



Universidade Estadual de Campinas

Instituto de Matemática, Estatística
e Computação Científica

DANIEL TEBALDI SANTOS

**O Uso de Algoritmos e Programação no
Ensino de Matemática**

CAMPINAS
2015

DANIEL TEBALDI SANTOS

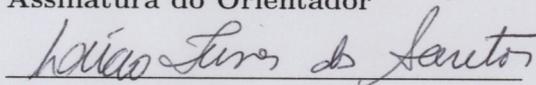
**O Uso de Algoritmos e Programação no
Ensino de Matemática**

Dissertação apresentada ao Instituto de Matemática, Estatística e Computação Científica da Universidade Estadual de Campinas como parte dos requisitos exigidos para a obtenção do título de Mestre.

Orientador: Lúcio Tunes dos Santos

ESTE EXEMPLAR CORRESPONDE À VERSÃO FINAL DA DISSERTAÇÃO/TESE DEFENDIDA PELO ALUNO DANIEL TEBALDI SANTOS, E ORIENTADA PELO PROF. DR. LÚCIO TUNES DOS SANTOS.

Assinatura do Orientador



**Campinas
2015**

Agência(s) de fomento e nº(s) de processo(s): CAPES

Ficha catalográfica
Universidade Estadual de Campinas
Biblioteca do Instituto de Matemática, Estatística e Computação Científica
Ana Regina Machado - CRB 8/5467

Santos, Daniel Tebaldi, 1981-
Sa59u O uso de algoritmos e programação no ensino de matemática / Daniel Tebaldi Santos. – Campinas, SP : [s.n.], 2015.

Orientador: Lúcio Tunes dos Santos.
Dissertação (mestrado profissional) – Universidade Estadual de Campinas, Instituto de Matemática, Estatística e Computação Científica.

1. Algoritmos. 2. Matemática (Ensino médio) - Estudo e ensino. I. Santos, Lúcio Tunes dos, 1962-. II. Universidade Estadual de Campinas. Instituto de Matemática, Estatística e Computação Científica. III. Título.

Informações para Biblioteca Digital

Título em outro idioma: Using algorithms and programming in teaching mathematics

Palavras-chave em inglês:

Algorithms

Mathematics (High school) - Study and teaching

Área de concentração: Matemática em Rede Nacional

Titulação: Mestre

Banca examinadora:

Lúcio Tunes dos Santos [Orientador]

Sandra Augusta Santos

Marina Andretta

Data de defesa: 10-12-2015

Programa de Pós-Graduação: Matemática em Rede Nacional

Dissertação de Mestrado Profissional defendida em 10 de dezembro de 2015 e aprovada pela Banca Examinadora composta pelos Profs. Drs.

Prof(a). Dr(a). LÚCIO TUNES DOS SANTOS

Prof(a). Dr(a). SANDRA AUGUSTA SANTOS

Prof(a). Dr(a). MARINA ANDRETTA

A Ata da defesa com as respectivas assinaturas dos membros encontra-se no processo de vida acadêmica do aluno.

A Deus por me dar a sabedoria. A minha esposa a quem eu amo. Aos meus pais. E a todos que me ajudaram nessa caminhada . . .

Agradecimentos

Em primeiro lugar agradeço a Deus, por me dar força e sabedoria para alcançar essa etapa e vencer mais esta batalha, superando as dificuldades e adversidades encontradas ao longo deste percurso. Agradeço a minha esposa pelos seus incentivos, para que eu continuasse em busca de conhecimentos e pela compreensão nos momentos em que precisei me dedicar a este trabalho. Agradeço a minha mãe e meu pai pela simplicidade com que me fizeram entender os valores das coisas mais importantes na vida. Agradeço aos meus amigos e colegas de graduação e mestrado, que foram importantes em momentos de dúvidas e compartilhamento de conhecimentos. Agradeço ao meu orientador Lúcio Tunes Santos por aceitar orientar este trabalho, acreditar nos seus objetivos e também pelas sugestões e ideias que foram muito importantes para esta pesquisa. Por fim agradeço ao Profmat, à Unicamp e à Capes por todo apoio e auxílio financeiro.

