

UNIVERSIDADE ESTADUAL DE MATO GROSSO DO SUL

UNIDADE UNIVERSITÁRIA DE NOVA ANDRADINA

CURSO DE MATEMÁTICA, LICENCIATURA.

GEOMETRIA RECURSIVA NO SUPERLOGO

LEANDRO FERREIRA DA SILVA

NOVA ANDRADINA-MS

2018

GEOMETRIA RECURSIVA NO SUPERLOGO

LEANDRO FERREIRA DA SILVA

Trabalho de conclusão de Curso de Matemática, Licenciatura, como parte das exigências para obtenção do grau de licenciado em Matemática pela Universidade Estadual de Mato Grosso do Sul, sob a orientação do Professor Me. Marcio Demetrius Martinez.

NOVA ANDRADINA-MS

2018

GEOMETRIA RECURSIVA NO SUPERLOGO

LEANDRO FERREIRA DA SILVA

Aprovado por:

Prof. Me. Marcio Demetrius Martinez-UEMS (Orientador)

Prof. Dr. Oyran Silva Rayzaro (UEMS)

Prof. Dr. José Felice (UEMS)

NOVA ANDRADINA-MS

2018

Agradecimentos

Eu agradeço, primeiramente, a Deus e a minha família, pois sem eles eu não conseguiria concluir este curso. Agradeço também ao professor Marcio, por ter me ajudado como aluno e agora também como orientador deste trabalho.

É claro, que não poderia deixar de agradecer todo corpo docente que passaram pela minha vida acadêmica, já que eles ajudaram a construir a base de todo esse trabalho. Agradeço também todos os funcionários da UEMS.

Por último, mas não menos importante, agradeço a todos os meus colegas de classe, na verdade muitos se tornaram amigos. Sempre me ajudaram com minhas dúvidas, e sempre me deram força para continuar. Também tivemos muitos momentos de alegria e descontração que levarei para o resto da vida.

“A Geometria dos Fractais não é apenas um capítulo da Matemática, mas também uma forma de ajudar os homens a verem o mesmo velho mundo diferentemente”.

Benoit Mandelbrot (1924).

Resumo

Este trabalho de conclusão de curso apresenta um estudo introdutório da Geometria Recursiva ou, mais especificamente, da Geometria Fractal. O estudo matemático dos fractais é relativamente novo; a maioria das descobertas nesse campo não tinha sido feita antes da segunda metade do século XX, quando os computadores se tornaram amplamente disponíveis para os matemáticos. Para melhor visualização e compreensão da fascinante geometria fractal, bem como proporcionar rapidez nos cálculos, verificou-se a necessidade da utilização de um recurso computacional ou software, chamado Superlogo. Embora esses objetos possam ser fascinantes por si mesmos, suas estruturas autossimilares podem nos ajudar a visualizar o conceito de recursão e explorar algumas de suas estruturas básicas como, área, perímetro, autossimilaridade, escala, complexidade e dimensão. Com esta monografia, espera-se que os leitores, sejam capazes de utilizar, explorar, bem como criar fractais em suas atividades didáticas destacando as várias áreas da Matemática na sua construção. Também com a inserção de alguns comentários históricos procura-se tornar o texto de interessante leitura.

Palavras-chaves: Geometria recursiva, fractais, Superlogo.

Abstract

This work of course completion presents an introductory study of Recursive Geometry or, more specifically, of Fractal Geometry. The mathematical study of fractals is relatively new; most of the discoveries in this field had not been made before the second half of the twentieth century when computers became widely available to mathematicians. For better visualization and understanding of the fascinating fractal geometry, as well as to provide quick calculations, it was verified the need to use a computational resource or software, called Superlogo. While these objects may be fascinating in their own right, their self-similar structures can help us visualize the concept of recursion and explore some of its basic structures such as area, perimeter, autosimilarity, scale, complexity, and dimension. With this monograph, readers, whether students or teachers, are expected to be able to use, explore, and create fractals in their didactic activities highlighting the various areas of mathematics in their construction. Also with the insertion of some historical comments it is tried to make the text of interesting reading.

Keywords: Recursive Geometry, fractals, Superlogo.

Sumário

INTRODUÇÃO	9
Capítulo 1.....	11
1.1 RECURSIVIDADE.....	11
1.2 A Sequência de Fibonacci	13
1.3 Modelando com Relações de Recorrência	14
Exemplo 1.1: Números Poligonais	14
Exemplo 1.2. Número de diagonais de polígonos regulares	16
Capítulo 2.....	17
2.1 SOLUÇÕES EM FORMAS FECHADAS E INDUÇÃO	17
2.2 Resolvendo Relações de Recorrência	18
2.4 Para ilustrar uma Solução por Indução	20
Capítulo 3.....	26
3.1 RECURSIVIDADE NO AMBIENTE SUPERLOGO.....	26
3.2 Principais Comandos.....	27
3.3 Menus	28
3.4 Programando Computadores	28
3.5 Sequências definidas recursivamente.....	31
3.6 Figuras Definidas Recursivamente	31
Características de um Fractal.....	34
3.7 Dimensão euclidiana e fractal.....	42
Mapa conceitual:.....	48
Considerações Finais.....	49
Apêndice	51
Bibliografia.....	61

INTRODUÇÃO

Durante séculos, os objetos e conceitos da geometria Euclidiana foram considerados como os que melhores descreviam o mundo em que vivemos. Euclides 325 - 265 a. C, (<http://wwwhistory.mcs.st-and.ac.uk/Biographies/Euclid.html>), desenvolveu vários estudos sobre as formas de objetos planos, mas objetos complexos como uma árvore, por exemplo, não tinham definição de forma e dimensão. Na tentativa de descrever alguns objetos complexos como este foi que se desenvolveu o estudo dos Fractais que nada mais são do que objetos e fenômenos da natureza que possuem formas irregulares, mas se observadas a diferentes escalas não perdem sua definição inicial, como por exemplo, as nuvens, montanhas, as árvores, batimentos do coração, a couve-flor, etc.

Nesta nova geometria, diferente da Euclidiana aprendida no curso de Matemática, é possível encontrar objetos com dimensão fracionária. Benoit Mandelbrot, (<http://www-history.mcs.st-and.ac.uk/Biographies/Mandelbrot.html>), foi o primeiro que utilizou a palavra “fractal” baseando-se no adjetivo fractus que, vem do verbo frangere, em latim, cujo significado é quebrar, fragmentar.

As principais características de um fractal são: a autossimilaridade e a complexidade infinita, pois os fractais são obtidos a partir de processos recursivos, isto é, a aplicação infinita de uma mesma regra de construção dentro de si mesma. Assim, eles tornam-se figuras com grande beleza e complexidade, as quais cada parte é semelhante ao todo.

Os fractais são uma das mais fascinantes descobertas da Matemática. A sofisticação e o exotismo de suas formas, bem como a ampla divulgação que lhes foi concedida pela mídia, despertaram a atenção do grande público e o interesse de grupos de pesquisadores, reduzidos no início, mas que se avolumaram à medida que os fractais começaram a invadir as áreas de outras ciências, como a Física, a Geologia, a computação gráfica, e encontrando inúmeras aplicações práticas, como a compreensão de arquivos de imagens, intensamente utilizada em multimídia, ingressando ainda no domínio das artes plásticas e adquirindo, dessa maneira, um caráter interdisciplinar.

Neste trabalho procuramos abordar muitos conceitos matemáticos, tais como lógica, razão, indução, semelhança, padrões geométricos, perímetros, áreas, sequências,

séries, progressão aritmética, progressão geométrica, limites, logaritmo, dimensão, dimensão fracionária, funções e gráficos, recursão, etc.

O auxílio da informática também é de importância para a compreensão da geometria fractal. Assim utilizamos o software Superlogo para traçar o fractal e apresentamos diversos programas recursivos na linguagem logo.

Capítulo 1

1.1 RECURSIVIDADE

Os galhos da árvore (Figura 1.1) seguem um padrão interessante, o tronco da árvore é um enorme caule central, com galhos saindo por todos os lados. Desses galhos saem caules mais finos, e assim continua, até o nível das agulhas. Um galho parece uma cópia em menor escala de toda a árvore, e mesmo um raminho se parece com a miniatura de um galho.

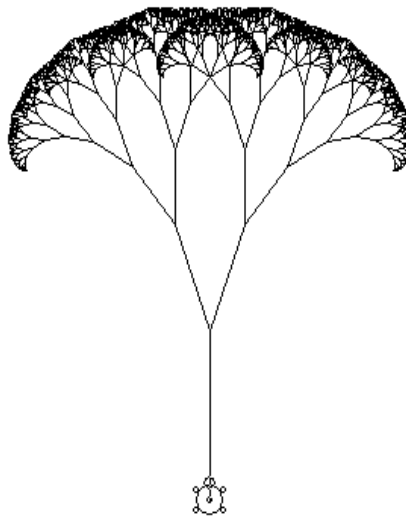


Figura 1.1 Arvore Fractal construída no Super-Logo

Na matemática, o tipo mais simples e mais concreto de objeto recursivo é a relação de recorrência. A definição de uma relação de recorrência é autorreferencial, enunciamos uma regra para calcular os valores de uma função em termos dela mesma.

Relações de recorrência têm duas partes: um caso base, que descreve o caso mais simples da função, e um caso recursivo, que descreve a função em termos de uma versão mais simples dela mesma.

Em outras palavras uma definição recursiva é aquela em que o item que está sendo definido é usado na sua própria definição, a princípio, pode parecer muito estranho que algo possa ser definido em termos de si próprio. Suponha que desejamos definir uma função $P: \mathbf{N} \rightarrow \mathbf{Z}$ cuja entrada é um número natural e cuja saída é um número inteiro. Uma forma natural de se fazer isso é dar uma regra ou fórmula explícita, como por exemplo, a que calcula a soma dos primeiros n números inteiros positivos,

$$P(n) = \frac{n(n+1)}{2} \quad (1.1)$$

Dado algum n pertencente aos naturais, basta substituir o valor de n na fórmula.

É sempre bom ter uma fórmula explícita para uma função, mas para alguns exemplos elas podem ser difíceis de encontrar ou existir. São comuns na matemática funções naturalmente definidas de maneira recursiva. Aqui está uma segunda forma de definirmos nossa função $P(n)$:

$$P(n) = \begin{cases} 1 & \text{se } n = 1 \\ n + P(n-1) & \text{se } n > 1 \end{cases} \quad (1.2)$$

Essa é uma função definição recursiva porque P é definida em termos de si mesma: P ocorre na fórmula que define P . Para todo $n \in \mathbb{N}$, podemos utilizar a definição na Equação 1.2 para calcular $P(n)$.

Para calcular $P(5)$, por exemplo,

A primeira abordagem é chamada de “ascendente” porque começamos de $P(1)$ e vamos “subindo” até chegarmos em $P(5)$. Pela primeira parte da definição, $P(1) = 1$. Pela segunda parte da definição, com $n = 2$, temos $P(2) = 2 + P(1) = 2 + 1 = 3$. Novamente, pela segunda parte da definição, agora com $n=3$, temos $P(3) = 3 + P(2) = 3 + 3 = 6$. Repetindo este processo, $P(4) = 4 + 6 = 10$, e finalmente $P(5) = 5 + 10 = 15$.

De forma alternativa, podemos fazer uma computação “descendente”. Sempre que $n > 1$, podemos aplicar a segunda parte da definição da função para substituir “ $P(n)$ ” por “ $n + P(n-1)$ ”. Essa substituição justifica os quatro primeiros usos do sinal =.

$$\begin{aligned} P(5) &= 5 + P(4) \\ &= 5 + 4 + P(3) \\ &= 5 + 4 + 3 + P(2) \\ &= 5 + 4 + 3 + 2 + P(1) \\ &= 5 + 4 + 3 + 2 + 1 \\ &= 15 \end{aligned} \quad (1.3)$$

O penúltimo sinal = é o resultado da aplicação da primeira parte da definição para substituir $P(1)$ por 1.

Note que, por si mesma, a equação, $P(n) = n + P(n-1)$ não definiria uma relação de recorrência, pois o cálculo ascendente não poderia nunca ser começado, e o cálculo descendente nunca terminaria. Uma relação de recorrência bem definida precisa de um caso base não recursivo que dê, de forma explícita, pelo menos um valor da função. Este valor às vezes é chamado de valor semente ou condição inicial.

1.2 A Sequência de Fibonacci

Um dos primeiros exemplos de pensamento recursivo é a famosa sequência de Fibonacci. No começo do século XIII, o matemático Leonardo Pisano Fibonacci (https://pt.wikipedia.org/wiki/Leonardo_Fibonacci) propôs o seguinte problema:

“Um certo homem coloca um par de coelhos em um lugar cercado em todos os lados por uma parede. Quantos pares de coelhos podem ser produzidos a partir desse par em um ano, se supõe que em todo mês cada par gera um novo par que se torna produtivo a partir do segundo mês?”

Seja $F(n)$ o que representa o número de pares de coelhos presentes no mês n . assumindo que o primeiro par de coelhos leva dois meses para tornar produtivo, temos

$$F(1) = F(2) = 1$$

Como um caso base, e o primeiro par de descendentes aparece no terceiro mês, então $F(3) = 2$. Todo mês, o número de novos pares de coelhos é igual ao número de coelhos presentes dois meses antes.

Pela definição temos que os números de Fibonacci $F(n)$ satisfazem a relação de recorrência a seguir:

$$F(n) = \begin{cases} 1 & \text{se } n = 1 \text{ ou } n = 2 \\ F(n-1) + F(n-2) & \text{se } n \geq 3. \end{cases} \quad 1.4$$

A segunda parte dessa definição, $F(n) = F(n-1) + F(n-2)$, é a parte recursiva. O termo $F(n-1)$ representa o número de pares de coelhos presentes no mês anterior. O termo $F(n-2)$ representa o número de pares de coelhos presentes dois

meses atrás. Portanto, a soma desses dois termos é o total de pares de coelhos presentes no mês n , o mês corrente. Alguns poucos dos primeiros termos da famosa sequência infinita de Fibonacci são

$$1, 1, 2, 3, 5, 8, 13, 21, 34, \dots$$

Apesar dessa origem um tanto excêntrica, a sequência de Fibonacci tem um número extraordinário de aplicações. Essa sequência é encontrada em uma variedade de contextos, incluindo cultivo de plantas, preços de ações, arquitetura, música e padrões de drenagem.

1.3 Modelando com Relações de Recorrência

O exemplo dos coelhos de Fibonacci mostra como pensar recursivamente sobre um problema, descrevendo esse problema com relação de recorrência. Como visto anteriormente uma relação de recorrência tem duas partes: um caso base, que descreve algumas condições iniciais, e um caso recursivo, que descreve um valor futuro em termos de valores prévios. Munido com essa forma de pensar, modelamos outros problemas usando relações de recorrência.

