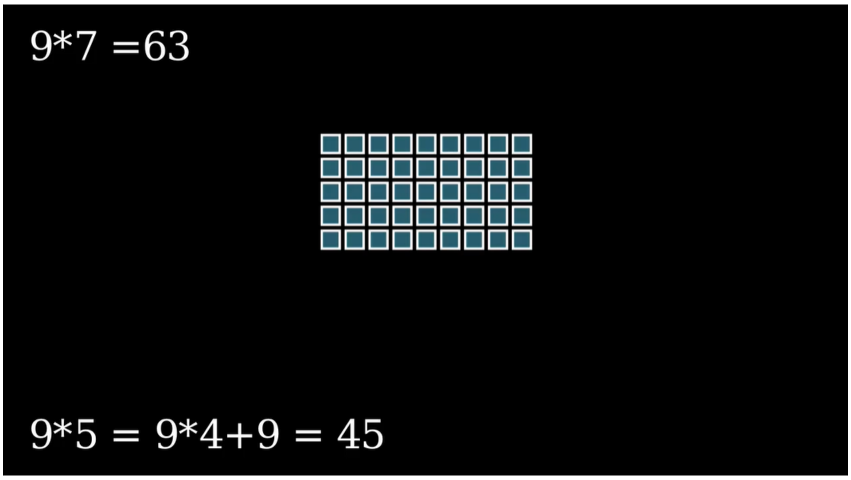


Figura 3.13

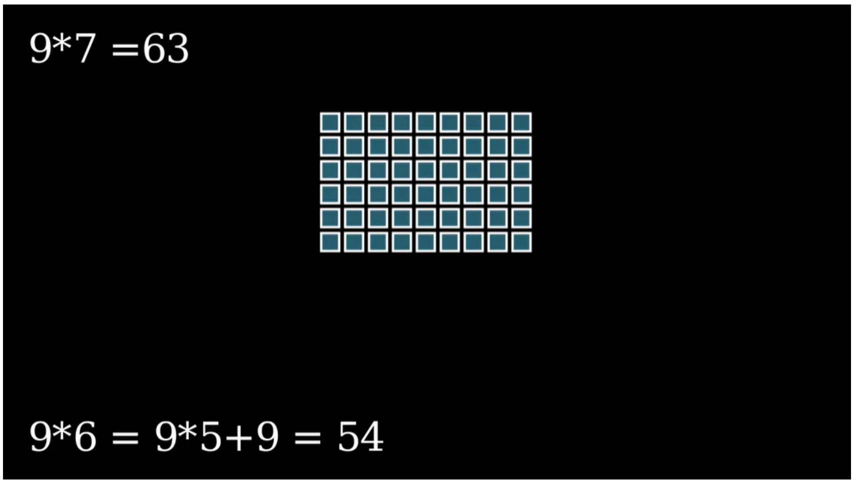


Figura 3.14

Educacional que ilustra essa ideia.

As figuras 3.16, 3.17, 3.18, 3.19, 3.20 formam uma sequência mostrando a animação

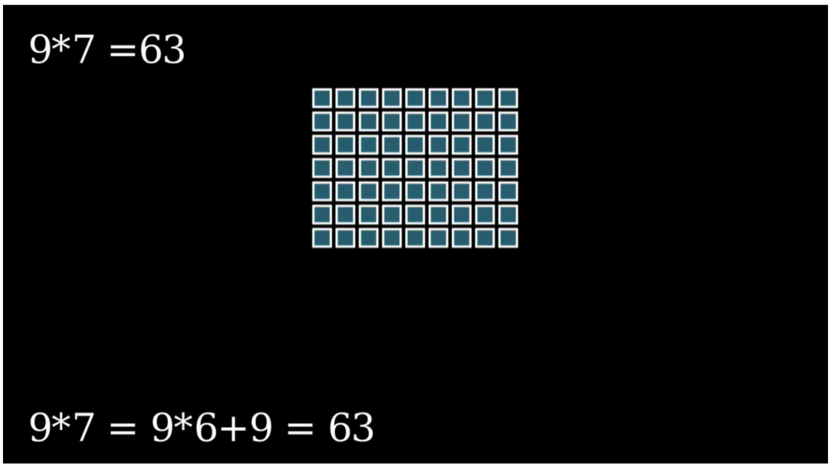


Figura 3.15

que ilustra, no Produto Educacional, a propriedade associativa da adição dos números naturais, já demonstrada aqui.

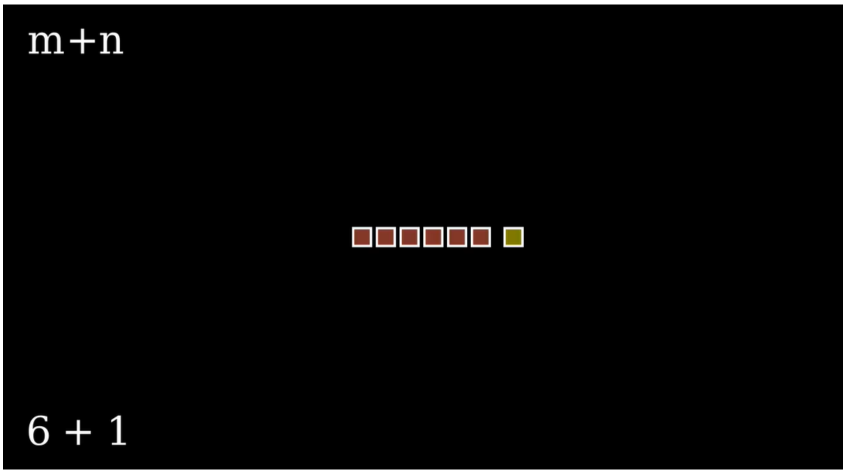


Figura 3.16

## 3.3 Um exemplo com expressão algébrica no Produto Educacional

As figuras 3.21, 3.22, 3.23, 3.24, 3.25, 3.26, 3.27 mostram, nessa ordem, uma sequência de cenas da animação envolvendo a ideia de monômios e das operações de adição e subtração de monômios de mesma parte literal.

Foi instanciado o valor `7` para a indeterminada `n`. O recomendado é que o professor mostre a sua turma a mesma animação com a indeterminada `n` assumindo outros valores, para que, indutivamente, o aluno possa generalizar as propriedades envolvendo as expressões na indeterminada `n` trabalhadas nas animações.