“Sempre faço o que não consigo
fazer para aprender
o que não sei”

Pablo Picasso

Resumo

Nesta pesquisa, foi apresentada uma proposta de trabalho para o desenvolvimento do conhecimento matemático, com alunos do ensino médio, relacionando alguns conteúdos matemático do ensino básico com a construção de algoritmos, com a intenção de proporcionar uma compreensão melhor desses conteúdos. Além disso, as atividades de construção de algoritmos foram desenvolvidas com o auxílio do programa de visualização de algoritmos VisuAlg.

Palavras-chave: algoritmos, matemática, ensino médio.

Abstract

In this research, a work proposal for the development of mathematical knowledge was presented, with high school students by relating some mathematical content of basic education with the construction of algorithms, with the intention of providing a better understanding of such content. In addition activities were developed for the construction of the algorithms with the help of the viewer program VisuAlg.

Keywords: algorithms, mathematics, high school.

Sumário

1	INTRODUÇÃO	12
2	ALGORITMOS HOJE	15
1	A importância dos algoritmos	15
2	O ensino de algoritmos	17
3	ALGORITMO	19
1	Regras Importantes	19
2	Conceitos Importantes	21
2.1	Pseudocódigo (Português Estruturado)	21
2.2	Constantes, variáveis e tipos de dados	23
2.3	Operadores lógicos	24
2.4	Operações lógicas	24
2.5	Comandos de repetição	25
4	VisuAlg	27
1	Escolha do programa de validação do algoritmo	27
1.1	Comandos do VisuAlg	29
5	DESENVOLVIMENTO DA PESQUISA	32
1	Escola	32
2	Materiais para a pesquisa	33
3	Dados e algoritmos construídos	33
3.1	Atividade 1	34
3.2	Atividade 2	35

3.3	Atividade 3	37
3.4	Atividade 4	38
3.5	Atividade 5	40
3.6	Atividade 6	42
3.7	Atividade 7	43
3.8	Atividade 8	45
3.9	Atividade 9	49
3.10	Atividade 10	51
6	CONSIDERAÇÕES FINAIS	55
	BIBLIOGRAFIA	55
	APÊNDICE	57
A	PROPOSTA DE ATIVIDADES	58
1	MDC e MMC	59
2	Sequência de Fibonacci e número áureo	62
3	Raiz quadrada	64
4	Calculando o valor de π	67

Capítulo 1

INTRODUÇÃO

Há cinco anos como professor de matemática da rede pública de ensino nos níveis fundamental e médio, venho percebendo uma constante piora nos níveis de ensino da matemática. Acredito que isto se deve ao tipo de metodologia aplicada à construção do conhecimento matemático em sala de aula. Contudo, ao longo das disciplinas do mestrado, conversando com colegas, professores e pensando em um possível tema para a minha dissertação, foi nas aulas de Cálculo Numérico que percebi que podia trabalhar em uma alternativa para melhorar a construção do conhecimento matemático dos alunos.

Encontrar uma forma de fazer o aluno organizar seu pensamento matemático para que ele possa analisar seu raciocínio lógico matemático na resolução das situações problemas foi uma das principais motivações deste trabalho. Isso, atrelado a uma teoria que se torna cada vez mais presente na vida das pessoas e de uma importância enorme no avanço da tecnologia computacional, que é a teoria de construção de algoritmos.

Apresentar essa teoria aos alunos da rede pública de ensino e com isso fazê-los usar essa teoria na resolução de situações-problema, demanda um trabalho nada trivial, até porque os alunos estão inseridos numa tecnologia desconstrutiva do ponto de vista do conhecimento científico. A maioria destes alunos não consegue relacionar essa tecnologia com os conhecimentos adquiridos em aula e também utilizá-la de forma a ajudá-los na construção do seu conhecimento escolar, assim tornando esse trabalho mais motivador e

desafiador [8].