Exemplo 1.1: Números Poligonais. Os primeiros membros da Associação de Pitágoras (<https://pt.wikipedia.org/wiki/Pit%C3%A1goras>) definiram números poligonais como sendo o número de pontos em determinadas configurações geométricas precisamente por formarem figuras de duas dimensões - polígonos - como os quadrados, pentágonos, hexágonos, etc. Partindo do número 1, representado por um único ponto criou sequências em que ao número anterior é somado outro número em função do polígono correspondente:

Para os triangulares, temos a sequência 1, 3, 6, 10, 15, ... (1, 1+2, 1+2+3, 1+2+3+4, 1+2+3+4+5, ...) e, portanto, a relação de recorrência para encontrar qualquer termo é dada por:

$$T(n) = \begin{cases} 1 & \text{se } n = 1 \\ T(n-1) + n & \text{se } n \geq 2. \end{cases} \quad 1.5$$

Note que o n -ésimo termo da sequência triangular, $T(n)$, é a série:

$$T(n) = 1 + 2 + 3 + 4 + \dots + n = \frac{n(n+1)}{2},$$

que é a soma de uma progressão aritmética, PA, com n termos.

Para os quadrados, a sequência infinita 1, 4, 9, 16, 25, ..., pode ser reescrita $1^2, 2^2, 3^2, 4^2, 5^2, \dots$. Assim, o n -ésimo termo dessa sequência é dado por $Q(n) = n^2$. Assim, concluímos que a relação de recorrência para os números quadráticos é:

$$Q(n) = \begin{cases} 1 & \text{se } n = 1 \\ Q(n-1) + 2n - 1 & \text{se } n \geq 2. \end{cases} \quad 1.6$$

Também, é interessante notar que a soma de dois números triangulares consecutivos determina um número quadrado e, portanto, temos que:

$$Q(n) = \begin{cases} 1 & \text{se } n = 1 \\ T(n-1) + T(n) & \text{se } n \geq 2, \end{cases} \quad 1.7$$

e que a soma a soma dos primeiros termos quadráticos consecutivos é:

$$1^2 + 2^2 + 3^2 + 4^2 + 5^2 + \dots + n^2 = \frac{n(n+1)(2n+1)}{6}. \quad 1.8$$

Finalmente, para os números pentagonais: 1, 5, 12, 22, 35, O modelo algébrico recursivo de um número pentagonal, pode ser escrito das seguintes formas:

$$P(n) = \begin{cases} 1 & \text{se } n = 1 \\ P(n-1) + 3n - 2 & \text{se } n \geq 2, \end{cases} \quad 1.9$$

ou,

$$P(n) = \begin{cases} 1 & \text{se } n = 1 \\ 3T(n-1) + n & \text{se } n \geq 2, \end{cases} \quad 1.10$$

ou,

$$P(n) = \begin{cases} 1 & \text{se } n = 1 \\ \frac{3Q(n-1) + 5n - 3}{2} & \text{se } n \geq 2, \end{cases} \quad 1.11$$

ou

$$P(n) = \begin{cases} 1 & \text{se } n = 1 \\ Q(n-1) + T(n-1) + 2n - 1 & \text{se } n \geq 2, \end{cases}$$

1.12

ou também, na forma explícita, $P(n) = \frac{n(3n-1)}{2}$.

O quadro seguinte ajuda a visualizar estas sequências que se tratam de regras de relações geométricas, ou seja, formais.

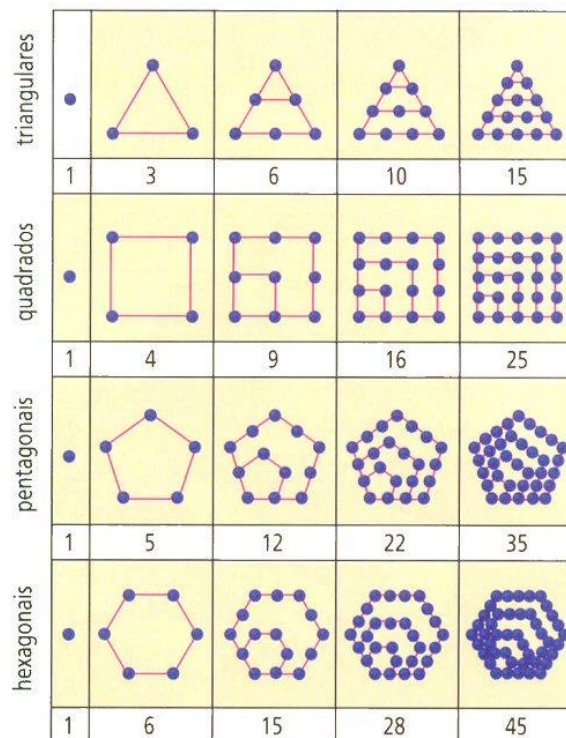


Figura 1.2 (extraído de http://obviousmag.org/archives/2006/08/perfeicao_divin_2.html)

Exemplo 1.2. Número de diagonais de polígonos regulares: Pela Figura 1.3 vemos que para um polígono de 4 lados, temos 1 diagonal partindo de um vértice; para um polígono de 5 lados, temos 2 diagonais partindo de um vértice; para um polígono de 6 lados temos 3 diagonais partindo de um vértice. Assim concluímos que o número de diagonais que parte de um vértice é igual a quantidade de lados do polígono menos 3. E para um polígono de N lados, teremos $N-3$ diagonais partindo de um vértice. Assim, podemos montar uma pequena tabela:

Número de lados	4	5	6	N
Número de diagonais de um vértice	1	2	3	$N - 3$

Tabela: 1.1

Como o número de vértices é igual ao número de lados do polígono, segue que teremos, com extremidade nos N vértices: $N \cdot (N-3)$ diagonais. No entanto, como cada diagonal tem extremidades em dois vértices, cada diagonal é contada duas vezes, então, basta dividirmos por dois:

$$d(N) = \frac{N(N-3)}{2},$$

ou, na forma recursiva,

$$d(N) = \begin{cases} 0 & \text{se } N = 3 \\ d(N-1) + N - 2 & \text{se } N \geq 4, \end{cases}$$

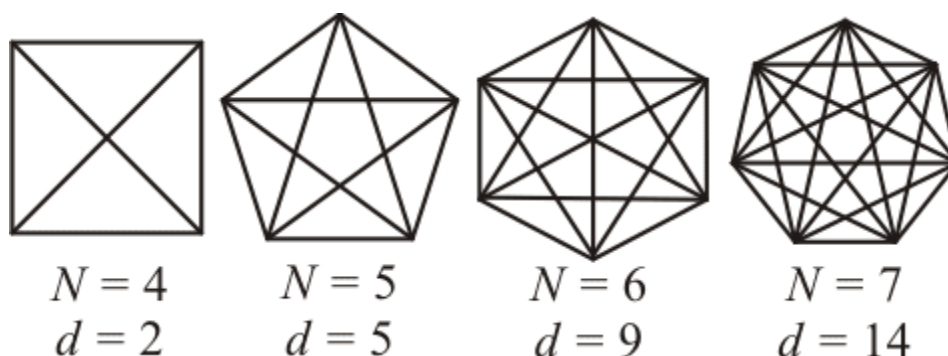


Figura 1.3

Assim, vimos que algoritmos recorrentes fornecem um modo natural de resolver determinados problemas chamando a mesma tarefa para versões menores do problema. Também, certas relações de recorrência têm solução em forma explícita ou fechada.

Capítulo 2

2.1 SOLUÇÕES EM FORMAS FECHADAS E INDUÇÃO

Embora as relações de recorrência tenham uma certa qualidade elegante, pode ser tedioso calcular com elas. Por exemplo, munidos apenas com a Definição 1.4, levaríamos 998 passos para calcular o número de Fibonacci $F(100)$. Nesta seção, iremos

explorar maneiras de encontrar uma solução em forma fechada não recursiva para uma relação de recorrência. O conceito essencial nesta seção é a ideia de indução matemática, o método pelo qual verificamos a correção de uma solução em forma fechada.

2.2 Resolvendo Relações de Recorrência

Suponha que $P(n)$ seja uma função definida por uma relação de recorrência. Gostaríamos de ter uma expressão:

$$P(n) = \text{uma função não recursiva } n$$

Assim poderíamos calcular $P(n)$ simplesmente substituindo n na fórmula, em vez de usar a relação de recorrência repetidas vezes. Uma fórmula como essa é chamada de *solução em forma fechada* da relação de recorrência. Lembre-se da Definição 1.4 e também dos números triangulares, descobrimos que os cinco primeiros valores da relação de recorrência eram 1,3,6,10, 15.

$$P(n) = \begin{cases} 1 & \text{se } n = 1 \\ n + P(n - 1) & \text{se } n > 1 \end{cases} \quad 2.1$$

Uma solução em forma fechada para essa relação de recorrência é a fórmula 1.2:

$$P(n) = \frac{n(n+1)}{2} \quad 2.2$$

Podemos verificar que os valores de $P(n)$ dados por essa fórmula não recursiva correspondem aos valores dados pela relação de recorrência:

n	1	2	3	4	5
$\frac{n(n+1)}{2}$	1	3	6	10	15

Tabela 2.1

Essa tabela de valores é uma evidência bem convincente de que descobrimos a solução correta em forma fechada, mas não é uma prova. Mais adiante nesta seção, veremos como verificar que tal solução está correta, mas primeiro iremos considerar o problema de encontrar soluções em forma fechada.

Uma equação como 2.2 na qual podemos substituir um valor e obter diretamente o que queremos, é chamando de uma solução em **forma fechada** para a relação de recorrência 2.2. Quando encontrada uma solução em forma fechada, dizemos que **resolvemos** a relação de recorrência.

Uma técnica para resolver relações de recorrência é uma abordagem do tipo “expandir, conjecturar e verificar”, que usa repetidamente a relação de recorrência para expandir a partir do n -ésimo termo até podermos ter uma ideia da forma geral. Finalmente essa conjectura é verificada por indução matemática.

Por exemplo, considere novamente a condição inicial e a relação de recorrência para a sequência Q do exemplo 1.1,

$$Q(n) = \begin{cases} 1 & \text{se } n = 1 \\ Q(n-1) + 2n - 1 & \text{se } n \geq 2. \end{cases}$$

Vamos fingir que não sabemos a solução em forma fechada e usar a abordagem de expandir, conjecturar e verificar para encontrá-la.

$$\begin{aligned} Q(n) &= Q(n-1) + 2n - 1 \\ &= Q(n-2) + 2(n-1) - 1 + 2n - 1 = Q(n-2) + 2n + 2(n-1) - 2 \\ &= Q(n-3) + 2(n-2) - 1 + 2(n-1) - 1 + 2n - 1 = Q(n-3) + 2n + 2(n-1) + 2(n-2) - 3 \end{aligned}$$

Olhando para o padrão que está se desenvolvendo, conjecturamos que, após k tais expansões, a equação tem a forma

$$Q(n) = Q(n-k) + 2n + 2(n-1) + 2(n-2) + \dots + 2(n-(k-1)) - k \quad 2.3$$

Essa expansão dos elementos de Q em função de elementos anteriores tem que parar quando $n-k=1$, isto é, quando $k=n-1$. Nesse ponto, temos

$$Q(n) = Q(1) + 2n + 2(n-1) + 2(n-2) + \dots + 2(2) + 2(1) - 2(1) - n + 1 = \frac{2n(n+1)}{2} - n - 1 = n^2,$$

que expressa solução em forma fechada.

2.4 Para ilustrar uma Solução por Indução

A indução matemática é a técnica geralmente utilizada quando se pode pensar sobre um problema de forma recursiva. Para ilustrar como ela funciona, imaginemos subindo uma escada infinitamente alta. Como sabemos se conseguiremos chegar a um degrau arbitrariamente alto? Suponhamos que façamos as seguintes hipóteses sobre nossa capacidade de subir:

- 1) Conseguimos alcançar o primeiro degrau;
- 2) Uma vez chegando a um degrau, somos capazes de chegar ao próximo. (Note que esta afirmativa é um condicional.)

O modelo geral para verificarmos uma solução para uma relação de recorrência é o seguinte. Temos:

$R(n)$ = uma relação de recorrência

$f(n)$ = uma fórmula hipotética em forma fechada

e gostaríamos de mostrar que $R(n) = f(n)$ para todos os valores de n . Para os propósitos desta discussão, vamos dizer que o primeiro valor de n para o qual $R(n)$ é definido é $n = 1$. Como já vimos, relações de recorrência começam em $n = 1$ ou $n = 3$, mas qualquer valor inicial de n é possível.

Para provar que $R(n) = f(n)$ para todo $n \geq 1$, precisamos usar a técnica da *indução matemática*. A ideia desse tipo de prova é análoga a subir em uma escada.

Subir uma escada é uma tarefa bastante repetitiva; se você sabe como subir um degrau, então você sabe que você precisa começar pela parte mais baixa da escada. Nesta demonstração, esse é o caso base.

Caso Base: Verifique que $R(1) = f(1)$.

Verificar o caso base é geralmente bastante fácil. Afinal, você provavelmente não teria escolhido $f(n)$ como uma candidata a solução se ela pelo menos não correspondesse ao caso $n = 1$.

Em seguida, você deve ser capaz de subir um degrau da escada. Note que não importa onde você está na escada, subir um degrau exige a mesma quantidade de

habilidades, esteja você na parte inferior da escada, no topo ou em algum lugar no meio. Então, vamos supor, para o bem do argumento, que você está em pé no $(k - 1)$ -ésimo degrau. Essa é a hipótese indutiva.

Hipótese Indutiva: Seja $k > 1$ algum número inteiro (não especificado). Suponha como hipótese indutiva que $R(k - 1) = f(k - 1)$.

A hipótese indutiva será crucial para nossa demonstração. Qualquer demonstração válida por indução deve-se usar a hipótese indutiva em algum lugar em seu argumento.

Finalmente, precisamos ser capazes de subir para o próximo degrau da escada. Veja a Figura 2.1.

Passo Indutivo: Prove que $R(k) = f(k)$.

Para fazer isso, vamos usar a relação de recorrência para calcular o K -ésimo valor. Quando precisarmos do $(k - 1)$ -ésimo valor, usemos a hipótese de indução. Então usamos a álgebra para mostrar que a resposta coincide ou não com a solução em forma fechada.

Por que isso funciona? Um argumento por indução prova a seguinte afirmação:

Seja $k > 1$. Se a relação de recorrência coincide com a solução em forma fechada para $n = k - 1$, então a relação de recorrência coincide com a solução em forma fechada para $n = k$.