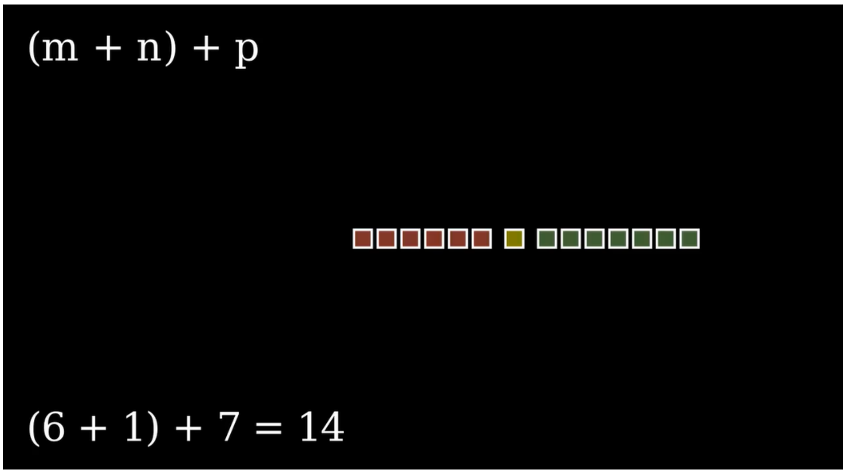


Figura 3.17

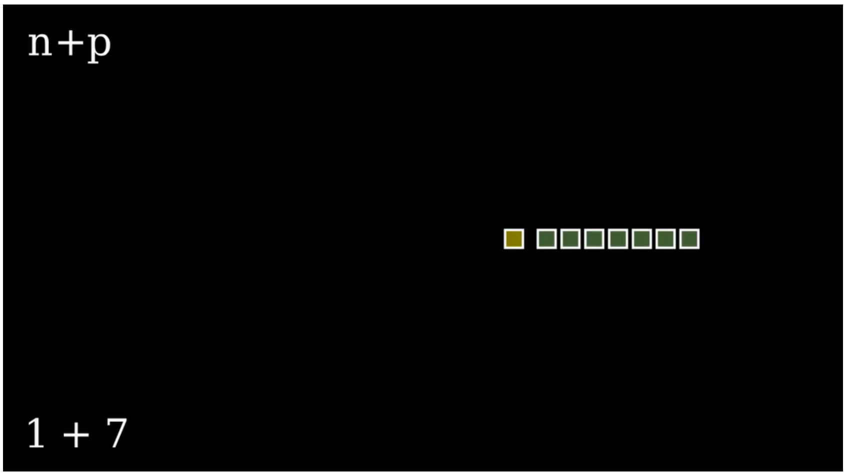


Figura 3.18

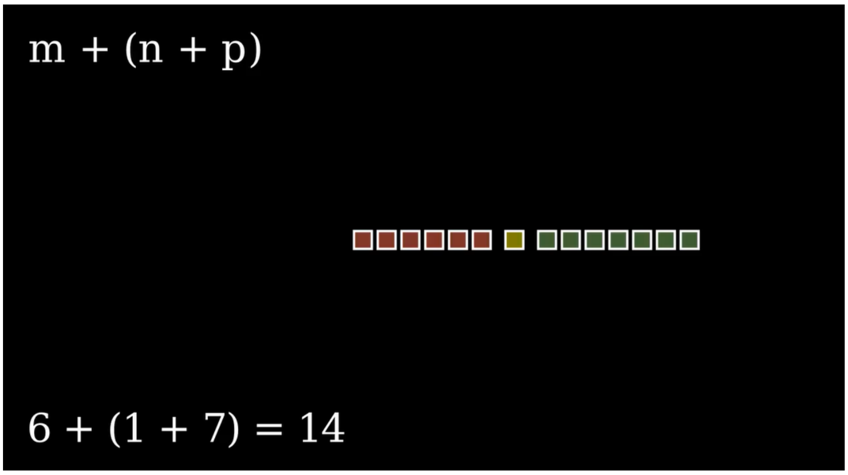


Figura 3.19

As animações exibem, a cada nova representação gráfica, em fileiras de quadradinhos, simultaneamente, no canto superior esquerdo da tela o tratamento que foi feito a