Pensar nas enormes dificuldades para se obter sucessos na construção do conhecimento matemático com alunos do ensino fundamental e médio da rede pública de ensino é uma das grandes demandas da educação nos dias atuais. Esta pesquisa de mestrado está mobilizada neste sentido, direcionada a tentar unir o algoritmo computacional com a programação, conhecimentos fundamentais no desenvolvimento de tecnologias computacionais e que estão incorporadas de uma forma definitiva na vida de todos, com o conteúdo matemático ensinado nas escolas da rede pública de ensino, como uma forma de produzir uma nova estratégia para o ensino da matemática por meios não tradicionais. Também buscamos introduzir a teoria de construção e validação de algoritmos na vida escolar do aluno.

A teoria de construção dos algoritmos pode ser inserida como estratégia de ensino-aprendizagem, nos mais diversos segmentos do conhecimento escolar, aprimorando técnicas construtivas do desenvolvimento do aprendizado [10].

A escola nos níveis fundamental e médio enfrenta problemas crônicos no ensino da matemática. As diversas mudanças na sociedade proporcionaram uma geração de alunos muito diferentes devido à influência dos meios tecnológicos computacionais. Superar estes problemas exige um trabalho de alinhamento entre o conhecimento tradicional e os meios tecnológicos, com propostas que possam ir ao encontro dos anseios, expectativas, interesses e necessidades futuras dos alunos.

O conhecimento matemático, ensinado de uma maneira tradicional ou não, precisa estar voltado à formação intelectual e da cidadania dos indivíduos. Deve, também capacitá-los como cidadãos críticos, instrumentalizando-os para a reflexão e a interação com o meio em que vivem, utilizando das ferramentas matemáticas que são disponibilizadas pelos currículos escolares. No entanto, ainda existe um abismo nesta relação, que dificulta transcender as barreiras da conexão das ferramentas matemáticas com o meio em que o aluno está inserido [10], de forma que a escola não possibilita ao aluno uma visão que faça com que ele avance na autonomia da construção do seu conhecimento.

Diminuir este abismo e transpor estas barreiras não está sendo uma tarefa

fácil para os professores, em especial os de Matemática. Com um currículo ainda distante da realidade do aluno, isto faz com que o professor não consiga proporcionar ao aluno situações com que ele possa fazer relações com o meio que está inserido, dando à aula uma dinâmica pouco interessante.

Portanto, levando em consideração todo argumento exposto e com a intenção de mapear as dificuldades encontradas na construção do conhecimento matemático no ensino fundamental e médio na rede pública, esta dissertação pretende elaborar uma proposta de trabalho que apresente ao aluno a teoria de construção de algoritmos computacional como uma prática que o auxilie na organização e resolução de situações-problema que demandem uma estratégia. Também visa a associação entre conceitos matemáticos e o meio tecnológico computacional, de forma que ele perceba os erros de construção da sua estratégia por meio da implementação do algoritmo construído em um programa de validação de algoritmos.

A pesquisa tem por finalidade motivar a inserção da teoria de construção de algoritmos computacionais no currículo do ensino de matemática, como estratégia de construção de conhecimento, de resolução de situações-problema. Também se propõe a apresentar um programa de validação de algoritmos para analisar o algoritmo construído, de forma a certificar se a lógica e o raciocínio aplicados na construção do algoritmo são perfeitamente entendidos por uma máquina e com isso mostrar ao aluno a importância destas técnicas para o desenvolvimento das tecnologias computacionais e que pode contribuir e tornar mais fácil o entendimento e a resolução de uma situação problema na matemática [7].