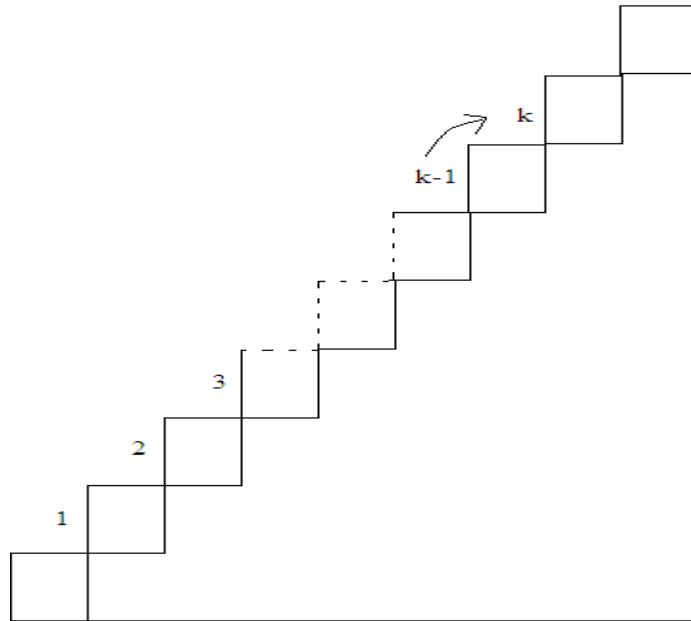


Figura 2.1 Criada no Paint- O passo indutivo mostra que você pode subir um degrau de uma escada.

Na linguagem matemática, temos:

$$R(k - 1) = f(k - 1) \rightarrow R(k) = f(k). \quad 2.5$$

Uma vez que a demonstração supôs um valor arbitrário de k , estamos autorizados a aplicar essa afirmação para *qualquer* valor particular de $k > 1$. Usando $k = 2$, temos a afirmação:

Se a relação de recorrência coincide com a solução em forma fechada para $n = 1$, a relação de recorrência coincide com a solução em forma fechada para $n = 2$.

Mas sabemos que a relação de recorrência coincide com a solução em forma fechada para $n = 1$; esse era o caso base. Portanto, a afirmação anterior (e *modus ponens*) nos diz que elas coincidem para $n = 2$. Agora aplicando a afirmação novamente, com $k = 3$:

Se a relação de recorrência coincide com a solução em forma fechada para $n = 2$, então a relação de recorrência coincide com a solução em forma fechada para $n = 3$.

Então sabemos que elas coincidem para $n = 3$. Ao repetir esse argumento, podemos verificar os casos $n = 4, 5, 6, \dots$, até qualquer número de quisermos. Em outras palavras, a relação de recorrência concorda com a solução em forma fechada para *qualquer* valor de n . Isso é exatamente o que precisávamos demonstrar. Nesses

símbolos de lógica, um argumento de indução estabelece a seguinte cadeia de implicações:

$$R(1) = f(1) \rightarrow R(2) = f(2)$$

$$\rightarrow R(3) = f(3)$$

$$\rightarrow R(4) = f(4)$$

$$\rightarrow R(5) = f(5)$$

$$\rightarrow \dots$$

O caso base dá a partida nessa cadeia de implicações, e uma vez que o passo indutivo funciona para qualquer valor de k , podemos continuar a cadeia de implicações indefinidamente para concluir que $R(n) = f(n)$ para qualquer $n \geq 1$.

Os exemplos a seguir mostram como uma demonstração típica por indução deve parecer.

Exemplo 2.1. Vamos voltar e demonstrar por indução o Exemplo 1.1.

Seja $Q(n)$ definido pela seguinte relação de recorrência:

$$Q(n) = \begin{cases} 1 & \text{se } n = 1 \\ Q(n-1) + 2n - 1 & \text{se } n \geq 2. \end{cases} \quad 2.6$$

Seja $f(n) = n^2$. Vamos mostrar que $Q(n) = f(n)$ para todo $n \geq 1$.

Demonstração. Usamos indução em n .

Caso Base: Se $n=1$, a relação de recorrência diz que $Q(1)=1$, e a fórmula diz que $f(1) = 1^2 = 1$, portanto elas coincidem.

Hipótese indutiva: Suponha como hipótese indutiva que, para algum $k > 1$,

$$Q(k-1) = (k-1)^2 \quad 2.7$$

Passo indutivo: Usando a relação de recorrência

$$Q(k) = Q(k-1) + 2k - 1 = (k-1)^2 + 2k - 1 = k^2 - 2k + 1 + 2k - 1 = k^2.$$

2.8

Então, por indução, $Q(n) = f(n)$ para todo $n \geq 1$.

Exemplo 2.2. Um polígono simples fechado (figura 2.2) consiste em n pontos no plano ligados em pares por n segmentos de reta; cada ponto é a extremidade de exatamente dois segmentos de reta. A seguir são dados dois exemplos:

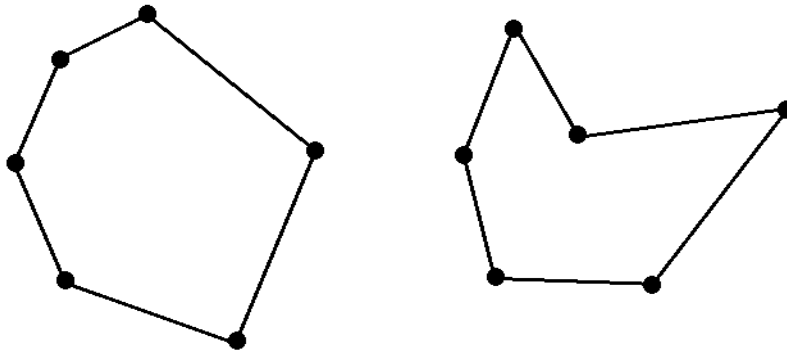


Figura 2.2 polígono simples fechado

A relação de recorrência que expressa a soma dos ângulos internos de um polígono simples fechado com n lados é

$$S(n) = \begin{cases} 180^\circ & \text{se } n = 3 \\ S(n-1) + 180^\circ & \text{se } n \geq 4. \end{cases} \quad 2.9$$

Mostremos que $S(n) = f(n) = (n-2) \cdot 180^\circ$ para todo $n \geq 3$.

Demonstração. Usamos indução em n .

Caso Base: Se $n=3$, a relação de recorrência diz que $S(3)=180^\circ$, e a fórmula diz que $f(3) = (3-2)180^\circ = 180^\circ$, portanto elas coincidem.

Hipótese indutiva: Suponha como hipótese indutiva que, para algum $k > 3$,

$$S(k-1) = (k-1-2)180^\circ \quad 2.10$$

Passo indutivo: Usando a relação de recorrência

$$S(k) = S(k-1) + 180^\circ = (k-1-2)180^\circ + 180^\circ = (k-2)180^\circ - 180^\circ + 180^\circ = (k-2)180^\circ. \quad 2.11$$

Então, por indução, $S(n) = f(n)$ para todo $n \geq 3$.

Como último exemplo.

Exemplo 2.3 Número de diagonais de polígonos regulares: (Veja o Exemplo 1.2)

Como vimos, o número de diagonais de um polígono regular é dado pela seguinte relação recorrente:

$$d(n) = \begin{cases} 0 & \text{se } n = 3 \\ d(n-1) + n - 2 & \text{se } n \geq 4, \end{cases} \quad 2.12$$

e a forma fechada é, para todo $n \geq 3$:

$$f(n) = \frac{n(n-3)}{2}. \quad 2.13$$

Demonstração. Usamos indução em n .

Caso Base: Se $n=3$, a relação de recorrência diz que $d(3)=0$, e a fórmula diz que $f(3) = \frac{3(3-3)}{2} = 0$, portanto elas coincidem.

Hipótese indutiva: Suponha como hipótese indutiva que, para algum $k > 3$,

$$d(k-1) = \frac{(k-1)(k-1-3)}{2}$$

Passo indutivo: Usando a relação de recorrência

$$d(k) = d(k-1) + k - 2 = \frac{(k-1)(k-1-3)}{2} + k - 2 = \frac{k(k-3)}{2} - \frac{k-3}{2} - \frac{k-1}{2} + k - 2 = \frac{k(k-3)}{2}.$$

Então, por indução, $d(n) = f(n)$ para todo $n \geq 3$.

Nunca teremos certeza se uma forma fechada concorda com uma relação de recorrência, a menos que demonstremos isso.

Capítulo 3

3.1 RECURSIVIDADE NO AMBIENTE SUPERLOGO

Neste capítulo serão apresentados brevemente alguns conceitos básicos de programação na linguagem Logo. Vamos criar e editar procedimentos, principalmente os procedimentos recursivos. Visto que, em muitos casos, um fractal pode ser gerado por um processo recorrente ou iterativo, esse será o nosso campo principal de aplicação.

Num primeiro momento faremos um breve reconhecimento do ambiente Superlogo e em seguida passaremos para exemplos de sequências definidas recursivamente, na criação de procedimentos recursivos e finalmente nas aplicações para alguns fractais.

A linguagem de programação Logo que foi projetada para ser um instrumento de aprendizagem, é uma linguagem poderosa e fácil de usar, criada nos anos 60 por Wallace Feurzeig, Daniel Bobrow e Seymour Papert. Existem muitas implementações da linguagem Logo, em diferentes idiomas e para diversos sistemas operacionais. A versão que utilizaremos é o Superlogo, um Logo para Windows que foi adaptado para o português pelo Núcleo de Informática Aplicada à Educação, da UNICAMP.

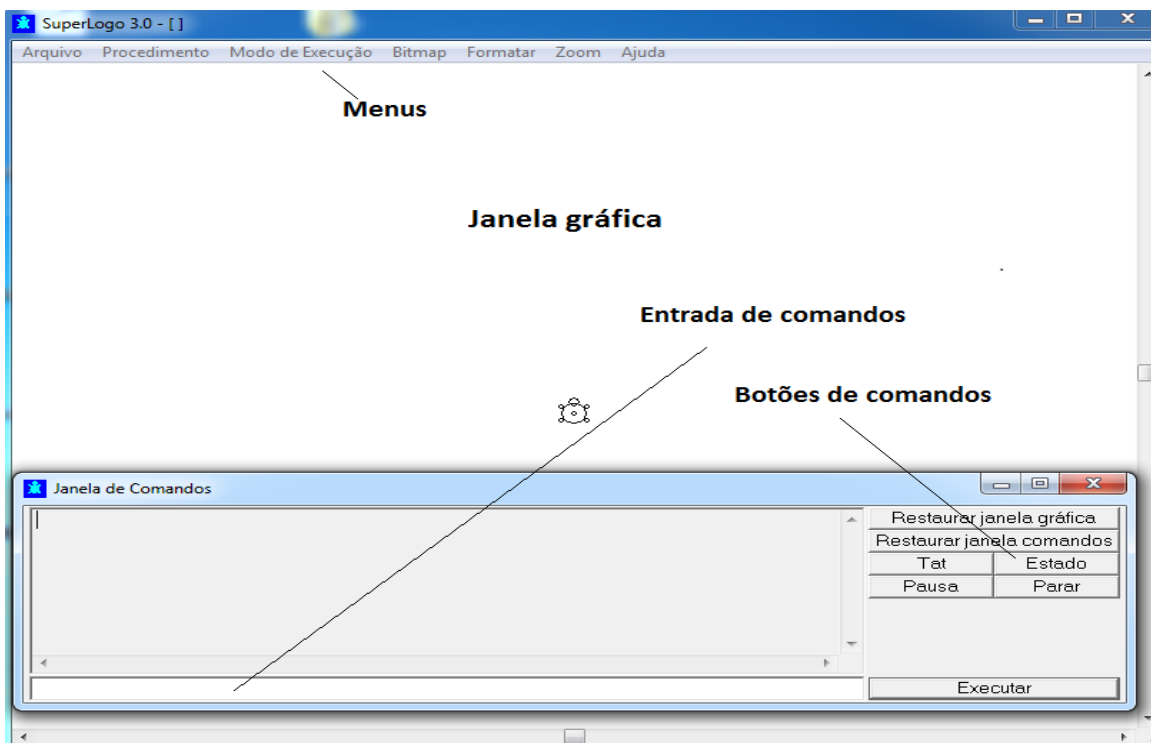


Figura 3.1 A Tela do SuperLogo 3.0

3.2 Principais Comandos

COMANDO	AÇÃO
PF 100	Movimenta a tartaruga 100 passos para frente (posição)
PT 200	Movimenta a tartaruga 200 passos para trás (posição)
PD 90	Gira a tartaruga 90 graus à direita (orientação)
PE 70	Gira a tartaruga 70 graus à esquerda (orientação)
TAT	Limpa a tela e posiciona a tartaruga na sua posição inicial (centro da tela - posição [0,0] no eixo de coordenadas cartesianas) e orientação (0° em relação a um eixo imaginário) originais.
UN	Permite movimentar a tartaruga sem desenhar a trajetória na tela
UB	Permite movimentar a tartaruga apagando traços que estejam na sua trajetória
UL	Permite movimentar a tartaruga desenhando a trajetória na tela
DT	Torna a tartaruga invisível
AT	Torna a tartaruga visível
MUDECL N.º	Muda a cor do traço
MUDECF N.º	Muda a cor do fundo
REPITA N.º	Usado para repetir uma ação ou seqüência de ações num determinado n.º de vezes.
PINTE	Comando usado para pintar o interior de uma região – para isso a tartaruga deve ser movimentada para dentro da região
+	Operação de adição
-	Operação de subtração
*	Operação de multiplicação
/	Operação de divisão

Tabela 3.1 Principais comandos do SuperLogo

3.3 Menus

Os menus do SuperLogo 3.0 são auto-explicativos e de fácil assimilação. No menu “Ajuda” opção “índice” estão listados todos os comandos do SuperLogo 3.0. Ao selecionar um dos comandos pode-se ter acesso à sua sintaxe. O índice é uma espécie de manual interno do software.

Os procedimentos que definimos após iniciar a nossa sessão Logo estão guardados apenas na memória do computador. Se terminarmos o programa Logo, todos eles serão perdidos.

Para salvá-los em um arquivo em disco, vamos para a barra de menu e selecionemos *Arquivo / Salvar como*. Na caixa de diálogo que se abrir, escolhamos o disco, a pasta onde desejamos salvar nossos procedimentos e o nome do arquivo que irá contê-los. Notemos que todos os procedimentos que estão na memória irão para esse arquivo. Se desejamos guardar algum procedimento, temos que apagá-lo da memória antes de salvar o resto. Isto é feito indo para *Procedimento / Apagar* na barra de menu e selecionando, na caixa de diálogo que se abrir, os procedimentos que serão descartados. Em uma próxima sessão Logo, os procedimentos salvos no arquivo podem ser carregados novamente para a memória, usando o menu *Arquivo / Abrir*.

3.4 Programando Computadores

Um procedimento computacional, ou programa de computador, é uma sequência de instruções escrita numa linguagem de programação de computadores e executado por um computador.

Programar o computador é ensiná-lo novas palavras. A versão do *Logo* que utilizaremos em nosso curso, o *SuperLogo* para *Windows*, possui um vocabulário de aproximadamente 273 palavras. Estas palavras que o computador “entende” são chamadas de *palavras primitivas* da linguagem *Logo*. Em outras linguagens de programação elas são, normalmente, chamadas de *comandos primitivos*.