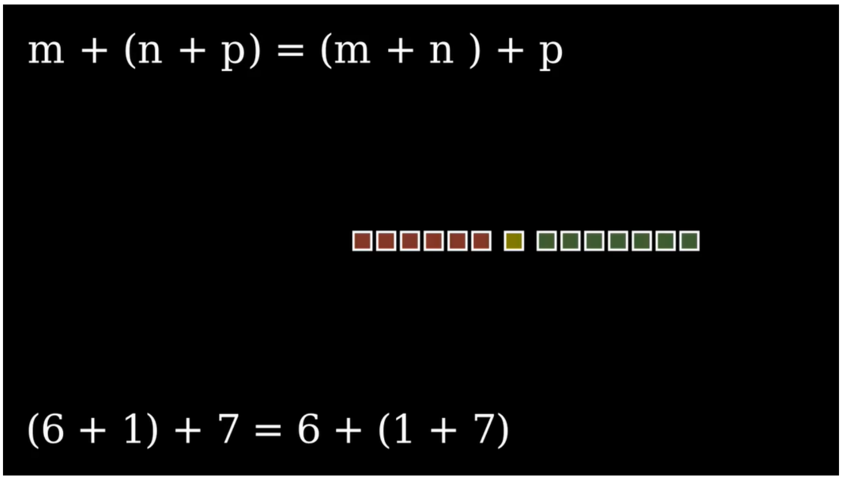


Figura 3.20

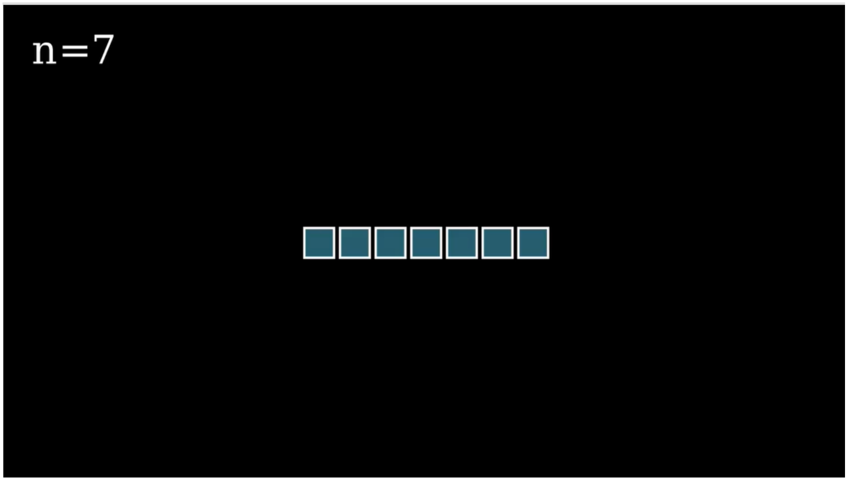


Figura 3.21

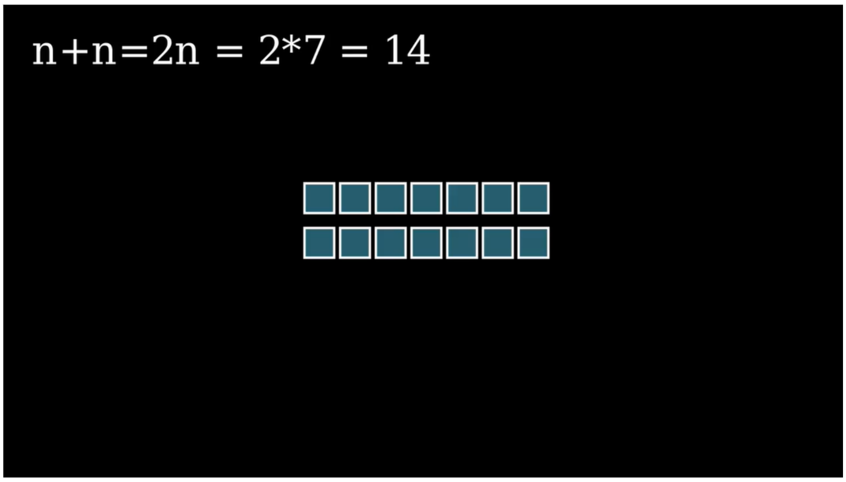


Figura 3.22

partir da expressão anterior para chegar à expressão atual e iguala à expressão atual e em seguida, calcula seu valor numérico, no caso para `n = 7`.

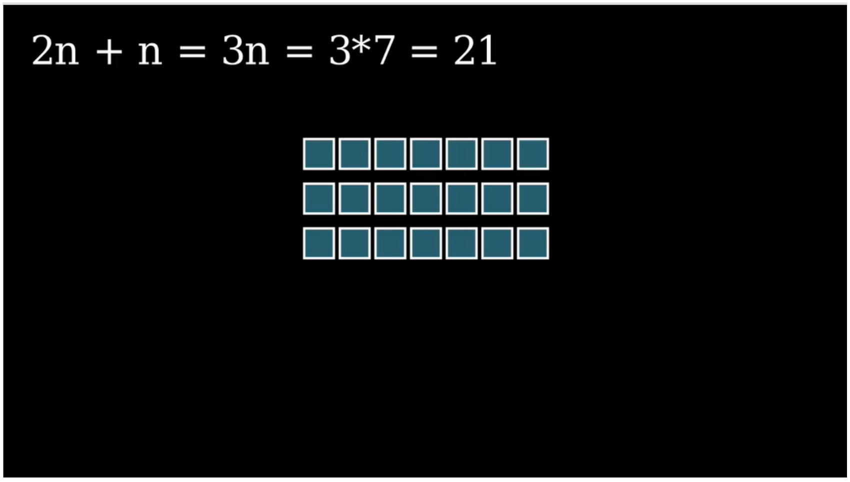


Figura 3.23

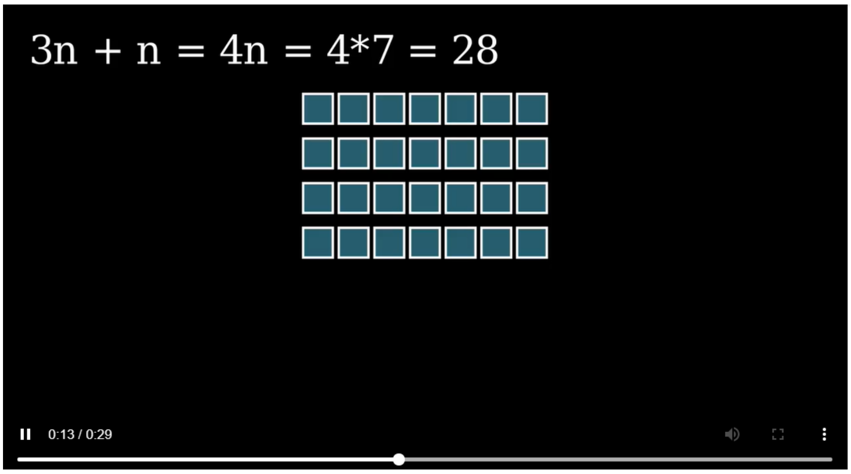


Figura 3.24

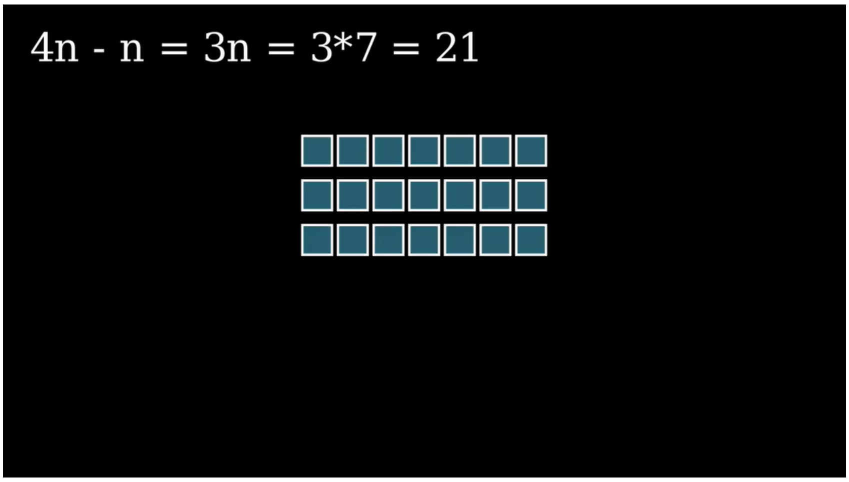


Figura 3.25

Aí são trabalhados três aspectos importantes do estudo de álgebra na segunda etapa do ensino fundamental, que é compreender como fazer corretamente operações entre