Capítulo 2

ALGORITMOS HOJE

1 A importância dos algoritmos

Para Berlinski [4], duas grandes ideias científicas brilham no mundo, a primeira grande ideia é o cálculo e a segunda grande ideia é o algoritmo, e define essa segunda grande ideia como a “A ideia que governa o mundo”. Algoritmos estão inseridos no nosso cotidiano de diversas maneiras e estão por toda a parte. Essa grande ideia científica proporcionou muitos avanços nas tecnologias computacionais existentes.

Na sua essência, um algoritmo é um método passo a passo para fazer uma determinada tarefa. Esta forma diferenciada de pensar é uma estratégia muito importante para muitas áreas do conhecimento, pois um computador com um algoritmo eficiente, pode executar uma tarefa em uma velocidade muito grande, usando um volume muito grande de dados. Entender como funcionam essas tecnologias presentes no nosso cotidiano é fundamental [13].

Um software, por exemplo, é essencialmente um algoritmo que dita ao computador os passos específicos e a ordem em que devem ser executados. O que mudou desde os tempos de Al-Khowârizmi¹? É o que os algoritmos estão fazendo agora: tudo [11]. Acumulam mais informação do que qualquer ser humano seria capaz e estabelecem relações que nenhum de nós vislumbraria.

Quase tudo que fazemos hoje é definido por um algoritmo, o que lemos,

¹Falaremos sobre Al - Khowârizmi no capítulo seguinte.

assistimos, pesquisamos e outras mais. Os algoritmos ajudam a organizar, manipular e pensar no mundo digital e computacional, criando estruturas simbólicas ou funcionais que formarão estes sistemas.

Com o crescimento rápido do acesso a informações digitais, tornam-se necessário desenvolver algoritmos muito eficientes para a organização dessas informações. A fusão entre um algoritmo eficiente e um computador potente pode produzir uma ferramenta muito poderosa para muitas áreas, como mercado financeiro, medicina, logística e até mesmo jornalismo. Atualmente muitas empresas estão investindo no desenvolvimento de algoritmos que minimizem seus custos proporcionando um aumento considerável em seus lucros.

Um bom exemplo do uso de algoritmos é no setor logístico, de acordo com a empresa norte-americana de logística UPS (United Parcel Service Inc.), que usa algoritmos para ajudar a entregar os milhões de encomendas que passam através de sua rede de transporte todos os dias da maneira mais eficiente possível. As rotas mais simples são fáceis de elaborar. Se um motorista tem apenas três destinos para visitar, ele pode levar apenas seis rotas possíveis. Mas o número de rotas possíveis explode quando os destinos aumentam. “Algoritmos fornecem benefícios quando as escolhas são tão grandes que eles são impossíveis de se processar em sua cabeça”, diz Jack Levis da UPS [1].

Um outro exemplo, pode ser encontrado no mercado financeiro, onde existem muitos algoritmos que ajudam investidores a tomarem decisões muito rápidas. As empresas que atuam no mercado financeiro contratam profissionais especialistas em desenvolvimento de algoritmos e programação, para criar programas e não divulgam os algoritmos que estão por trás destes programas, que são chamados de algoritmos secretos.

Os softwares para o mercado financeiro, programados a partir destes algoritmos, são usados principalmente para obter ganhos em volume a partir de transações pequenas. A função do programa é prever alterações imediatas no preço das ações e realizar operações de compra e venda em alta velocidade. Enquanto os corretores humanos mais dinâmicos costumam comprar meia dúzia de papéis pela manhã e vender outro tanto à tarde, os programas de computador conseguem operar em escala sobre-humana, negociando cen-

tenas de ações em alguns segundos e obtendo, como lucro de 1 ou 2 centavos em cada uma [13].

“Os algoritmos tentam identificar padrões que se repetem no mercado financeiro e, com algumas funções, identificar as altas e baixas”, explica Carlos Martins autor do Trader Gráfico, um software de informações financeiras para investidores [1].