Existem algumas maneiras de se ensinar uma nova palavra ao computador. Uma delas é através do uso da *palavra primitiva aprenda*. Uma outra maneira de ensinar uma nova palavra é através da palavra *edite*. Também podemos, na janela principal do Superlogo, clicar em Procedimento que abrirá o Editor, conforme vemos na Figura 3.2.

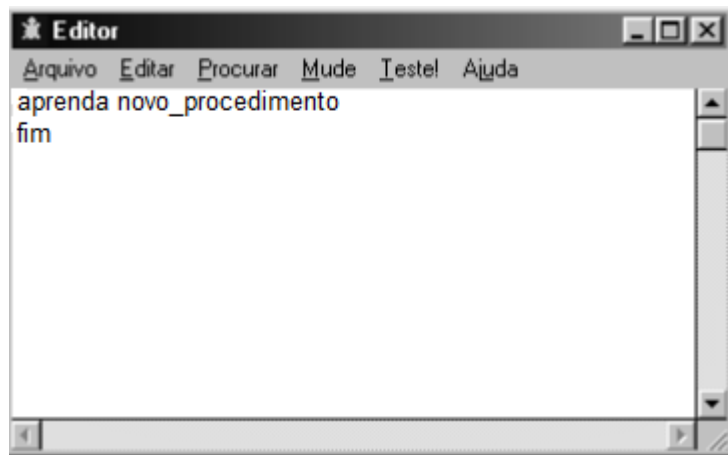


Figura 3.2 Editor de Procedimento

As “novas” *palavras definidas* podem ser usadas da mesma forma que as *palavras primitivas* para definir outras palavras novas. Isto não é incrível. Uma máquina possui capacidade de aprendizagem infinita!

O que chamamos de “ensinar uma nova palavra à tartaruga” é a mesma coisa que, nos livros de computação, se chama de “definir um procedimento” ou “fazer um programa”. Vamos definir alguns exemplos de procedimentos numéricos com variáveis.

PROCEDIMENTO 1 - Progressão Aritmética

Definição 1: Progressão aritmética (PA) é toda sequência na qual cada termo, a partir do segundo, é a soma do termo anterior com uma constante dada.

<pre> aprenda pa :n :pt :r repita :n [mostre :pt atribua "pt :pt+:r] fim </pre>
<p>Por exemplo, na janela de comandos digite pa 5 -2 4</p>

Tabela 3.2

Em que:

:*n* é o número de termos da PA que se mostrará;

:*pt* é o primeiro termo da PA

:*r* é a razão da PA

Execute o procedimento digitando na linha de comandos, por exemplo, pa 10 3 5 obtendo como resposta a seguinte sequência -2 2 6 10 14

PROCEDIMENTO 2 - Progressão Geométrica.

Definição 2: Progressão geométrica (PG) é toda sequência na qual cada termo, a partir do segundo, é o produto do termo anterior por uma constante dada q .

```
aprenda pg :n :pt :q
repita :n [mostre :pt atribua "pt :pt*:q]
fim
```

Por exemplo, na janela de comandos digite pg 7 13 7

Tabela 3.3

Em que:

:*n* é o número de termos da PG que se mostrará;

:*pt* é o primeiro termo da PG

:*q* é a razão da PG

Execute o procedimento digitando na linha de comandos, por exemplo, pg 7 13 7 obtendo como resposta a seguinte sequência 13 91 637 4459 31213 218491.

Agora, ao invés de 273 palavras a linguagem *Logo* possui 275 palavras com pa e pg.

PROCEDIMENTO 3 – Polígono regular

Abaixo apresentamos uma procedure que desenha um polígono regular de *n* lados e calcula o perímetro, número de diagonais e a soma dos ângulos internos.

```
aprenda poligonoregular :n :l
repita :n[pf :l pd 360/:n]
escreva [Perímetro:]
escreva :n*:l
escreva [Número de diagonais:]
escreva :n*(n-3)/2
escreva [Soma dos ângulos internos:]
escreva (:n-2)*180
fim
```

Por exemplo, na janela de comandos digite, poligonoregular 5 100

Tabela 3.4

Em que:

:*n* é o número de lados do polígono regular;

:*l* é o comprimento do lado do polígono regular

Na Janela Gráfica, obtemos:

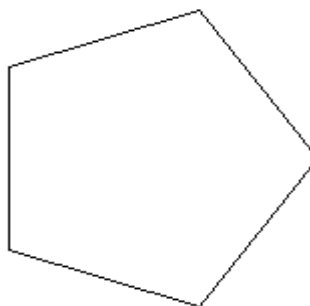


Figura 3.3 polígono regular

Na Janela de Comandos, obtemos:

Perímetro:

500

Número de diagonais:

5

Soma dos ângulos internos:

540

3.5 Sequências definidas recursivamente

Uma *sequência* é uma lista de objetos dispostos segundo uma determinada ordem. Ou seja, numa sequência existe um primeiro elemento, um segundo e assim por diante. Podemos definir recursivamente uma sequência explicitando o valor do(s) primeiro(s) elemento(s) e definindo os demais elementos a partir deles. Por exemplo, a sequência P que tem como cinco primeiros elementos os números 1, 3, 6, 10, 15, pode ser definida através do procedimento recursivo para fornecer o n -ésimo elemento como abaixo:

```
aprenda seqP :n
se :n=1 [envie 1]
envie :n+seqP :n-1
fim
```

Por exemplo, na janela de comandos digite, escreva seqP 5

Tabela 3.5

Como podemos observar, o procedimento seqP 5 é, praticamente, a transcrição da definição matemática recursiva dada anteriormente.

Apresentamos abaixo o procedimento recursivo da sequência de Fibonacci.

```
aprenda seqF :n
se ou :n=1 :n=2 [envie 1]
envie (seqF :n-2) + (seqF :n-1)
fim
```

Por exemplo, na janela de comandos digite, escreva seqF 15

Tabela 3.6

Resposta: 610

3.6 Figuras Definidas Recursivamente

Nesta parte, apresentamos exemplos de fractais utilizando o Superlogo para traçar o fractal e calculamos alguns de seus dados como, área, lado e dimensão.

Como visto anteriormente, um fractal é uma figura que pode ser quebrada em pequenos pedaços, sendo cada um desses pedaços uma reprodução do todo. Não podemos ver um fractal porque é uma figura limite, mas as etapas de sua construção

podem dar uma ideia da figura toda. Seu nome se deve ao fato de que a dimensão de um fractal não é um número inteiro.

Algumas curvas da geometria podem ser mais facilmente definidas pela estratégia recursiva. O exemplo clássico é o da famosa curva de Koch descoberta em 1904. Ela é gerada a partir de duas formas que chamamos de *iniciador* e *gerador* mostradas a seguir:

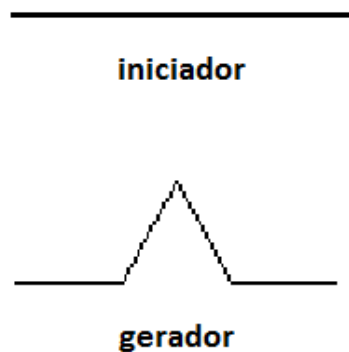


Figura 3.4

Para definir um procedimento que desenhe a curva de Koch podemos adotar a seguinte estratégia. Inicialmente definimos um procedimento que desenhe o gerador, como mostrado a seguir:

```
aprenda koch :t
pd 90
pf :t/3
pe 60
pf :t/3
pd 120
pf :t/3
pe 60
pf :t/3
fim
```

Para chamar a procedure, por exemplo, digite koch 200 e em seguida Executar.

Tabela 3.7

O processo de construção se inicia pela substituição de cada segmento de reta do iniciador, no caso apenas um, pelo gerador. Obtemos, após esta substituição, o primeiro estágio da construção da curva de Koch, mostrado na figura seguinte, que, nesse caso se assemelha ao gerador.



Estágio 1

Figura 3.5

O segundo estágio é obtido pelo mesmo processo aplicado agora à curva obtida no 1º estágio. Ou seja, cada segmento de reta do estágio 1 é substituído por uma cópia do gerador na devida escala para que sejam mantidos os mesmos pontos inicial e final. A curva obtida no segundo estágio da geração da curva de Koch é mostrada abaixo.



Estágio 2

Figura 3.6

Repetindo-se o mesmo processo indefinidamente geramos a curva de Koch.

Depois de testar o procedimento definido para verificar se ele realmente desenha o gerador, substituiremos cada segmento de reta do nosso procedimento (pf :t/3) por uma chamada recursiva (kochg :t/3 :n-1) tomando o cuidado de estabelecer a condição de parada do nosso procedimento, como podemos ver abaixo:

```
aprenda kochg :t :n
Se :n=0 [pf :t pare]
kochg :t/3 :n-1
pe 60
kochg :t/3 :n-1
pd 120
kochg :t/3 :n-1
pe 60
kochg :t/3 :n-1
fim
```

Para chamar a procedure, por exemplo, digite kochg 100 3 e em seguida Executar.

Tabela 3.8

Vamos organizar alguns dos dados em uma tabela:

Nível	0	1	2	3	N
Número de segmentos	1	4	16	64	4^n
Comprimento de cada segmento	1	$\frac{1}{3}$	$\frac{1}{9}$	$\frac{1}{27}$	$\frac{1}{3^n}$
Comprimento total da curva	1	$\frac{4}{3}$	$\frac{16}{9}$	$\frac{64}{27}$	$\left(\frac{4}{3}\right)^n$

Tabela 3.9: Comprimento da curva de Koch

A curva de Koch recebeu este nome por ter sido proposta, em 1904, pelo matemático sueco *Helge von Koch*. Na época, a curva de Koch foi rejeitada pelos matemáticos, pois durante séculos, eles consideravam como legítimas somente as curvas que possuíssem tangentes bem definidas em cada ponto. Além disso, consideravam, platonicamente, que as coisas da natureza deveriam se aproximar desse tipo de curva suave, sem pontos de fratura. Mas, a curva que acabamos de construir tem infinitos pontos de fratura. Por isso, a curva de Koch foi chamada de “monstro matemático” e de “curva patológica”. Mandelbrot foi o primeiro a questionar esta posição platônica, afirmando que objetos matemáticos complexos, como a curva de Koch, que ele chamou de curvas fractais, são a regra e não a exceção na matemática e na natureza.

Características de um Fractal

Em nosso estudo, estamos considerando como Fractais objetos que podem ser obtidos geometricamente através de processos recursivos apresentando determinadas características que por vezes são encontradas em formas da natureza. Que características são essas? Essas características são: autossimilaridade, escala, complexidade e dimensão.

Uma figura é autossimilar se apresenta sempre o mesmo aspecto visual a qualquer escala que seja ampliada ou reduzida, ou seja, se parte de uma figura assemelha à figura vista como um todo.

A autossimilaridade pode ser notada em figuras geradas por processos matemáticos em que, o conjunto total é formado por pequenas réplicas perfeitas delas mesmas, ou seja, é formado através de um processo iterativo como é o caso, por exemplo, do triângulo e do tapete de Sierpinski e da curva de Koch.

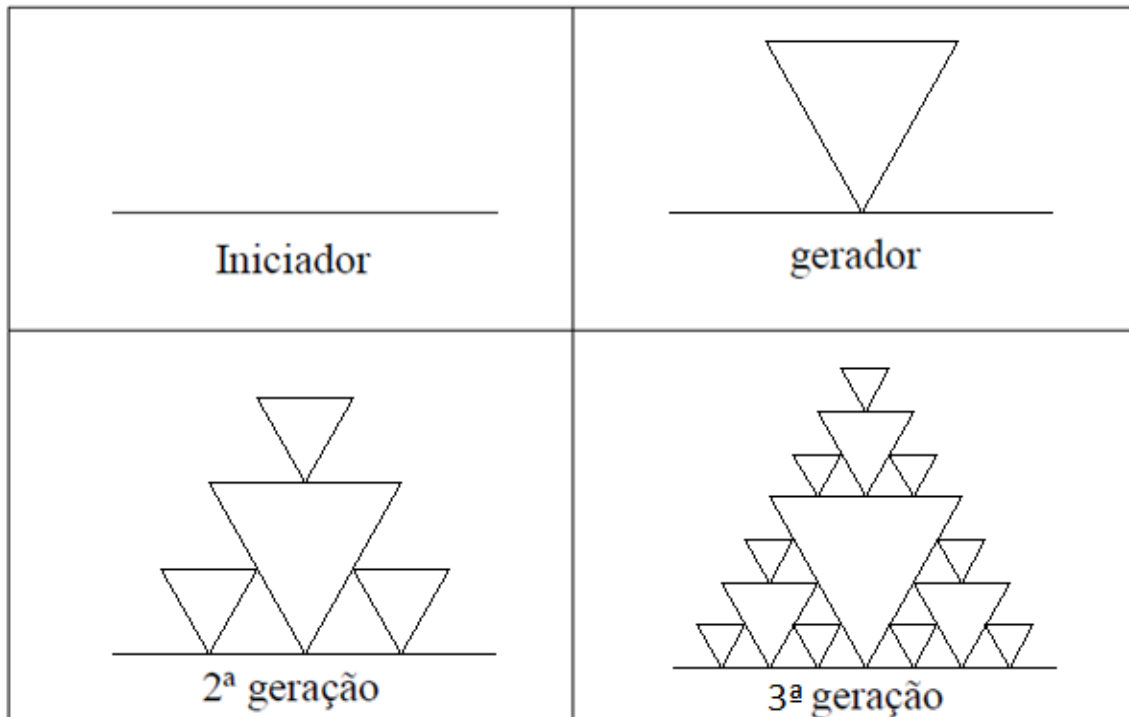


Figura 3.7 Fractal construído no Super-Logo

Esta figura retrata a construção de um fractal conhecido como Triângulo de Sierpinski, (Waclaw Sierpinski, matemático polonês, 1882-1969) descrito na figura acima.

Esta característica (autossimilaridade) resulta do fato das figuras serem construídas pela iteração da mesma regra de construção, indefinidamente.

A iteração é a repetição de um procedimento consecutivamente. Vejamos o procedimento de construção da sequência iterativa do Triângulo de Sierpinski.

Iniciamos com um segmento de reta de comprimento t e de seu ponto central construímos um triângulo equilátero de comprimento $t/2$ com vértice nesse ponto central. Ficamos, portanto, com a figura geradora. A 2ª geração ou iteração é obtida através de uma aplicação da figura geradora a cada um dos segmentos horizontais de comprimento $t/2$ que a constituem. A figura final deste passo de construção é o elemento de construção da figura seguinte (3ª iteração), por aplicação da figura geradora. O processo iterativo consiste em aplicar a mesma regra a cada um dos segmentos horizontais que resultam da iterada anterior, como está ilustrado na figura acima. O processo é repetido (iterado) indefinidamente obtendo-se a figura limite a que chamamos de Triângulo de Sierpinski.