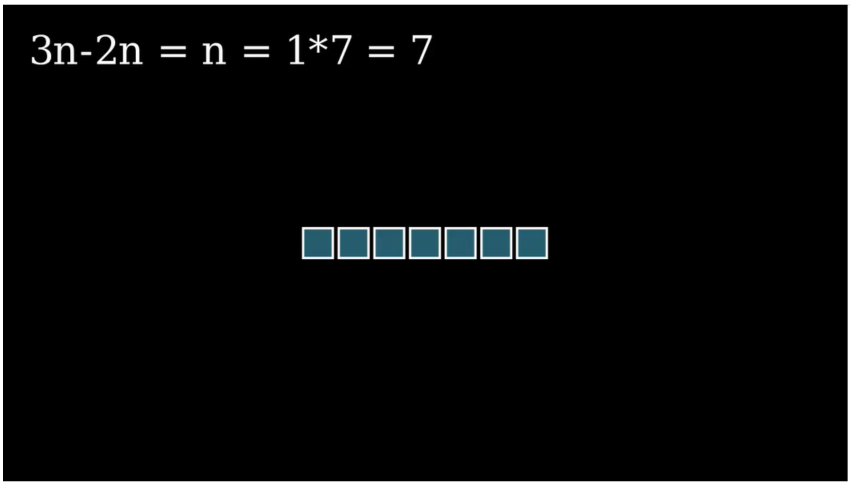


Figura 3.26

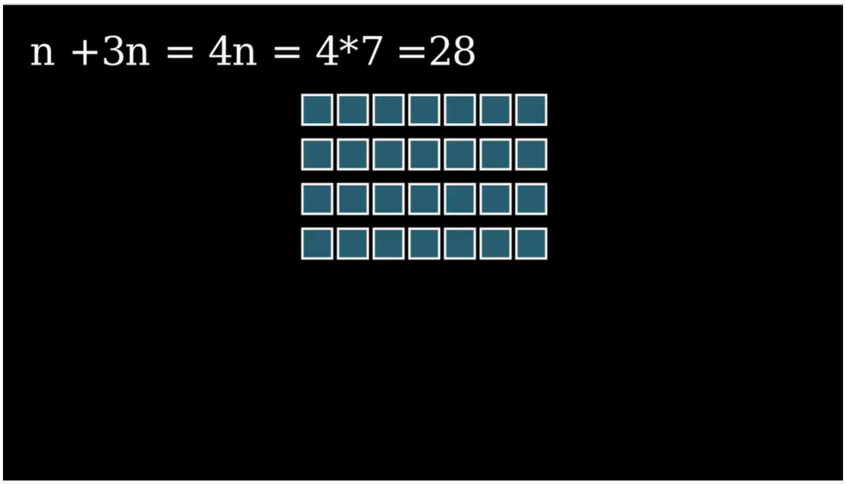


Figura 3.27

expressões algébricas ($2n+n = 3n$, por exemplo), compreender o significado da expressão $3n$, que pode ser feita ao se comparar a expressão com a representação gráfica pictórica que aparece na animação, que é 3 de uma coisa que chamamos de $n$ (as colunas com $n = 7$, quadradinhos, podendo $n$ assumir qualquer outro valor) e valor numérico (se eu tenho 3 fileiras de $n$ objetos mas $n$ é igual a 7 e, corresponde o que se é visto, que são 3 fileiras de 7 objetos, tem-se 3 vezes 7 objetos, isto é 21 objetos).

Então, em cada cena da animação é trabalhada tanto a conversão entre registros de diferentes sistemas semióticos como tratamentos dentro de um mesmo sistema. Se eu quero somar $2n$ com $n$, existe uma correspondente fase da animação em que se tem duas fileiras de $n$ objetos o que corresponde a $2n$ e acrescenta-se mais uma fileira, que corresponde a $n$, obtendo 3 fileiras de $n$ objetos que corresponde a $3n$, ao mesmo tempo em consonância com o tratamento algébrico, que obtém os mesmos $3n$. Pois a regra do tratamento algébrico é somar os coeficientes e conservar a a parte literal em comum, neste caso a indeterminada $n$, mas o coeficiente de $n$ é 1, aí faz se $2+1$ e conserva-se o $n$, obtendo $3n$.

Lembremos do conceito de discriminação visual, que vimos no primeiro capitulo como a habilidades que permite comparar dois ou mais objetos identificando suas semelhanças e diferenças. Tal conceito pode ser trabalhado, a cada etapa da animação na qual ocorre acréscimo ou retirada de fileiras na representação figural, simultaneamente a mudança no valor do coeficiente na representação algébrica/simbólica. De fato, esses são elementos significativos do objeto (número de fileiras e coeficiente do monômio) nas suas respectivas representações e existe uma *correspondência semântica* entre os mesmos, portanto, uma mudança neles em um registro causa automaticamente uma mudança no outro e vice-versa, em virtude da mudança do objeto representado.

Embora a indeterminada $n$ se conserve na representação simbólica e seu correspondente figural, que é o número de quadrados em cada fileira, mantenha a semelhança entre o antigo e o novo objeto (monômio), os dois elementos significativos, que foram alterados evidenciando suas diferenças, é o que caracteriza o fato de eles serem distintos.

## 3.4 A estrutura geral dos códigos em Python por trás de cada animação

Em todas as animações uma das primeiras coisas que foram feitas foi definir os objetos que seriam colocados na cena, que era uma ou mais fileiras de quadradinhos de determinado tamanho e determinada cor, que definimos no código. Esses quadradinhos, que já são objetos definidos na biblioteca Manim, restando a nós definir suas especificações, são usados para construir um outro objeto ao serem agrupados pelo método `VGroup`. Nesse método, também, precisamos definir a quantidade de quadradinhos que queremos agrupar. Como eu quero que, a cada execução do programa, a animação seja feita com uma quantidade distinta e imprevisível de quadradinhos, a biblioteca `random` foi importada, no inicio do código para usarmos como recurso um método que gera um número inteiro aleatório dentro de um intervalo de valores.