“Geralmente, o que o pessoal de mercado gosta de fazer é programar operações de compra e venda em alguns segundos. Se a máquina for programada para esperar muito, informações de fora (falências, fusões, notícias sobre a economia do país) vão criar um sinal falso e a pessoa toma um prejuízo enorme, ressalta”. “Todo o dia o algoritmo precisa ser atualizado, para acompanhar o comportamento do mercado. Se deixar parado, uma programação que dá lucro em um mês vai causar prejuízo no mês seguinte”, lembra. “O algoritmo não faz nada diferente do que o mercado sempre fez, só que em maior quantidade.”, defende Martins [1].

Todos os exemplos citados evidenciam a importância dos algoritmos nos avanços da sociedade e também mostram a relevância de trabalhar esse conhecimento para contribuir na construção de saberes no ensino básico. Além de tudo, os algoritmos têm a propriedade de promover o desenvolvimento do raciocínio lógico e auxiliar na nossa capacidade de resolver problemas.

2 O ensino de algoritmos

Um país que foi pioneiro em incluir o ensino de programação no currículo do ensino básico, foi Israel. No fim da década de 1990 Israel [6] já revolucionava o seu currículo da escola básica, incluindo nele o ensino de algoritmos e programação.

Para época, podemos dizer que Israel estava arriscando em ensinar uma ciência, que para muitos países ainda não era muito conhecida, tampouco, havia muitos estudiosos desta ciência, que começou a ser mais explorada no final da Segunda Guerra Mundial, com as contribuições de Alan Turing, no desenvolvimento da Máquina de Turing, que foi a precursora dos computadores.

Já no século XXI, podemos afirmar que Israel já estava visualizando o futuro das tecnologias digitais, que hoje são de fundamental importância para a sociedade. Mais de uma década depois, muitos países estão seguindo os passos de Israel, como Nova Zelândia, Dinamarca, Austrália, alguns estados alemães e recentemente a Inglaterra [2].

Em setembro de 2014, o ensino de ciência da computação foi integrado ao currículo do ensino básico da Inglaterra [5], como disciplina obrigatória, portanto crianças a partir dos cinco anos agora estão aprendendo a construir algoritmos e conseqüentemente nos anos seguintes irão aprender a programar.

Esta decisão foi tomada depois que autoridades inglesas perceberam que há uma escassez de mão de obra nesta área e também após receber conselhos de duas empresas que são potências mundiais nesta área, a Google e a Microsoft, indicando que o currículo das escolas públicas de ensino básico estava fora de sintonia com os padrões técnicos modernos.

Além da questão econômica do país, que foi o ponto destacado pelas autoridades, a equipe que desenvolveu o novo currículo do ensino básico inglês destaca, “a educação de computação de alta qualidade equipa os alunos a usar o pensamento computacional e criatividade para compreender e mudar o mundo. Computar possui laços profundos com a matemática, ciência e design e tecnologia, e fornece insights sobre ambos os sistemas naturais e artificiais” [2].

Cabe salientar que o objetivo desta pesquisa de mestrado é mostrar como o uso de algoritmos no ensino de matemática pode promover uma compreensão maior dos conteúdos, o que está em consonância com as perspectivas do novo currículo do ensino básico na Inglaterra.

Capítulo 3

ALGORITMO

1 Regras Importantes

Mas afinal, o que é um algoritmo e por que se fala cada vez mais deles?

Na matemática, é todo conjunto de regras e operações para fazer cálculos, realizar tarefas ou solucionar problemas [3].