A autossimilaridade é um elemento integrante do processo de construção. Cada triângulo preenchido contido na iterada k , está reduzido por um fator de 3 em relação aos triângulos preenchidos que compõem a iterada anterior, e cada um dos quatro

triângulos que compõem a iterada k , é uma cópia reduzida por um fator 3 de toda a estrutura da iterada imediatamente anterior.

Assim, o objeto que resulta da passagem ao limite deste processo recursivo é exatamente autossimilaridade, ou seja, é igual a uma cópia ampliada dos elementos que o constituem.

Os fractais são formados por um processo recursivo aplicado indefinidamente. Quanto maior for o número de iterações deste processo, mais detalhes serão apresentados e assim nunca obteremos uma *imagem final*. Daí a expressão complexidade infinita. Dizemos por isso que o triângulo de Sierpinski e cada um dos exemplos acima são os limites do processo iterativo.

Apresentamos o procedimento que gera o Triângulo de Sierpinski:

<pre> aprenda Siertri :t :n se :n=0 [pf :t pare] Siertri :t/2 :n-1 pe 120 pf :t/2 pd 120 Siertri :t/2 :n-1 pd 120 pf :t/2 pe 120 Siertri :t/2 :n-1 Fim </pre>
<p>Para chamar a procedure, por exemplo, digite Siertri 200 50 e em seguida Executar.</p>

Tabela 3.10

Construindo exemplos de curvas geradas como a curva de Koch.

Exemplo 1.

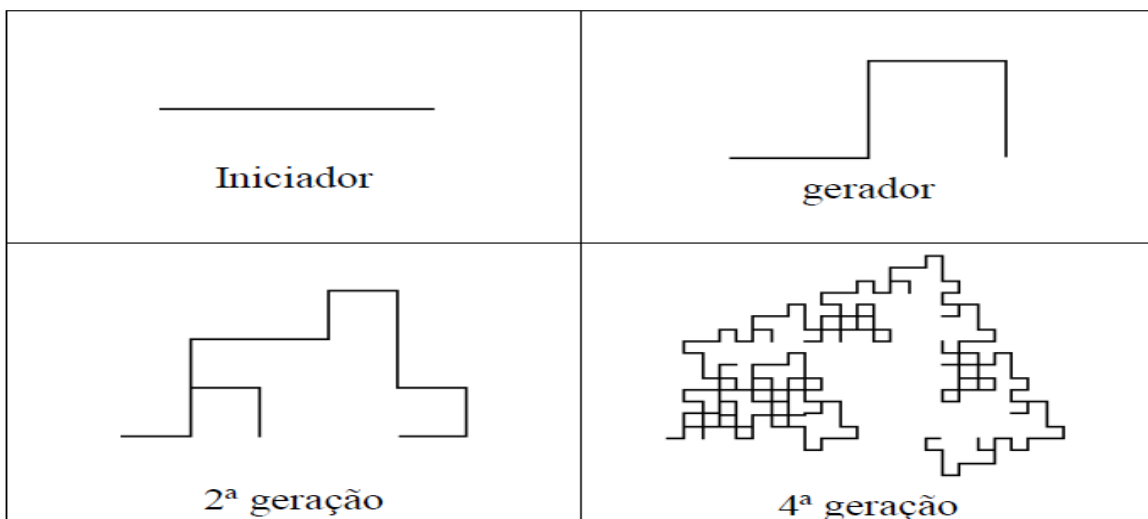


Figura 3.8 Fractal construído no Super-Logo

Código do Exemplo 1

```

aprenda koch2 :t :n
se :n=0 [pf :t pare]
koch2 :t/2 :n-1
pe 90
koch2 :t/2 :n-1
pd 90
koch2 :t/2 :n-1
pd 90
koch2 :t/2 :n-1
pe 90
koch2 :t/2 :n-1
fim

```

Para chamar a procedure, por exemplo, digite koch2 50 3 e em seguida Executar.

Tabela 3.11

Exemplo 2.

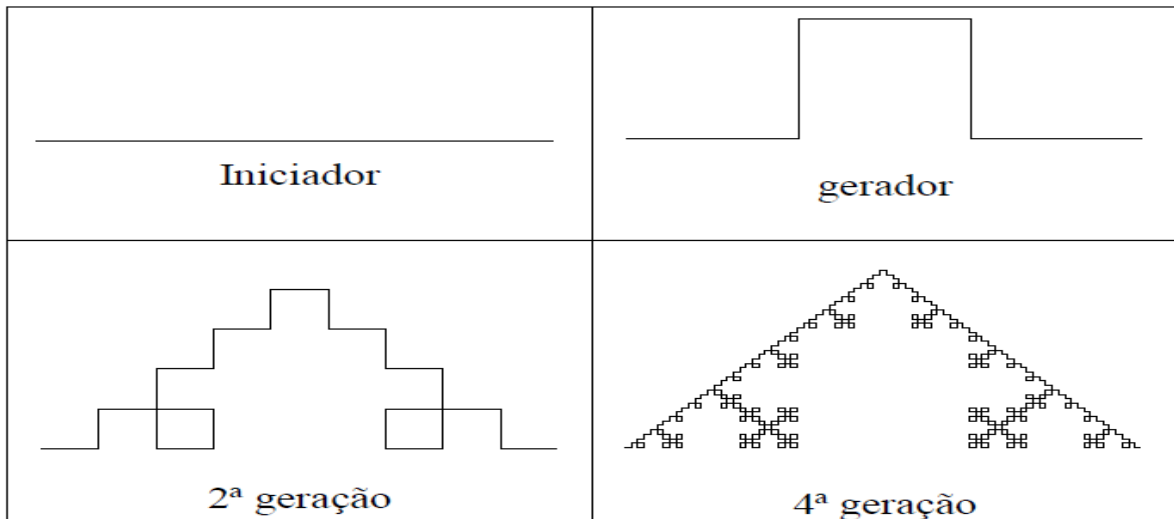


Figura 3.9 Fractal construído no Super-Logo

Código do Exemplo 2

```

aprenda koch3 :t :n
se :n=0 [pf :t pare]
koch3 :t/3 :n-1
pe 90
koch3 :t/3 :n-1
pd 90
koch3 :t/3 :n-1
pd 90
koch3 :t/3 :n-1
pe 90
koch3 :t/3 :n-1
fim

```

Para chamar a procedure, por exemplo, digite koch3 50 3 e em seguida Executar.

Tabela 3.12

Exemplo 3. Conjunto de Cantor, (George Cantor 1845-1918).

Em 1883, Cantor, publicou um trabalho no qual é construído um conjunto, chamado hoje “Conjunto de Cantor” (às vezes “*Polvo de Cantor*” ou “*Poeira de Cantor*”), como exemplo de conjunto excepcional, um dos “monstros matemáticos”

A construção do Conjunto de Cantor é simples e parte de um segmento de reta, compreendido entre os valores de 0 a 1. Começa por retirar o terço médio, resultando em dois segmentos de extensão $1/3$. Logo, é retirado o terço médio destes dois segmentos, ficando quatro segmentos com extensão de $1/9$. O processo é repetido indefinidamente, com tendência para o infinito, chegando a um ponto onde o resultado final é uma sucessão de pontos, também chamada de “poeira de cantor”.

O conjunto numérico de Cantor na sua fase inicial, a), corresponde ao intervalo fechado $[0,1]$. Na fase b) teremos dois intervalos fechados: $[0,1/3]$ e $[2/3,1]$; e em c) teremos quatro: $[0,1/9]$, $[2/9,1/3]$, $[2/3,7/9]$ e $[8/9,1]$. Na fase d) obteremos oito intervalos fechados: $[0,1/27]$, $[2/27,1/9]$, $[2/9,7/27]$, $[8/27,1/3]$, $[2/3,19/27]$, $[20/27,7/9]$, $[8/9,25/27]$ e $[26/27,1]$ e assim sucessivamente.

Mas, afinal, qual é o conjunto de Cantor? A resposta, em linguagem simples, poderia ser: - É o conjunto de pontos (números) que permanecem após as infinitas fazes.

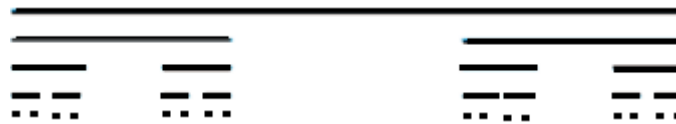


Figura 3.10

Gerando o Conjunto de cantor

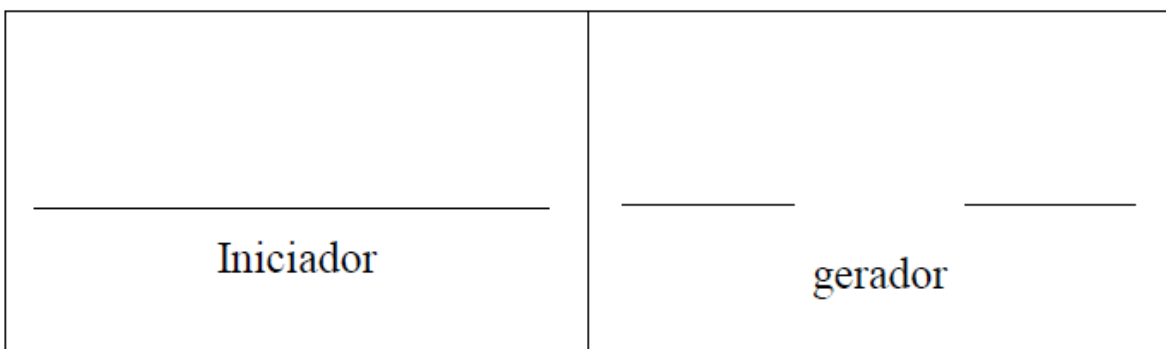


Figura 3.11

Código do conjunto de cantor:

```

aprenda cantor :t :n
se :n=0 [pf :t pare]
cantor :t/3 :n-1
un
pf :t/3
ul
cantor :t/3 :n-1
fim

```

Para chamar a procedure, por exemplo, digite cantor 40 30 e em seguida Executar.

Tabela 3.13

Vamos organizar alguns dos dados em uma tabela:

Etapa	Nº de segmentos	Comprimento de um segmento	Comprimento total
0	1	1	1
1	2	1/3	2/3
2	2 ²	1/3 ²	2 ² / 3 ²
3	2 ³	1/3 ³	2 ³ / 3 ³
<i>n</i>	2 ^{<i>n</i>}	1/3 ^{<i>n</i>}	2 ^{<i>n</i>} / 3 ^{<i>n</i>}

Tabela 3.14

Exemplo 4. Curvas que preenchem um quadrado

O aparecimento das curvas que preenchem o quadrado foi muito importante para o desenvolvimento do conceito de dimensão. Em meados do ano 1891, Peano e Hilbert discutiram estas curvas questionando a sua percepção intuitiva, ou seja, dada uma parte de um plano (bidimensional) há uma curva (unidimensional) que encontra, pelo menos uma vez, todos os pontos desse plano durante o seu percurso.

Em 1890, tratando do aprofundamento das noções de continuidade e dimensão, Peano publica a sua famosa curva (Curva de Peano), (Giuseppe Peano, matemático italiano, 1858-1932), outro monstro matemático, proposta como cobrindo totalmente uma superfície plana quadrangular: - “*Sur une courbe qui remplit toute une aire plane*”, a qual será nosso objeto de estudo.

Para construção da curva de Peano também usamos um processo recursivo. Começamos com um pequeno segmento de reta, por exemplo, com uma unidade de comprimento. Depois de dividirmos o segmento de reta em três sub-segmentos iguais, construímos um retângulo sobre o sub-segmento intermédio ficando com dois quadrados de lado igual a cada um dos sub-segmentos. Obtemos, portanto, uma curva geradora com 9 sub-segmentos, tal como apresentamos na figura abaixo.

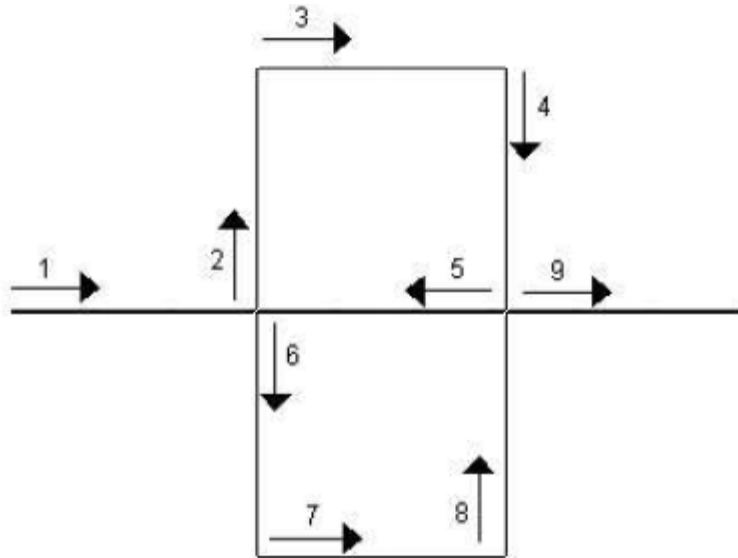


Figura 3.12 <http://www.frattali.it/peano.htm>

A seguir vemos o procedimento para gerar a Curva de Peano

```

aprenda peano :t :n
se :n=0 [pf :t pare]
peano :t/3 :n-1
pe 90
peano :t/3 :n-1
pd 90
peano :t/3 :n-1
pd 90
peano :t/3 :n-1
pd 90
peano :t/3 :n-1
pe 90
peano :t/3 :n-1
pe 90
peano :t/3 :n-1
pd 90
peano :t/3 :n-1
pd 90
fim

```

Para chamar a procedure, por exemplo, digite peano 150 2 e em seguida Executar.

Tabela 3.15

Vamos organizar alguns dos dados em uma tabela:

Nível	0	1	2	3	4	n
Número de segmentos	1	9	81	729	6561	9^n
Comprimento de cada segmento	1	$\frac{1}{3}$	$\frac{1}{9}$	$\frac{1}{27}$	$\frac{1}{81}$	$\left(\frac{1}{3}\right)^n$
Comprimento total da curva	1	3	9	27	81	3^n

Tabela 3.16. Estudo da curva de Peano

Exemplo 6. Curva de Hilbert

A curva de Hilbert foi apresentada por David Hilbert (matemático da antiga Prússia Oriental, 1862-1943) e tal como a curva de Peano é construída através de um processo recursivo mas com algumas particularidades. A figura inicial é um quadrado unitário, como podemos ver na figura XX e a figura geradora consiste em dividi-lo em quatro quadrados iguais, unindo os pontos centrais de cada um desses quadrados, tal como podemos ver na figura YY. A curva é formada, não pelos quadrados mas sim pelos segmentos de reta formados pela ligação desses pontos centrais. Os sucessivos passos, são construídos utilizando o processo anterior, como está descrito na figura ZZ.