Em todas as animações a escolha foi feita entre números de 1 a 9, pelo método `randint`. Uma vez agrupados os quadrados pelo método `VGroup`, é necessário definir como se quer que eles fiquem espacialmente dispostos. O interesse aqui é que eles fiquem dispostos de forma a gerar uma fileira horizontal dos mesmos. Para isso, se aplica ao agrupamento o método `arrange` e se define `RIGHT` como seu parâmetro, para indicar que queremos dispor os quadrados um um, com o seguinte à direita do atual, o que gera o efeito de fileira horizontal, conforme se é desejado.

Até então, se executarmos o código, simplesmente só aparecerá a tela totalmente preta, pois só foi executado o que foi colocado em código até então, que foi a construção desses objetos, que já estão disponíveis para serem adicionados e, portanto, serem exibidos, porém isso ainda não foi pedido em código (adição desses objetos em tela) e só é feito se aplicarmos o método `self.add` sobre o mesmo, o que fará agora aparecer em tela, a respectiva fileira horizontal que representa visualmente na cena cada objeto inicialmente construído no código.

Uma vez explicado quais são as primeiras ações geradas pelo código nas animações, de modo geral, vejamos os códigos criados pra representar a adição e a multiplicação e em seguida a explicação de mais alguns comandos importantes com seus respectivos efeitos. Vejamos:

```
%%manim -qm -v  WARNING adicao

class adicao(Scene):
  def construct(self):
    m = random.randint(1,10)
    n = random.randint(1,10)
    blue_squares = VGroup(*[Square(fill_color=BLUE, fill_opacity=0.5,
     side_length=0.3) for _ in range(m)])
    blue_squares.arrange(RIGHT, buff=0.1)
    texto_1a = Text(f"{m}+{n} ={m+n}")
    texto_1a.to_edge(UL)
    self.add(texto_1a)
    self.add(texto_1a)
    expressao_1 = f' {m} + {1}={m+1}'
    texto_1 = Text(expressao_1)
    texto_1.to_corner(DL)
    self.add(texto_1)
    self.add(blue_squares)
    self.play(FadeOut(texto_1))
    self.wait(3)

  for i in range(1, n+1):
     squares=Square(fill_color=BLUE, fill_opacity=0.5, side_length
   =0.3)
       blue_squares = squares.next_to(blue_squares,RIGHT, buff = 0.1)
       expressao_1 = f' {m} + {i} = {m}+{i}*{1} = {m+i}'
       texto_1 = Text(expressao_1)
       texto_1.to_corner(DL)
       self.add(texto_1)
       self.add(squares)
       self.wait(3)
       self.play(FadeOut(texto_1))
```

```python
%%manim -qm -v  WARNING multiplicacao

class multiplicacao(Scene):
  def construct(self):
    m = random.randint(1,10)
    n = random.randint(1,10)
    blue_squares = VGroup(*[Square(fill_color=BLUE, fill_opacity=0.5,
     side_length=0.3) for _ in range(m)])
    blue_squares.arrange(RIGHT, buff=0.1)
    texto_1a = Text(f"{m}*{n} ={m*n}")
    texto_1a.to_edge(UL)
    self.add(texto_1a)
    self.add(texto_1a)
    expressao_1 = f' {m} * {1}={m*1}'
    texto_1 = Text(expressao_1)
    texto_1.to_corner(DL)
    self.add(texto_1)
    self.add(blue_squares)
    self.play(FadeOut(texto_1))
    self.wait(3)
    for i in range(2, n+1):
       blue_squares_copy = blue_squares.copy()
       blue_squares_copy.next_to(blue_squares,UP, buff = 0.1)
       blue_squares = blue_squares_copy
       expressao_1 = f' {m}*{i} = {m}*{i-1}+{m} = {m*i}'
       texto_1 = Text(expressao_1)
       texto_1.to_corner(DL)
       self.add(texto_1)
       self.add(blue_squares)
       self.wait(3) self.play(FadeOut(texto_1))\\

```

Comparando os dois códigos acima, pode-se observar, nos mesmos, uma mesma estrutura geral, o que reflete o fato de o raciocínio usado na construção de cada um deles se basear na estrutura recursiva da definição das mesmas por meio do princípio de indução, conforme já foi mostrado nesse trabalho.

Os 3 demais códigos — comutatividade e associatividade da adição e monômios — estão disponíveis no link abaixo do GeoGebra Book Todos os códigos escritos para esse Produto Educacional podem ficar mais compactos, isto é, com menos linhas mas sem comprometer sua funcionalidade ou propósito. Ao leitor interessado fica a tarefa de fazer isso, dando uma continuidade ao trabalho, o qual está disponível para a comunidade acadêmica e escolar, para ser estudado, analisado, revisado e melhorado, a fim de aplicá-lo, de um modo mais eficiente e eficaz, a suas práticas de ensino e adaptando-o a cada realidade escolar individual. O propósito didático de compactar o código é facilitar uma maior exploração da ideia por professores que não tenham suficientes habilidades em programação, ajudando-o a enxergar melhor, tendo uma mínima compreensão do que fazer para alterar o código de modo que atenda a uma específica ideia, no trabalho de determinado conceito com os alunos.

Segue a explicação de comandos importantes presentes nos códigos.

`texto_1a = Text(f"{m}+{n} ={m+n}")` e linhas subsequentes similares gera um objeto da classe texto, contendo uma `string` formada pela expressão da adição dos valores numéricos das variáveis `m` e `n`, mostrando sua soma após o sinal de igualdade.

`texto_1a.to_edge(UL)` : Esse comando é um método aplicado a objetos da classe texto, como `text_1`, e seu efeito sobre este objeto é inseri-lo na borda superior esquerda da tela. Por `default`, isto é, convencionalmente, se você não aplica esse método sobre o texto, ao ser adicionado à tela, ele é automaticamente inserido no seu centro, assim como todos os demais objetos inseridos na cena. Se é solicitado o código que algum objeto seja adicionado à cena, mas o programador não definiu onde, ele deve ser adicionado em algum lugar e, o esperado é que por `default`, ele seja adicionado ao centro da tela. Organizamos a disposição espacial dos objetos à cena aplicando a eles métodos como este que o colocam em uma posição particular da tela ou em relação a um outro objeto já pré-adicionado.