De acordo com Knuth [9] a palavra é derivada do nome do matemático persa do século IX, Muhammad iben Muça Al-Khowârizmi, que aparece por deturpação do seu nome. Suas obras foram traduzidas para o mundo ocidental no século XII sob o título “*Algorithmi de número indorum*”, um livro de aritmética onde são descritos os cálculos na numeração decimal. Desde então a palavra algoritmo passou a ser utilizada para descrever os planos de resolução de algum problema. Para Berlinsk [4],

“O computador digital é uma máquina e, como qualquer objeto, um prisioneiro ao fim de áridas leis da termodinâmica. Quando o tempo acaba, o combustível acaba também. Da mesma forma que o programador de computador martelando em um teclado com a ponta de dois dedos tensos. Como todos nós. Mas um algoritmo é outra coisa. Ocupando o espaço entre a agulhada do desejo e a resultante bolha de satisfação, é um instrumento abstrato de coordenação, que fornece os procedimentos para várias finalidades.”

Feito de sinais e símbolos, os algoritmos, como os pensamentos, vivem em um mundo além do tempo.

Um algoritmo é um procedimento eficaz, um modo de fazer uma coisa em um número finito de passos discretos.

No mundo de onde surge um matemático e para o qual o matemático, como nós, deve voltar, um algoritmo, por assim dizer, é um conjunto de regras, uma receita, uma prescrição para a ação, um guia, uma diretiva concatenada e controlada, uma intimidação, um código, um esforço feito para jogar um complexo xale verbal sobre o caos inarticulado da vida.”

A definição usada de forma moderna, não só leva em questão os procedimentos, rotinas ou métodos bem definidos, para a resolução do problema, mas também, a presença de cinco características indispensáveis às aplicações diversificadas dos algoritmos atualmente, e são elas:

Finitude:

Um algoritmo deve sempre terminar após um número finito de passos.

Assertividade:

Cada passo de um algoritmo deve ser precisamente definido. Em outras palavras, as ações devem ser definidas rigorosamente e de forma nunca ambígua em cada caso em que apliquem.

Entrada:

Um algoritmo deve ter zero ou mais entradas, isto é, quantidades que lhe são fornecidas antes do algoritmo iniciar ou de forma dinâmica.

Saídas:

Um algoritmo deve ter uma ou mais saídas, na forma de um conjunto de valores entregues por ele e que estão associados ao conjunto de relação específica com as entradas.

Efetividade:

Um algoritmo deve ser efetivo. Isto significa que todas as operações devem ser suficientemente simples de modo que possam ser em princípio executadas manualmente, com precisão em um tempo finito.

2 Conceitos Importantes

2.1 Pseudocódigo (Português Estruturado)

Este nome é uma alusão à posterior implementação em uma linguagem de programação, ou seja, quando formos programar em uma linguagem, por exemplo VisuAlg¹, estaremos gerando código em VisuAlg. Por isso os algoritmos são independentes das linguagens de programação. Ao contrário de uma linguagem de programação, não existe um formalismo rígido de como deve ser escrito o algoritmo.

O algoritmo deve ser fácil de interpretar e de codificar, ou seja, ele deve ser o intermediário entre a linguagem falada e a linguagem de programação.

Para escrever um algoritmo precisamos descrever a sequência de instruções, de maneira simples e objetiva. Para isso utilizaremos algumas técnicas:

- Usar somente um comando por linha;
- Imaginar que você está desenvolvendo um algoritmo para pessoas que não trabalham com informática;
- Usar frases curtas e simples;
- Ser objetivo;
- Usar palavras que não tenham sentido duplo;
- O algoritmo deve funcionar;
- O algoritmo não deve realizar tarefas inúteis. Ele deve ser feito da forma mais simples possível;
- O algoritmo deve ser claro, isto é, fácil de ler e entender;

Cada operação descrita em um algoritmo tem que ser bem definida e perfeitamente clara. Também cada operação deve ser efetiva e o algoritmo, pelo

¹Programa de visualização de algoritmos, que falaremos no capítulo seguinte.

menos a princípio, permitir a uma pessoa munida apenas de papel e lápis executá-lo em um tempo finito. Entretanto, ao montar um algoritmo, precisamos primeiro dividir o problema apresentado em três fases fundamentais, que são [14]:

Entrada:

São os dados iniciais para começar o algoritmo.