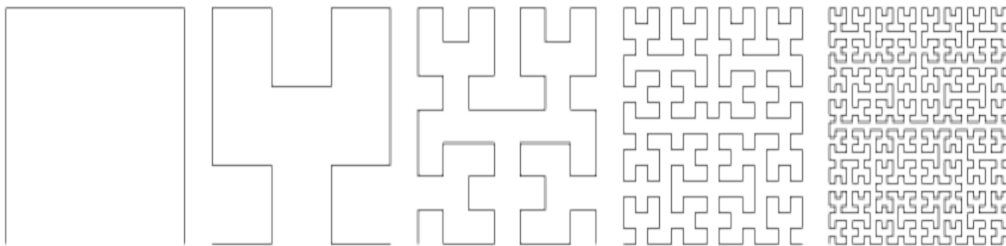


Figura 3.13. <http://mathworld.wolfram.com/HilbertCurve.html>

O fractal de Hilbert é um tanto mais complexo. Abaixo, apresento uma tentativa de solução. Deve-se escrever na janela de comandos algo como hilbert distância nível direção, em que distância é o número de pixels que corresponde aos passos da tartaruga, nível é o número de iterações e direção é o valor 1 ou -1.

A seguir vemos o procedimento para gerar a Curva de Hilbert:

```
aprenda hilbert :t :n :d
se :n>0 [pd -90 * :d
hilbert :t :n-1 (-1) *:d pf :t/4 pd 90 * :d
hilbert :t :n-1 :d pf :t/4
hilbert :t :n-1 :d pd 90 * :d pf :t/4
hilbert :t :n-1 (-1) *:d pd -90 *:d]
fim
```

Exemplo: na janela de comandos, escreva **hilbert 50 4 1**

Tabela 3.17

Vamos organizar alguns dos dados em uma tabela:

Nível	-1	0	1	2	3	n
Número de segmentos (N_n)	0	3	15	63	255	$4^{n+1} - 1$
Comp. de cada segmento	-	1	$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{8}$	$\frac{1}{2^n}$
Comp. total da curva	-	3	$\frac{15}{2}$	$\frac{63}{4}$	$\frac{225}{8}$	$\frac{4^{n+1} - 1}{2^n}$
Lado do quadrado auxiliar	-	$1 + \frac{1}{2}$	$\frac{3}{2} + \frac{1}{4}$	$\frac{7}{4} + \frac{1}{8}$	$\frac{15}{8} + \frac{1}{16}$	$\sum_{k=0}^{n+1} \frac{1}{2^k}$

Tabela 3.18. Segmentos em cada nível de construção da curva de Hilbert

3.7 Dimensão euclidiana e fractal

Sabemos que o espaço em que vivemos é de dimensão 3, é nossa dimensão espacial. A dimensão euclidiana tem origem na geometria euclidiana plana que se encontra descrita nas famosas obras de Euclides e que foram objeto de estudo de muitos matemáticos.

Neste trabalho, assumimos que:

- um ponto possui dimensão zero;
- uma linha reta possui dimensão um;
- uma superfície plana tem dimensão dois;
- um sólido, dimensão três.

e, por indução, pode ampliar-se sucessivamente até n dimensões. Para Euclides, todas as formas da natureza podiam ser reduzidas a formas geométricas simples como quadrados, circunferências, etc. No entanto, a geometria euclidiana era insuficiente para explicar e descrever fenômenos ditos como monstros matemáticos, como é o caso, por exemplo, das curvas que preenchem o espaço. Como é que uma curva, de dimensão 1, podia preencher o espaço de um quadrado, de dimensão 2?

Observemos que um segmento de reta, um quadrado, ou um cubo, respectivamente de dimensões 1, 2 e 3, possuem a propriedade de auto similaridade. As três figuras podem ser repartidas em objetos autos similares:

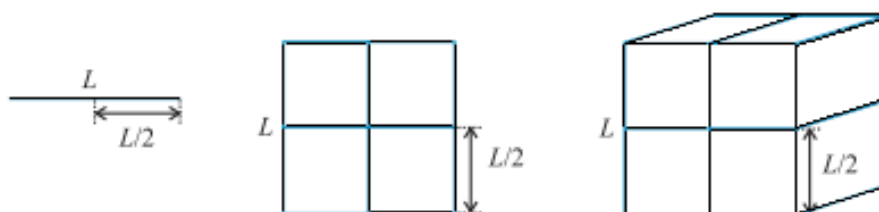


Figura 3.14

Nas figuras anteriores ilustramos as divisões:

- a) de um segmento de reta em 5 peças;
- b) de um quadrado em 9 peças quadrangulares congruentes, repartindo o lado em 3 ;
- c) de um cubo em 8 peças cúbicas, tendo dividido cada aresta em 2.

Cada pequena peça é auto similar ao todo, assim, para que cada peça fique igual ao todo devemos ampliá-la por uma escala ou fator de aumento (coeficiente de proporcionalidade) igual respectivamente a 5, a 3 e a 2.

Em resumo, o número de peças em cada caso é igual:

- a) ao fator de aumento (5^1);
- b) ao *quadrado* do fator de aumento (3^2);
- c) ao cubo do fator de aumento (2^3).

E em geral, o número n de peças é dado por $n = m^d$, em que m é o fator de aumento e d a dimensão.

Portanto, chamamos de dimensão fractal, a seguinte fórmula:

Dimensão=log(número de peças)/log(fator de aumento) ou $d = \frac{\log n}{\log m}$.
--

Exemplificaremos com alguns fractais dados acima.

1) Curva de Koch

Para esse fractal temos $n=4$ peças e fator de aumento $m=3$, e assim sua dimensão fractal é:

$$d = \frac{\log 4}{\log 3} \cong \frac{0,60206}{0,47712} \cong 1,262.$$

A dimensão fractal de um objeto mede o seu grau de irregularidade, a sua estrutura e o seu comportamento, seja de uma figura ou de um fenômeno físico, biológico ou social. Os fractais têm dimensões diferentes e próprias de cada imagem.

Por exemplo, ilustraremos para dois fractais do tipo de curvas de Koch com os geradores iniciais dados por:

a) Ponta mais achatada



Figura 3.15 Procedimento construído no Super-Logo

b) Ponta mais alongada



Figura 3.16 Procedimento construído no Super-Logo

Usamos em ambos $n=4$ peças como na curva de Koch padrão, e também ambos com comprimento total do segmento (: $t=350$ pixels) 3,5 unidades. Em a) cada peça tem $2/7$ do total, porém em b) cada uma tem $2/5$, valores respectivamente inferior e superior a cada peça da curva padrão de Koch, que possuem $1/3$ do total.

Com esses dados às respectivas dimensões são:

$$d = \frac{\log 4}{\log 3,5} \cong \frac{0,60206}{0,54407} \cong 1,106.$$

$$d = \frac{\log 4}{\log 2,5} \cong \frac{0,60206}{0,39794} \cong 1,512.$$

Desde que os fatores de aumento são dados pelos inversos dos fatores de redução $1/(2/7)=7/2=3,5$ e $1/(2/5)=5/2=2,5$.

Abaixo construímos e apresentamos os fractais correspondentes com seus respectivos códigos para o nível $n=3$ e concluímos, por observação, que: a) é menos áspero e menos denso, enquanto b) é mais áspero e mais denso.

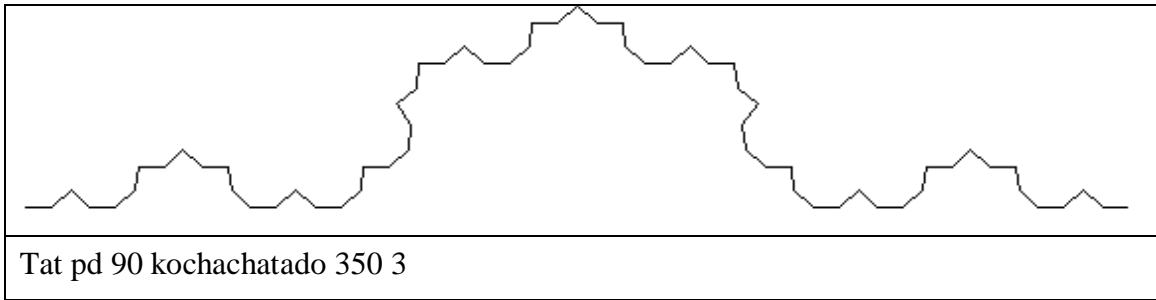


Figura 3.17 Fractal construído no Super-Logo

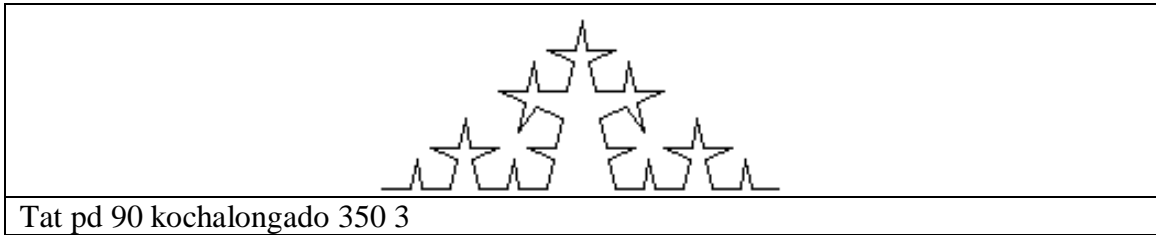


Figura 3.18 Fractal construído no Super-Logo

A seguir vemos o procedimento para gerar a curva de Koch achatada.

```

aprenda kochachatado :t :n
Se :n=0 [pf :t pare]
kochachatado :t/3 :n-1
pe arccos(3/4)
kochachatado :t/3 :n-1
pd 2*arccos(3/4)
kochachatado :t/3 :n-1
pe arccos(3/4)
kochachatado :t/3 :n-1
fim

```

Para chamar a procedure, por exemplo, digite Tat pd 90 kochachatado 350 3

Tabela 3.18

A seguir vemos o procedimento para gerar a curva de Koch alongada.

```

aprenda kochalongado :t :n
Se :n=0 [pf :t pare]
kochalongado :t/3 :n-1
pe arccos(10/49)
kochalongado :t/3 :n-1
pd 2*arccos(10/49)
kochalongado :t/3 :n-1
pe arccos(10/49)
kochalongado :t/3 :n-1
fim

```

Para chamar a procedure, por exemplo, digite Tat pd 90 kochalongado 350 3

Tabela 3.19

2) Conjunto de Cantor

Sabendo que, em cada iteração, ficamos com 2 peças ou segmentos que serão novamente divididos em três partes iguais, e que o coeficiente de redução é $\frac{1}{3}$ então o fator de aumento é $m=3$ e a dimensão do fractal conjunto de Cantor é dada por:

$$d = \frac{\log 2}{\log 3} \cong \frac{0,30102}{0,47712} \cong 0,631.$$

3) Triângulo de Sierpinski

Para esse fractal temos $n=3$ e fator de aumento $m=2$. Assim,

$$d = \frac{\log 3}{\log 2} \cong \frac{0,47712}{0,30103} \cong 1,585.$$

4) Curva de Peano

Sabendo que o coeficiente de redução é $r = 1/3$ e, portanto, o fator de aumento é $m=3$ e que o número de partes obtidas em cada segmento de reta é $n = 9$, temos,

$$d = \frac{\log 9}{\log 3} = 2.$$

5) Curva de Hilbert

Sabendo que o coeficiente de redução é $r = 1/2$ e, portanto, o fator de aumento é $m=2$ e que o número de partes obtidas em cada segmento de reta é $n = 4$, temos,

$$d = \frac{\log 4}{\log 2} = 2.$$

Construção de uma árvore fractal

“Não há ramo da Matemática por mais abstrato que seja que não possa um dia vir a ser aplicado aos fenômenos do mundo real”. Nikolai I. Lobachewsky.

Uma das maravilhas da natureza são as árvores. Em muitas delas o processo de crescimento exponencial é bastante regular, em outras são aproximadamente regulares.

Podemos gerar figuras em forma de árvore mais facilmente se usarmos definições recursivas e a ideia de reescrita de termos. Vejamos a construção de uma árvore em que cada galho vai se bifurcando em 2 menores e diminuindo 0.2 (20%) do anterior até chegar às folhas. Imaginemos, pois, que temos como iniciador uma bifurcação como a mostrada abaixo que pode ser gerada pelo procedimento ao lado para $t=100$ pixels.


	<pre> aprenda arvore :t pf :t pe 20 pf 0.8*:t pt 0.8*:t pd 40 pf 0.8*:t pt 0.8*:t pe 20 pt :t fim </pre>
---	--

Figura 3.19

A ideia é substituir cada um dos dois ramos por uma cópia da própria bifurcação recursivamente. Podemos obter o resultado esperado substituindo os comandos `pf 0.8*:t` e `pt 0.8*:t`, que são os ramos, por chamadas recursivas ao procedimento. Ver procedimento árvore abaixo.

<pre> aprenda arvore :t :n se :n=0 [pare] pf :t pe 20 arvore 0.8*:t :n-1 pd 40 arvore 0.8*:t :n-1 pe 20 pt :t fim </pre>
--

Tabela 3.20

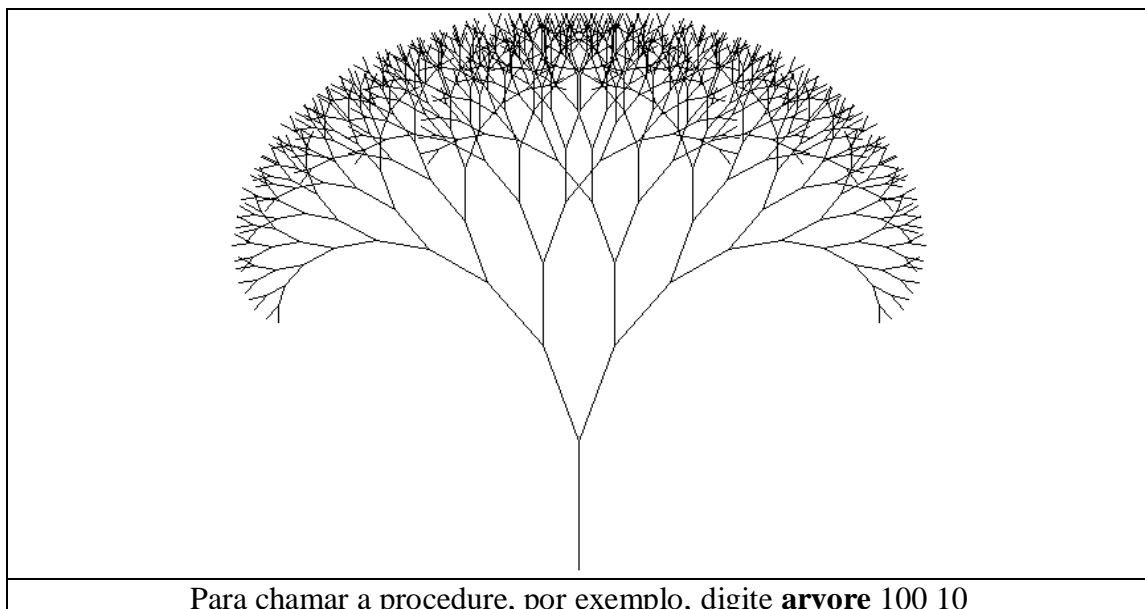


Figura 3.20. Arvore Fractal construída no Super-Logo

Mapa conceitual:

Apresentamos um mapa conceitual inicial do que podemos abordar com os fractais no processo de ensino aprendizagem.