`self.add(texto_1a)` adiciona o o objeto `texto_1a` criado à tela.

Promover uma compreensão mais detalhada do mecanismo pelo qual se faz as operações de adição e multiplicação, que tem base conceitual na sua definição é a intenção de gerarmos aleatoriamente vários valores para serem trabalhados na animação em cada execução do programa a fim de que para o aluno fique não a conta de dois valores específicos em si mas a abstração por trás de cada conjunto de valores que é sorteado ao rodar cada vez o código.

E essa abstração está justamente contida no modo como essas linhas de código pedem que o programa rode, que é destacando os aspectos do que é conceitualmente uma adição, a ideias de tomar sucessor uma ou mais vezes, e do que é uma multiplicação, que é a ideia de somar um certo valor com ele próprio sucessivas vezes.

Algumas das mais imediatas implicações lógicas dessa definição do que é adição, como suas propriedades comutativas e associativa, também são ilustradas na representação pictórica dos quadradinhos que é trabalhada em todas as animações. O professor pode, então aproveitar esse fato é pedir aos alunos que imaginem e descubram, junto, outros fatos a respeito das propriedades operacionais dos números naturais, contemplando a habilidades **EF06MA03** da BNCC.

Esses fatos podem ser um modo inteligente de aplicar essas propriedades ou uma extensão delas, que também é válida para números inteiros e outros conjuntos, como a seguinte: para facilitar certos cálculos, que de outro modo seria mais complicado. Por exemplo, fazer $999 * 4$, fica menos custosa de se resolver se for notado que $999 = 1000 - 1$ e aplicar a propriedade distributiva dos números inteiros fazendo $999 * 4 = (1000 - 1) * 4 = 4000 - 4 = 3996$.

Assimilar as propriedades operatórias permite ao aluno um maior dinamismo e flexibilidade para desenvolver diversos modos de resolver aritmeticamente um problema matemático, inclusive, optar por aquele que faça a resolução ficar menos trabalhosa para si, além de permitir que ele veja diferentes aspectos do mesmo problema, expressos pelas suas distintas resoluções. Analisemos, nesse sentido, os itens (a) e (b) do problema abaixo:

*(Saresp-SP)* A tabela abaixo mostra a distribuição dos alunos dos 3 turnos de uma escola, de acordo com o sexo:

| | 1 turno | 2 turno | 3 turno |
|---|---|---|---|
| Meninas | 135 | 120 | 105 |
| Meninos | 120 | 115 | 125 |

É correto afirmar que:

(a) todos os turnos possuem o mesmo número de alunos.

(b) a escola tem um total de 360 alunosl

(c) o número de meninas é maior que o número de meninos.

(d) O 3 turno tem 230 alunos.

No item (a), o aluno precisa verificar se, somando os dois valores de cada coluna, que resulta no número de alunos de cada turno, se obtém o mesmo valor. Em cada turno a ser verificado, se o aluno achar mais custoso cognitivamente somar o número de meninas com o de meninos, por exemplo, ele pode facilitar o processo, somando o número de meninas com o de meninos e, pela propriedade comutativa, se obtém o mesmo resultado.

Responder o item (b), se valendo das propriedades associativa e comutativa da adição, pode, também, permitir ao aluno ver o problema sob dois aspectos. Um deles é que o número de alunos da escola é a soma do número de meninos com o número de meninas da turma. Um outro modo de se obter uma expressão aritmética que resulta nesse número é vê-lo como o somatório do número de alunos de cada um desses três turnos.

Desse modo, isso corrobora a Teoria dos Registros de Representação Semiótica no que já foi dito que diferentes representações de um mesmo objeto contemplam diferentes aspectos semânticos do mesmo. No presente caso, um tratamento aritmético é feito, aplicando a propriedade associativa a uma das expressões aritméticas para o total de aluno, que contempla um aspecto semântico dessa quantidade, obtendo uma outra expressão aritmética que carrega um outro ponto de vista sobre o significado da mesma.

Vejamos um exemplo disso no caso da multiplicação em uma questão do livro [10]:

Um campo de futebol tem a forma retangular e mede $120m$ por $90m$. A área desse campo é obtida multiplicando o comprimento pela largura. Qual é a área do sítio que corresponde a 15 vezes a área desse campo?

A modelagem da situação indica que para calcular área do sitio deve-se calcular a área de um campo de futebol, que é $120 \times 90$ e multiplicar pela quantidade de campos destes equivalente a esse sítio, que é 15, sendo, portanto, a área do círculo expressa por $(120 \times 90) \times 15$. Mas se for conveniente, menos trabalhoso ao aluno, por exemplo, a associação $120 \times (90 \times 15)$ ou até uma outra configuração, se valendo da propriedade comutativa, também, como $90 \times (120 \times 15)$, isso pode ser feito sem prejuízo do resultado procurado.

Em termos da modelagem da situação em questão só faz sentido a expressão $(120 \times 90) \times 15$ mas numa concepção estritamente aritmética, temos que a área procurada é simplesmente $120 \times 90 \times 15$, a despeito de quaisquer formas produzidas ao se aplicar o tratamento de comutar ou associar os fatores desse produto.

## 3.5 Recursos do Python na versão GeoGebra5 (beta)

A versão 5 do GeoGebra permite fazer construções mais elaboradas dentro da plataforma, lançando mão da sintaxe e dos recursos da linguagem Python. Dentro do contexto do nosso trabalho, usamos essa nova ferramenta como um meio de fazer animações ilustrando representações pictóricas das propriedades aritméticas de adição e multiplicação dos números naturais e suas respectivas propriedades comutativas.

Com uma única biblioteca (`time`) e com recursos do GeoGebra, escrevendo poucas linhas de código Python, é possível construir essas animações. A vantagem dessas construções, em relação as que foram feitas com a biblioteca Manim, é que, uma vez feita a construção, fica fácil para o professor, sem muito conhecimento em programação, e para os alunos fazer a manutenção e adaptação dos códigos às necessidades demandadas por cada turma ou aluno.