Processamento:

São os procedimentos utilizados nos dados iniciais para chegar ao resultado final.

Saída:

São os dados que serão exibidos no final do processamento.

Imagine o seguinte problema:

Calcular a média final de um aluno do 1^a ano do Ensino Médio. O aluno realizou quatro provas, que terão notas n_1, n_2, n_3 e n_4 , onde:

$$\text{Média Final} = \frac{n_1+n_2+n_3+n_4}{4}$$

Para montar o algoritmo proposto, teremos três partes:

- Os dados de **entrada** são n_1, n_2, n_3 e n_4 .
- O **procedimento** será somar os 4 dados de entrada (n_1, n_2, n_3, n_4) e dividir o seu resultado por 4.
- O dado de **saída** será a média final processada.

Algoritmo: *Média final de um aluno do 1^a ano do Ensino Médio*

Início do algoritmo

Entrada n_1

Entrada n_2

Entrada n_3

Entrada n_4

Média = $(n_1 + n_2 + n_3 + n_4)/4$

Saída Média

Final do algoritmo

2.2 Constantes, variáveis e tipos de dados

O objetivo de um algoritmo é manipular dados e chegar à solução de um problema. Para manipular dados em uma linguagem de programação qualquer que seja, utiliza-se variáveis e constantes [14].

Uma **variável** é um espaço reservado na memória do computador para armazenar um tipo de dado determinado. Variáveis devem receber nomes para poderem ser referenciadas e modificadas quando necessário. Um programa deve conter declarações que especificam de que tipo são as variáveis que ele utilizará e às vezes um valor inicial.

Tipos podem ser por exemplo: inteiros, reais, caracteres e etc. As expressões combinam variáveis e constantes para calcular novos valores.

Uma **constante** é um determinado valor fixo que não se modifica ao longo do tempo, durante a execução de um programa. Conforme o seu tipo, a constante é classificada como sendo numérica, lógica ou literal.

As variáveis e as constantes podem ser basicamente de quatro tipos:

- As **numéricas** são específicas para armazenamento de números, que posteriormente poderão ser utilizados para cálculos. Podem ser ainda classificadas como inteiras ou reais. As variáveis do tipo inteiro são para armazenamento de números inteiros e as reais são para o armazenamento de números que possuam casas decimais.
- Os **caracteres** específicos para armazenamento de conjunto de caracteres que não contenham números (literais). Ex: nomes.
- As **alfanuméricas** específicas para dados que contenham letras e/ou números. Pode em determinados momentos conter somente dados numéricos ou somente literais. Se usado somente para armazenamento de números, não poderá ser utilizada para operações matemáticas.
- As **lógicas** armazenam somente dados lógicos que podem ser verdadeiro ou falso.

As variáveis só podem armazenar valores de um mesmo tipo, de maneira que também são classificadas como sendo numéricas, lógicas ou literais.

2.3 Operadores lógicos

As expressões podem conter operadores lógicos de três tipos, caracterizando o seu valor lógico, que podem ser:

- Uma expressão com o operador **e** é verdadeira, se todas as condições forem verdadeiras.
- Uma expressão com o operador **ou** é verdadeira, se pelo menos uma das condições for verdadeira.
- Uma expressão com o operador **não**, inverte o valor da expressão ou condição, se verdadeira inverte para falsa e vice-versa.

2.4 Operações lógicas

Operações lógicas são utilizadas quando se torna necessário tomar decisões em um algoritmo. Num algoritmo, toda decisão terá sempre como resposta o resultado verdadeiro ou falso.

Os comandos de decisão ou desvio fazem parte das técnicas de construção de algoritmos que conduzem a estruturas de algoritmos que não são totalmente sequenciais. Com as instruções de salto ou desvio pode-se fazer com que o algoritmo proceda de uma ou outra maneira, de acordo com as decisões lógicas tomadas em função dos dados ou resultados anteriores.