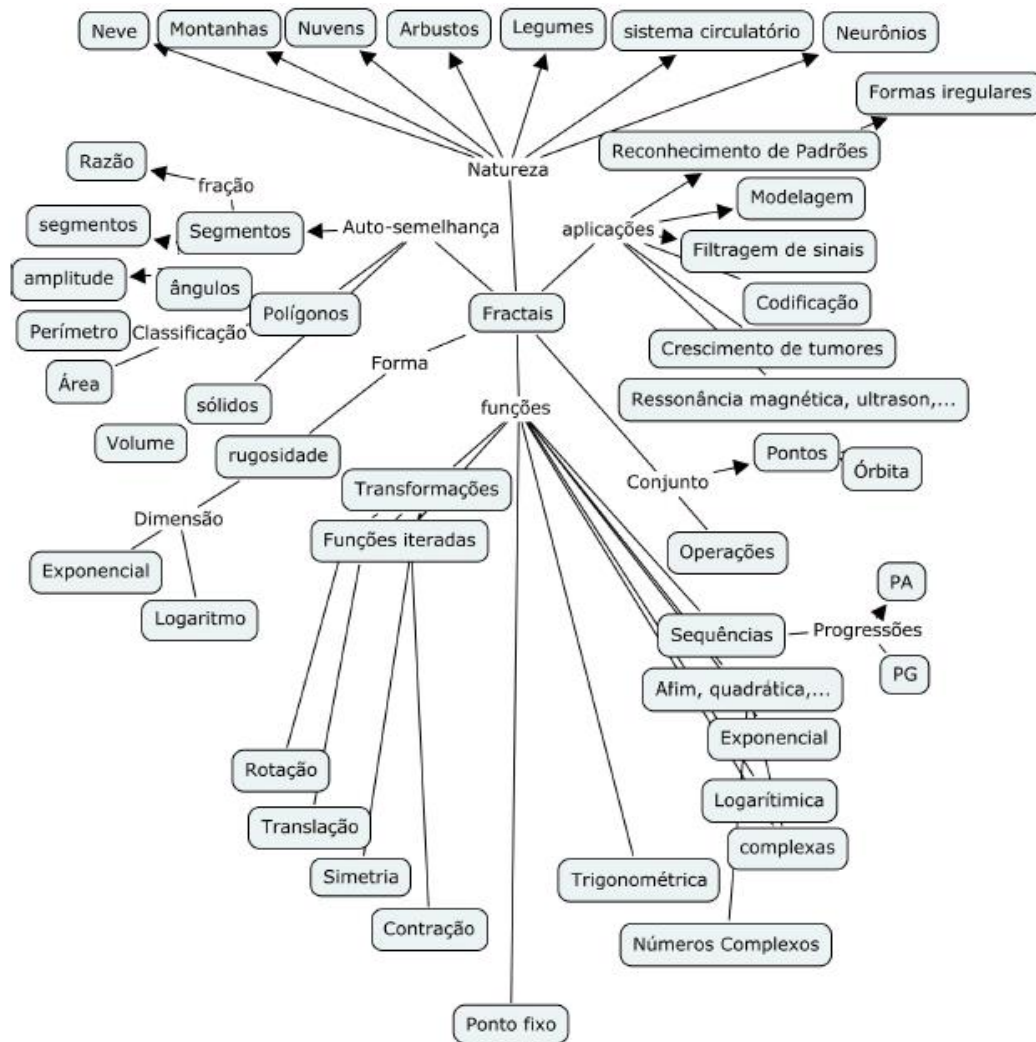


Figura3.21. http://www.avaad.ufsc.br/moodle/mod/hiperbook/view.php?id=2089&pagenum=2&target_navigation_chapter=3710&show_navigation=1

Considerações Finais

A realização deste trabalho de conclusão de curso (TCC) ampliou meus conhecimentos e objetivou o desenvolvimento de atitudes investigativas sobre Geometria Recursiva. A abordagem de conceitos e técnicas básicas apresentadas contribuiu para minha formação e integralizou ainda mais meu curriculum.

O que apresentamos foi apenas uma introdução, uma ideia intuitiva do mundo dos fractais, mas esperamos que tenha ficado claro que, todo aquele que quiser se embrenhar pelo mundo dos fractais, terá que estudar tópicos sobre conjuntos, distâncias, métricas, espaços métricos, pontos fixos, contração, etc, e, com certeza, vai se maravilhar com as estruturas que irá conhecer e descobrir no fantástico mundo da Matemática.

Ressaltamos também que foi um grande aprendizado poder construir e gerar figuras recursivas através do software Superlogo. A linguagem de programação Logo tradicional envolve uma tartaruga gráfica, um robô preparado para responder aos comandos do usuário. O resultado é mostrado após digitar-se o comando, incentivando o aprendizado. Nela, o usuário aprende tentando ensinar a tartaruga a movimentar-se através de tentativa e erro.

Ao utilizar um software para gerar figuras recursivas, ou fazer cálculos recursivos, podemos destacar duas coisas.

Primeiro, suas vantagens:

- Soluções recursivas são, geralmente, mais elegantes que suas contrapartes iterativas;
- Uma função recursiva é menor que sua versão iterativa.

Segundo, as desvantagens:

- Recursão é lenta. Ao chamar uma função, precisamos copiar os parâmetros e inicializar as variáveis. Isso demanda tempo;
- Recursões grandes consomem uma grande quantidade de memória, pois é necessário guardar o estado das funções que estão esperando a próxima retornar;
- Um deslize na definição da condição de parada pode fazer seu programa chamar a função infinitamente. Por isso, sempre se assegure que há uma condição de parada.

Embora não demos uma definição formal, matemática, de fractal, mas as figuras geradas computacionalmente mostram que um fractal é uma figura que pode ser quebrada em pequenos pedaços, sendo cada um desses pedaços uma reprodução do todo. Não podemos ver um fractal porque é uma figura limite, mas as etapas de sua construção podem dar uma ideia da figura toda. Seu nome se deve ao fato de que a dimensão de um fractal não é um número inteiro.

Finalizamos ressaltando que este trabalho pode ser utilizado como material de apoio para análises de outras descobertas sobre o assunto, bem como, trabalhar com materiais concretos no ensino básico. Como mencionado, a introdução de fractais no ensino básico, além de satisfazer a curiosidade de quantos já ouviram falar neles, propicia a oportunidade de trabalhar com processos iterativos, escrever fórmulas gerais, criar algoritmos, calcular áreas e perímetros de figuras com complexidade crescente, introduzir uma ideia intuitiva do conceito de limite e é um excelente tópico para aplicação de progressões geométricas e estímulo ao uso de tabelas.

Sabemos que é um desafio do professor tornar os conceitos matemáticos mais interessantes e motivadores para seus alunos. Sendo assim, esperamos que este TCC possa ter deixado alguns elementos da teoria fractal para exploração de conceitos matemáticos que certamente motivarão o aprendizado dos alunos e os incentivarão a olharem através de um novo horizonte para descobrirem o fantástico mundo das ciências e Matemática.

Apêndice

- **Parte Histórica**

Capítulo 1

Euclides

Euclides de Alexandria (em grego antigo: Εὐκλείδης *Eukleidēs*; fl. c. 300 AC) foi um professor, matemático platônico e escritor possivelmente grego, muitas vezes referido como o "Pai da Geometria". Além de sua principal obra, Os Elementos, Euclides também escreveu sobre perspectivas, seções cônicas, geometria esférica, teoria dos números e rigor.

A geometria euclidiana é caracterizada pelo espaço euclidiano, imutável, simétrico e geométrico, metáfora do saber na antiguidade clássica e que se manteve incólume no pensamento matemático medieval e renascentista, pois somente nos tempos modernos puderam ser construídos modelos de geometrias não-euclidianas.

Benoit Mandelbrot

Benoît B. Mandelbrot (Varsóvia, 20 de novembro de 1924 — Cambridge, 14 de outubro de 2010), foi um matemático francês de origem judaico-polonesa. É conhecido principalmente por suas contribuições no campo da geometria fractal, tendo o termo "fractal" sido por ele cunhado em 1975. Foi aluno do matemático francês Paul Lévy.

Benoit B. Mandelbrot foi um matemático de origem polonesa, com nacionalidade francesa e americana, notável pelo desenvolvimento de uma teoria da rugosidade e da autossimilaridade natureza e no campo da geometria fractal. Descobridor do fractal atualmente conhecido como conjunto de Mandelbrot que consiste de intrincados, intermináveis formas fractais.

Mandelbrot foi um dos primeiros a usar computação gráfica para criar e exibir imagens geométricas fractais, levando à sua descoberta da Mandelbrot em 1979. Ao fazer isso, ele foi capaz de mostrar como a complexidade visual pode ser criado a partir de regras simples. Ele disse que as coisas normalmente consideradas "agitadas", "uma bagunça" ou "caóticas", como nuvens ou costas, na verdade, teve um "grau de ordem". Sua carreira de investigação incluiu contribuições para áreas como geologia, medicina,

cosmologia, engenharia e as ciências sociais. O escritor de Ciência Arthur C. Clarke acredita que o trabalho de Mandelbrot como sendo "uma das descobertas mais surpreendentes de toda a história da matemática".

Leonardo Fibonacci

Leonardo Fibonacci, também conhecido como Leonardo de Pisa, Leonardo Pisano ou ainda Leonardo Bigollo, (Pisa, c. 1170 — Pisa ?, c. 1250); mais reconhecido como Fibonacci foi um matemático italiano nomeado como o primeiro grande matemático europeu da Idade Média. É considerado por alguns como o mais talentoso matemático ocidental da Idade Média. Ficou conhecido pela grande descoberta da sequência de Fibonacci e pela sua participação na introdução dos algarismos arábicos na Europa.

Como outros matemáticos do seu tempo, contribuiu para o renascimento das ciências exatas, após a decadência do último período da antiguidade clássica e do início da Idade Média, mas Fibonacci se destacou ao escrever o *Liber Abaci*, em 1202 (atualizado em 1254), a primeira obra importante sobre matemática desde Eratóstenes, isto é, mais de mil anos antes. O *Liber Abaci* introduziu os numerais hindu-arábicos na Europa, além de discutir muitos problemas matemáticos.

Fibonacci é também conhecido pela sequência numérica nomeada após sua morte como sequência de Fibonacci. Ele não descobriu, mas usou-a como exemplo no *Liber Abaci*.

Capítulo 2

Pitágoras

Pitágoras de Samos em grego: (c. 570 – c. 495 a.C.) foi um filósofo e matemático grego jônico creditado como o fundador do movimento chamado Pitagorismo. A maioria das informações sobre Pitágoras foram escritas séculos depois que ele viveu, de modo que há pouca informação confiável sobre ele. Nasceu na ilha de Samos e viajou o Egito e Grécia e talvez a Índia, em 520 a.C. voltou a Samos. Cerca de 530 a.C., se mudou para Crotona, na Magna Grécia.

Os pitagóricos interessavam-se pelo estudo das propriedades dos números. Para eles, o número, sinônimo de harmonia, é constituído da soma de pares e ímpares (os números pares e ímpares expressando as relações que se encontram em permanente

processo de mutação), sendo considerado como a essência das coisas, criando noções opostas (limitado e ilimitado) e a base da teoria da harmonia das esferas.

Segundo os pitagóricos, o cosmo é regido por relações matemáticas. A observação dos astros sugeriu-lhes que uma ordem domina o universo. Evidências disso estariam no dia e noite, no alterar-se das estações e no movimento circular e perfeito das estrelas. Por isso o mundo poderia ser chamado de cosmos, termo que contém as ideias de ordem, de correspondência e de beleza. Nessa cosmovisão também concluíram que a Terra é esférica, estrela entre as estrelas que se movem ao redor de um fogo central. Alguns pitagóricos chegaram até a falar da rotação da Terra sobre o eixo, mas a maior descoberta de Pitágoras ou dos seus discípulos (já que há obscuridades em torno do pitagorismo, devido ao caráter esotérico e secreto da escola) deu-se no domínio da geometria e se refere às relações entre os lados do triângulo retângulo. A descoberta foi enunciada no teorema de Pitágoras.

Pitágoras foi expulso de Crotona e passou a morar em Metaponto, onde morreu, provavelmente em 496 a.C. ou 497 a.C.. Segundo o pitagorismo, a essência, que é o princípio fundamental que forma todas as coisas é o *número*. Os pitagóricos não distinguem forma, lei, e substância, considerando o número o elo entre estes elementos. Para esta escola existiam quatro elementos: terra, água, ar e fogo.

Assim, Pitágoras e os pitagóricos investigaram as relações matemáticas e descobriram vários fundamentos da física e da matemática.

Capítulo 3

Wallace Feurzeig

Wallace "Wally" Feurzeig (10 de junho de 1927 - 4 de janeiro de 2013), foi co-inventor, com Seymour Papert e Cynthia Solomon, da linguagem de programação Logo, e um conhecido pesquisador em inteligência artificial.

Wallace Feurzeig nasceu em Chicago aos pais Mandel e Pauline Feurzeig. Ele é bacharel em Filosofia e Bacharel em Ciências pela Universidade de Chicago e mestre em Ciências pelo Illinois Institute of Technology. Ele trabalhou no Argonne National Laboratory e na Universidade de Chicago antes de se juntar a Bolt, Beranek e Newman. Durante o início dos anos 1960, a BBN se tornou um importante centro de pesquisa em ciência da computação e aplicativos inovadores. Wally Feurzeig ingressou na empresa

em 1962 para trabalhar com suas novas instalações no Departamento de Inteligência Artificial, uma das primeiras organizações de IA. Seus colegas estavam ativamente engajados em alguns dos trabalhos pioneiros de IA no reconhecimento de padrões computacionais, compreensão de linguagem natural, demonstração de teoremas, desenvolvimento de linguagem LISP e resolução de problemas de robôs.

Daniel Bobrow

Daniel Gureasko Bobrow (29 de novembro de 1935 - 20 de março de 2017) foi um cientista da computação americano que foi pesquisador no Laboratório de Sistemas Inteligentes do Centro de Pesquisa de Palo Alto e criou um programa de inteligência artificial frequentemente citado, com o qual obteve seu PhD.

Nascido em Nova York, ele obteve seu bacharelado pela RPI em 1957, SM em Harvard em 1958 e PhD em Matemática pelo MIT sob a supervisão de Marvin Minsky em 1964.