O primeiro passo, antes de tudo, é importar a biblioteca `time`. Segue, então, o código que ilustra com apenas 4 linhas a representação pictórica da operação de adição usual sobre números naturais:

```
for i in range(0,5):
  time.sleep(0.4)
  for x in range(1,4):
```

```
4     Polygon([Point(x, i+1), Point(x+i, 1+i), Poin(x+1, 2+i), Point(x, 2+i)])
```

No caso desse código, é ilustrado o produto entre 3 e 5, ficando fácil sua adaptação para outros valores, fazendo uma mera mudança nos valores dos parâmetros no código. É a aplicação do método que vai dar a dinâmica da animação, construindo a cada 0.4 segundos (linha 2) uma fileira com 3 quadrados (linhas 3 e 4) em uma posição acima da fileira abaixo, se não for a primeira, (linha 4), 5 vezes consecutivas (linha 1), passando a ideia de contar de 3 em 3, 5 vezes, isto é, fazer $3 \times 5$.

Para ilustrar o comutatividade, podemos, abaixo desse código dar uma pausa, aplicando, novamente o método `sleep`, e construir o código análogo a esse porém construindo 3 fileiras verticais de 5 quadrados, cada uma, exceto a primeira, colocada, gradativamente, a direita da anterior, ilustrando agora a multiplicação de $5 \times 3$. Essa animação termina na sobreposição da figura que ilustra $5 \times 3$ sobre a figura que ilustra $3 \times 5$, mostrando a igualdade entre $5 \times 3$ ou $3 \times 5$, resultado da propriedade multiplicativa dessa operação que é evidenciada a cada vez que mudamos os valores dos parâmetros, além de 3 e 5. Segue o referido código:

```
for i in range(0,3):
  time.sleep(0.4)
  for x in range(1,6):
    Polygon([Point(1+i,y), Point(1+i, y+1), Point(2+i, y+1), Point(2+i,y)])
```

Em poucas linhas, os códigos seguintes ilustram a soma entre 5 e 3 e a soma entre 3 e 5, respectivamente. Colocados um abaixo do outro para serem executados um após o outro para serem executados em sequência, a figura que ilustra $5+3$ sobrepõe a figura que ilustra $3+5$, mostrando a igualdade entre essas expressões. Executando o programa com outros valores de parâmetro, além de 3 e 5, um número suficiente de vezes, se evidencia a comutatividade dessa operação. Seguem os códigos abaixo e em seguida as figuras, exibindo uma sequência de cenas da animação gerada por esses códigos:

```
time.sleep(0.4)
for x in range(1,6):
     Polygon([Point(x,1), Point(x+1, 1), Point(x+1,2), Point(x,2)])
time.sleep(0.4)
for i in range(1,4):
    time.sleep(0.4)
    Polygon([Point(x+i,1), Point(x+1+i, 1), Point(x+1+i,2), Point(x+i,2)])
```

```
time.sleep(0.4)
for x in range(1,4):
     Polygon([Point(x,1), Point(x+1, 1), Point(x+1,2), Point(x,2)])
```

```
time.sleep(0.4)
for i in range(1,6):
    time.sleep(0.4)
    Polygon([Point(x+i,1), Point(x+1+i, 1), Point(x+1+i,2), Point(x+i
   ,2)])
```

O leitor pode acessar a página abaixo é digitar os códigos acima na área de código para visualizar as animações descritas, que eles produzem. Trata-se, exatamente do mesmo código, exceto pela troca dos valores de dois parâmetros de entrada para ilustrar a igualdade $5 + 3 = 3 + 5$.

# Capítulo 4

# Considerações Finais

Esse trabalho termina reforçando a ideia central que permeia todo texto, que é o fato de que ao estudarmos, escrevermos, ao trabalharmos em sala de aula com o tema recursos computacionais no ensino de matemática, assim como qualquer outro tema de matemática e seu ensino, é importante, se não se faz necessário levar em conta a natureza abstrata dos entes matemáticos e como isso determina em uma melhor e mais eficiente aprendizagem de matemática. Uma teoria que considera tal natureza apontando pra tais implicações na aprendizagem dessa disciplina é a Teoria dos Registros de Representação Semiótica de Duval, que foi um dos principais objetos de discussão desse trabalho. Levando isso em conta, é esperado que esse trabalho motive novas pesquisas que confirmem a resposta afirmativa que fizemos, por uma metodologia estritamente teórico-argumentativa, baseada em toda literatura anterior a respeito dos temas envolvidas, de que o uso de animações computacionais como meio de compreender um objeto matemático através de um reconhecimento e articulação entre diferentes representações semióticas potencializa a aprendizagem matemática nesse sentido em relação a outros meios didáticos, até mesmo via uso de materiais concretos (e manipulativos). O objetivo de tudo isso é que este estudo se reflita melhores resultados em sala de aula, nas aulas de matemática, via a da leitura e aplicação de trabalhos acadêmicos desenvolvidos nessa temática pelos professores, com exploração e adaptação a cada realidade de sala de aula dos produtos educacionais resultantes desses trabalhos, começando pelo produto educacional desenvolvido neste.

# Apêndice

## Plano de Aula: O Conceito de Variável

**Discilina**: Matemática
**Público Alvo**: Alunos do 8º Ano do Ensino Fundamental
**Duração**: 2 aulas (50 min. cada)

## Conteúdo Programático

**Aula 1** : Aplicação do animação computacional para introduzir e ilustrar o conceito algébrico de variável

**Aula 2**: Discussão em grupo e verificação do que foi aprendido.

Seguem os links dos notebooks do GoogleColab, onde estão respectivamente, os códigos que geram as animações ilustrativas da comutatividade da adição, associatividade da adição e de soma de monômios.

`https://colab.research.google.com/drive/1eRON_LglW6OkAaAM4DDIYS16Oy4WGHv-`

`https://colab.research.google.com/drive/1CUL__HkyGkNy43JM9UcfDwZ_KD9TssQP#scrollTo=F6KrH1wLigx1`

## Objetivos de Aprendizagem

### Objetivo Geral

- Desenvolver e aplicar o conceito de variável na resolução de problemas matemáticos.

### Objetivos Específicos

- Desenvolver e aplicar o conceito de variável na resolução de problemas matemáticos.
- Identificar o conceito de variável no cotidiano.

- Relacionar a representação simbólica de expressões algébricas com convenientes representações pictóricas via conversão.
- Relacionar os tratamentos feito sobre expressões algébricas em suas respectivas representações simbólicas e pictóricas;

# Recursos

- Computador com acesso a internet (para uso do Google Colab e do GeoGebra Beta).
- Projetor, para exibir as animações.
- Quadro branco.
- Marcador de quadro e apagador
- Papel e caneta, *chromebooks* ou outro tipo de dispositivo portátil para os estudantes.

# Habilidades e Competências (BNCC)

- COMPETÊNCIA ESPECÍFICA 3: [**?**] Utilizar estratégias, conceitos, definições e procedimentos matemáticos para interpretar, construir modelos e resolver problemas em diversos contextos, analisando a plausibilidade dos resultados e a adequação das soluções propostas, de modo a construir argumentação consistente;
- (EM13MAT405) Utilizar conceitos iniciais de uma linguagem de programação na implementação de algoritmos escritos em linguagem corrente e/ou matemática;
- COMPETÊNCIA ESPECÍFICA 5: Investigar e estabelecer conjecturas a respeito de diferentes conceitos e propriedades matemáticas, empregando estratégias e recursos, como observação de padrões, experimentações e diferentes tecnologias, identificando a necessidade, ou não, de uma demonstração cada vez mais formal na validação das referidas conjecturas.

# Passo a passo da aula

:

1. Introdução (10 minutos): a. Contextualização: Explicar ao estudante que, nessa aula, será introduzido uma nova noção,de fundamental relevância para a álgebra, que é a noção de variável. b. Discussão: Perguntar aos alunos se eles já ouviram falar de variáveis antes e se eles tem alguma noção do que uma variável seja.

2. Definição e representação de variáveis (20 minutos): a. Apresentação teórica: Explicar a noção de variável, usando exemplos simples e mostrando sua representação por letras. b. Atividade prática: Pedir aos alunos que construam uma expressão algébrica simples para representar uma situação do cotidiano. Por exemplo se eu compro $s$ pães de sal e $m$ pães de milho, ao todo, comprei $s + m$ pães.

3. Animações da Biblioteca Manim e do GeoGebra Python (20 minutos): a. Demonstração: Apresentar as animações construidas pno Manim e no GeoGebra Python b. Exploração Prática: Dividir os estudantes em grupo e pedir que eles alterem os parâmetros do código em seus chromebooks para uma melhor exploração das animações do GeoGebra Python, testando sua execução com diferentes valores, afim de construr uma boa generalização das ideias ali trabalhadas.

4. Grupo de discussão (Aula 2): Um integrante de cada grupo compartilha com o restante da turma as conclusões e aprendizados tirados da manipulação do *software*.

# Referências Bibliográficas


[1] A essência da criatividade é . . . Jerome Brunner. Acessado em 2024-02-17.

[2] Congruência semântica. https://1library.org/article/congru%C3%AAncia-sem%C3%A2ntica-no%C3%A7%C3%B5es-de-registros-de-representa%C3%A7%C3%A3o-semi%C3%B3tica.ydve66ly. Accessed: 2024-02-17.

[3] O computador ideal deve funcionar como...alan kay. https://www.pensador.com/frase/MjMzMDE0Mw/. Accessed: 2024-02-17.

[4] Lourdes Maria Werle de Almeida and Karina Alessandra Pessoa da Silva. Abordagens Semióticas em Educação Matemática. *Bolema*, 32(61):696–726, 2018.

[5] Geni Pereira Cardoso and Raimundo Luna Neres. A Mobilização e Coordenação de Registros de Representação Semióticos no Ensino e Aprendizagem de Fração nos Iniciais. *Educação Matemática em Revista*, 26(72):9–21, 2021.

[6] R. Duval. *Semiósis e pensamento humano: registros semióticos e aprendizagens intelectuais*. Livraria da Física, 2009.

[7] Raymond Duval. Registros de representações semiótica e funcionamento cognitivo da compreensão em matemática. In Silvia Dias Alcântara Machado, editor, *Aprendizagem em matemática: registros de representação semiótica*, pages 11–33. Papirus, Campinas, 2003.

[8] Raymond Duval. Registro de Representação Semiótica e Funcionamento Cognitivo do Pensamento. *Revemat: Revista Eletrônica de Educação Matemática*, 7(2):266–297, 2012.

[9] Jordan Ellenberg. *O Poder do Pensamento Matemático: A Ciência de Como Não Estar Errado*. Zahar, 2014.

[10] José Ruy Giovanni Junior and Benedicto Castrucci. *A conquista da matemática*. FTD, São Paulo, 2018.

[11] Muriel Lefebvre. *Images, Écritures et Espace de médiation.* Thèse de doctorat, Université Strasbourg I, Strasbourg, Décembre 2001.

[12] José Augusto NG Manzano and Jayr Figueiredo de Oliveira. *Algoritmos lógica para desenvolvimento de programação de computadores.* Saraiva Educação SA, 2000.

[13] Jenny Patricia Acevedo Rincón and Campo Elías Flórez Pabón. Aproximações ao ensinoaprendizagem de expressões algébricas: Uma experiencia na licenciatura.

[14] Simone Semmer, Sani de Carvalho Rutz da Silva, and Marcos Cesar Danhoni Neves. Arte e Matemática: Teoria dos registros de representação semiótica e proposta triangular. *Revista Ciências& Ideias*, 5(2):19–32, 2014.

[15] Valessa Leal Lessa de Sá Pinto, Abel Rodolfo Garcia Lozano, Angelo Santos Siqueira, and Adriano Vargas Freitas. Reflexões para o ensino de aritmética nos anos iniciais da educação básica: o pensamento lógico matemático e o desenvolvimento da abstração. *Revista UNEABEU*, 7(15):227–238, 2014.

[16] Floriano Viseu, Ana Luísa Pires, Luís Menezes, and Ana Maria Costa. Semiotic Representations in the Learning of Rational Numbers by $2^o$ Grade Portuguese Students. *International Electronic Journal of Elementary Education*, 13(5):611–624, 2021.