David Hilbert

David Hilbert (Königsberg, 23 de janeiro de 1862 — Göttingen, 14 de fevereiro de 1943) foi um matemático alemão. Foi eleito membro estrangeiro da Royal Society em 1928.^[2] David Hilbert é um dos mais notáveis matemáticos, e os tópicos de suas pesquisas são fundamentais em diversos ramos da matemática atual.

David Hilbert nasceu em Königsberg, atualmente Kaliningrado, onde estudou na Universidade de Königsberg. Em 1895 foi nomeado professor da Universidade de Göttingen, onde lecionou até se aposentar, em 1930. Está sepultado no Stadtfriedhof de Göttingen. Hilbert é frequentemente considerado como um dos maiores matemáticos do século XX, no mesmo nível de Henri Poincaré. Devemos a ele principalmente a lista de 23 problemas, alguns dos quais não foram resolvidos até hoje, apresentada em 1900 no Congresso Internacional de Matemáticos em Paris.

Suas contribuições à matemática são diversas, e envolvem, entre outras a consolidação da teoria dos invariantes, que foi o objeto de sua tese, a transformação da geometria euclidiana em axiomas, com uma visão mais formal que Euclides, para torná-la consistente, publicada no seu *Grundlagen der Geometrie (Fundamentos da geometria)*, trabalhos sobre a teoria dos números algébricos, a criação dos espaços que levam seu nome, durante seus trabalhos em análise sobre equações integrais e

contribuição para as formas quadráticas, que serviriam como bases matemáticas da teoria da relatividade de Albert Einstein.

Seymour Papert

Seymour Papert (Pretória, 1 de Março de 1928 – Blue Hill, Maine, 31 de julho de 2016) foi um matemático e proeminente educador estadunidense nascido na África do Sul. Lecionava no Massachusetts Institute of Technology (MIT).

Papert estudou na Universidade de Witwatersrand, graduado em 1949 e obteve um PhD em matemática em 1952. Recebeu outro título de PhD, também em matemática, na Cambridge University em 1959, onde foi orientado por Frank Smithies.

Ele foi o teórico mais conhecido sobre o uso de computadores na educação, um dos pioneiros da inteligência artificial e criador da linguagem de programação LOGO (em 1967), inicialmente para crianças, quando os computadores eram muitos limitados, não existia a interface gráfica e muito menos a internet.

Helge von Koch.

Niels Fabian Helge von Koch (Estocolmo, 25 de janeiro de 1870 – Danderyd, 11 de março de 1924) foi um matemático sueco, que deu seu nome ao famoso fractal conhecido como o floco de neve Koch, que foi um dos primeiros fractais de curvas a ser descrito.

Von Koch passou algum tempo na Universidade de Uppsala em 1888. Foi aluno de Mittag-Leffler na Universidade de Estocolmo. Em 1891, ele escreveu o primeiro dos dois documentos sobre aplicações de determinantes infinitos para resolver sistemas de equações diferenciais com coeficientes analíticos. Os métodos que ele utilizou tiveram por base os publicados por Henri Poincaré (1854 - 1912) cerca de seis anos antes. O segundo trabalho de von Koch foi publicado em 1892, ano em que von Koch obteve seu doutorado defendendo sua tese que continha os resultados dos dois trabalhos. Von Koch recebeu o doutorado em Matemática pela Universidade de Estocolmo em 26 de Maio de 1892.

Waclaw Sierpinski

Wacław Sierpiński (Varsóvia, 14 de março de 1882 — Varsóvia, 21 de outubro de 1969) foi um matemático polonês.

Apesar de dificuldades impostas pela ocupação da Polônia pelo Império Russo, Sierpiński entrou para o departamento de física e matemática da Universidade de Varsóvia em 1899.

Vários objetos matemáticos foram denominados em honra a Sierpiński, ou foram estudados por ele: Triângulo de Sierpiński, Tapete de Sierpiński, Esponja de Menger, Número de Sierpiński, Curva de Sierpiński, Constante de Sierpiński, Espaço de Sierpiński, Teorema da aridade

George Cantor

Georg Ferdinand Ludwig Philipp Cantor (São Petersburgo, 3 de março de 1845 — Halle, 6 de janeiro de 1918) foi um matemático alemão nascido no Império Russo.

Conhecido por ter elaborado a moderna teoria dos conjuntos, foi a partir desta teoria que chegou ao conceito de número transfinito, incluindo as classes numéricas dos cardinais e ordinais e estabelecendo a diferença entre estes dois conceitos, que colocam novos problemas quando se referem a conjuntos infinitos. Nasceu em São Petersburgo (Rússia), filho do comerciante dinamarquês, George Waldemar Cantor, e de uma musicista russa, Maria Anna Böhm. Em 1856 sua família mudou-se para a Alemanha, continuando aí os seus estudos. Estudou no Instituto Federal de Tecnologia de Zurique. Doutorou-se na Universidade de Berlim em 1867. Teve como professores Ernst Kummer, Karl Weierstrass e Leopold Kronecker.

Cantor provou que os conjuntos infinitos não têm todos a mesma potência (potência significando "tamanho"). Fez a distinção entre conjuntos numeráveis (ou enumeráveis) (em inglês chamam-se *countable* - que se podem contar) e conjuntos contínuos (ou não-enumeráveis) (em inglês *uncountable* - que não se podem contar). Provou que o conjunto dos números racionais \mathbb{Q} é (e)numerável, enquanto que o conjunto dos números reais \mathbb{R} é contínuo (logo, maior que o anterior). Na demonstração foi utilizado o célebre argumento da diagonal de Cantor ou método diagonal. Nos últimos anos de vida tentou provar, sem o conseguir, a "hipótese do contínuo", ou seja, que não existem conjuntos de potência intermédia entre os numeráveis e os contínuos - em 1963, Paul Cohen demonstrou a indemonstrabilidade desta hipótese. Em 1897, Cantor descobriu vários paradoxos suscitados pela teoria dos conjuntos.

Giuseppe Peano

Giuseppe Peano (27 de agosto de 1858 – Turim, 20 de abril de 1932) foi um matemático e glottologista italiano. Autor de mais de 200 livros e artigos, foi um dos fundadores da lógica matemática e da teoria dos conjuntos, para as quais ele também contribuiu bastante da notação. A axiomatização padrão dos números naturais é chamada de axiomas de Peano, em sua homenagem. Como parte desse esforço, fez contribuições fundamentais para o tratamento rigoroso e sistemático moderno do método da indução matemática. Passou a maior parte da sua carreira ensinando matemática na Universidade de Turim.

Em 1886, ele começou a ensinar também na Academia Militar Real, e foi promovido a Professor Primeira Classe em 1889. No ano seguinte, a Universidade de Turim concedeu-lhe sua cátedra. A famosa curva que preenche o espaço de Peano apareceu em 1890 como um contraexemplo. Ele a usou para mostrar que uma curva contínua não pode ser sempre incluída em uma região arbitrária pequena. Isso era um exemplo adiantado do que viria a ser conhecido como fractal.

Nikolai I. Lobachewsky

Nikolai Ivanovich Lobachevsky (Níjni Novgorod, 2 de Novembro de 1792–Kazan, 24 de Fevereiro de 1856) foi um matemático russo.

Estudou no Instituto de Kazan, a partir de 1802, destacando-se, desde cedo, por seu pendor pela matemática. Aos 14 anos estava preparado para ingressar na Universidade de Kazan, recentemente fundada, iniciando seus estudos superiores em 1807. Passou quarenta anos de sua vida ligado a essa universidade, como aluno, professor ajudante (1824), professor ordinário (1826) com apenas 23 anos, bibliotecário, conservador do museu e, por fim, reitor(1827), cargo que ocupou até 1846..

Lobachevsky foi considerado por Clifford (1845-1879) como o "Copérnico da geometria", em virtude de suas descobertas relacionadas com as chamadas geometrias não-euclidianas. Nas funções de bibliotecário, pode ser considerado o verdadeiro organizador da Universidade de Kazan - que chegou a salvar de um incêndio, ocorrido em 1842, responsável pela destruição de quase metade da cidade, inclusive de alguns dos grandes edifícios da universidade.

Lobachevsky trabalhou em álgebra, nomeadamente nas aproximações numéricas às raízes das equações algébricas. No entanto, a sua fama provém do facto de ter sido o primeiro matemático (independentemente de János Bolyai) a publicar uma descrição de uma geometria não euclidiana. A geometria não euclidiana que Lobachevsky desenvolveu é referida como a geometria hiperbólica.

- **Atividades de Criação de fractais**

Agora que você adquiriu uma ideia intuitiva e quer se embrenhar pelo mundo dos fractais, estude tópicos sobre conjuntos, distâncias, métricas, espaços métricos, pontos fixos, contração, etc. que vai se maravilhar com as estruturas que irá conhecer e descobrir no fantástico mundo da matemática.

1ª) Fractal Quadrado: Utilize o programa abaixo, chamado `quadrado_fractal`, para gerar diversas figuras, conforme proposto.

<pre> aprenda quadrado_fractal :t :n ;caso base se :n = 0 [pare] ; caso recursivo: desenhe o quadrado tal que em cada canto tem um quadrado fractal repita 4 [pf :t pd 90 quadrado_fractal :t * 0.4 :n-1] fim </pre>
<pre> quadrado_fractal 100 3 quadrado_fractal 200 4 quadrado_fractal 300 6 </pre>

Tabela 4.1

2ª) Construção de uma árvore fractal: Faça outros programas análogos ao programa ao que foi apresentado para gerar uma árvore fractal com alguns testes, por exemplo, substituindo o primeiro comando de deslocamento da tartaruga para a direita de um ângulo de 90° por um ângulo de 30°, 45° ou 60°.

3ª) Faça um programa em Superlogo para gerar as figuras fractais propostas em vestibulares pelas universidades abaixo.

(UFF) Certas imagens captadas por satélites espaciais, quando digitalizadas, são representadas por formas geométricas de aspecto irregular ou fragmentado, conhecidas por fractais. Podem-se obter tais fractais pela alteração da forma original de uma curva por meio de um processo em que os resultados de uma etapa são utilizados como ponto

de partida para a etapa seguinte. Considere o processo tal que, em todas as etapas, cada segmento de reta é transformado em uma poligonal cujo comprimento é quatro vezes a terça parte do segmento original, como ilustrado na figura a seguir:



Figura 4.1

Por esse processo, a partir de um quadrado com 1 metro de lado, obtém-se a sequência de figuras:

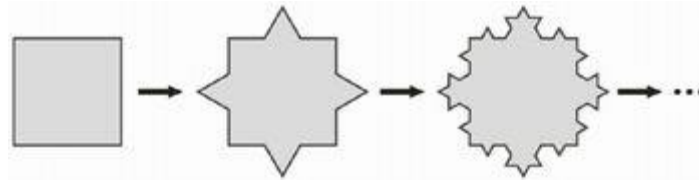


Figura 4.2

(UFRJ) A região fractal F , construída a partir de um quadrado de lado 1cm, é constituída por uma infinidade de quadrados e construída em uma infinidade de etapas. A cada nova etapa consideram-se os quadrados de menor lado (L) acrescentados na etapa anterior e acrescentam-se, para cada um destes, três novos quadrados de lado $L/3$. As três primeiras etapas de construção de F são apresentadas a seguir.

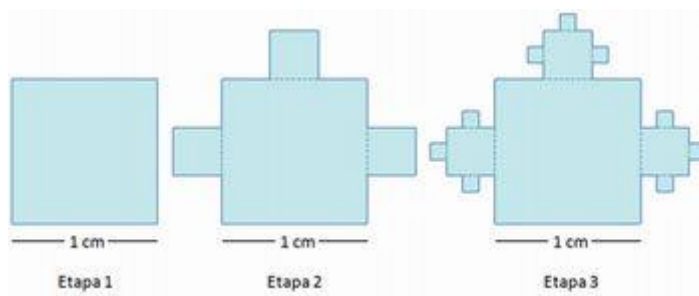


Figura 4.3

4ª) O programa a seguir, chamado de `qcirculo_fractal` gerou a figura abaixo para raio igual a 100 pixels e número de iterações igual a 3. Observamos que o total de círculos gerados incluindo o círculo inicializador foi de 21 círculos. Faça uma tabela e tente descobrir o número de círculos gerados para um número qualquer de iterações e raio fixo.

```

aprenda qcirculo_fractal :r :n
Se :n=0 [pare]
repita 360 [pf 3.1415 * :r / 180 pd 1] repita 4[pd 45 ub pf 1.4142*:r ul pd 45 qcirculo_fractal :r/2
:n-1]
fim

```

Para chamar a procedure, por exemplo, digite, Tat circulo_fractal 100 3

Tabela 4.2

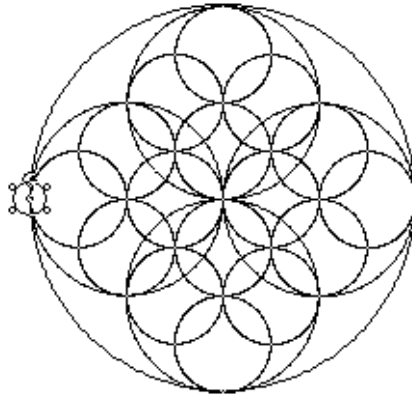


Figura 4.4

5ª) As três figuras abaixo mostram etapas de um fractal gerado por círculos. As duas acima representam o primeiro e terceiro termo de uma sequência cujo limite é o fractal da figura abaixo. Faça um programa em Superlogo que gera o fractal para um número qualquer de iterações e para um raio qualquer do círculo.

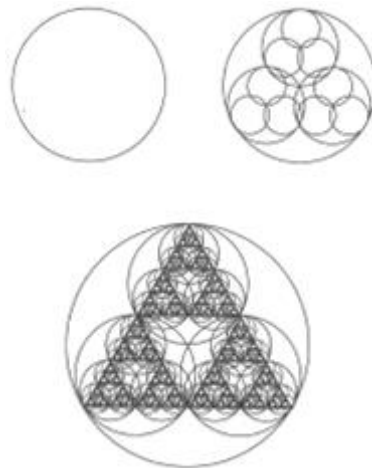


Figura 4.5

Bibliografia

BARBOSA, R. M. *Descobrimdo a geometria fractal para sala de aula*, Belo Horizonte: Autêntica, 2002.

BOYER, B, C. *História da matemática*, Edt. Edgard Blücher Ltd. 1996.

HUNTER, D. J. *Fundamentos da matemática discreta*, Rio de Janeiro: LTC, 2010.

GERSTING, J. L. *Fundamentos matemáticos para a ciência da computação*. quinta ed.

www.educadores.diaadia.pr.gov.br/arquivos/file/2010/artigos_teses/matematica/monografiafractais.pdf. (acessado em: 23/09/2018).

www.profmat.uem.br/dissertacoes-2/rogerio_paixao.pdf. (acessado em: 16/08/2018).