As principais estruturas de decisão são:

- A estrutura de decisão **se/então**, vem sempre acompanhada de uma ou mais condições, ou seja, se determinada ou determinadas condições forem satisfeitas no comando **se** então execute determinado comando.
- A estrutura de decisão **se/então/senão**, funciona exatamente como a estrutura **se**, com apenas uma diferença, em **se** somente podemos executar comandos caso a ou as condições sejam verdadeiras, diferentemente da estrutura **se/então/senão**, pois sempre um comando será executado independente da ou das condições, ou seja, caso a ou as

condições sejam verdadeiras, o comando da ou das condições serão executados, caso contrário o comando da ou das condições falsas serão executados.

No exemplo abaixo, foi construído um algoritmo para determinar dentre três números qual é o maior. Podemos também dentro de uma mesma condição testar outras condições.

Algoritmo: *Decidir dentre três números qual é o maior.*

Início do algoritmo

Entrada n_1

Entrada n_2

Entrada n_3

Se $(n_1 > n_2)$ e $(n_1 > n_3)$ Então

Saída n_1

Senão

Se $(n_2 > n_3)$ Então

Saída n_2

Senão

Saída n_3

Final do algoritmo

2.5 Comandos de repetição

Você deverá utilizar os comandos de repetição quando desejar que um determinado conjunto de instruções ou comandos sejam executados um número definido ou indefinido de vezes, ou enquanto um determinado estado de coisas prevalecer ou até que seja alcançado. Veja abaixo os comandos de repetição que são utilizados:

- A estrutura **para/até/faça**, repete uma sequência de comandos um determinado número de vezes até chegar em uma valor limite.

- A estrutura **enquanto/ faça**, repete uma sequência de comandos enquanto uma determinada condição, especificada através de uma expressão lógica, for satisfeita.
- A estrutura **repita/ até** repete uma sequência de comandos até que uma determinada condição, especificada através de uma expressão lógica, seja satisfeita.

Nestes casos, o bloco de operações será executado enquanto a condição x for verdadeira. O teste da condição será sempre realizado antes de qualquer operação. Enquanto a condição for verdadeira o processo se repete. Podemos utilizar essa estrutura para trabalharmos com contadores.

No próximo exemplo, foi construído um algoritmo para imprimir os números ímpares entre 1 e 20, onde foi usada a estrutura de comando de repetição **enquanto/ faça**:

Algoritmo: *Números ímpares entre 1 e 20*

Início do algoritmo

$x = 1$

Enquanto $x \leq 20$ Faça

Saída x

$x \leftarrow x + 2$

Fim do enquanto

Final do algoritmo

Observação: A seta (\leftarrow) indica que o valor de x vai ser trocado por outro, que será o próprio x adicionado de 2 a cada iteração do algoritmo.

Capítulo 4

VisuAlg

1 Escolha do programa de validação do algoritmo

O processo de construir um algoritmo computacional não termina só na construção, é preciso também testá-lo para que se possa tornar esse algoritmo válido, ou seja, implementá-lo em um programa de validação de algoritmos. “Rodar o chinês” é a expressão usada nas ciências computacionais para verificarmos se a sequência lógica do algoritmo é perfeitamente entendida por uma máquina.

Existem muitos programas de validação de algoritmos, por isso precisávamos escolher um que combinasse com a ideia do trabalho e que tivesse as seguintes características:

- Voltado para alunos que tenha pouco ou nenhum contato com linguagens de computadores;
- Preferencialmente no idioma brasileiro;
- Que tenha uma estrutura de construção muito parecida com a construída fora de um programa, com o mínimo de termos de uma linguagem computacional como a Linguagem C;
- Que seja gratuito.

De acordo com as características apresentadas acima a linguagem escolhida foi o VisuAlg. O VisuAlg é um programa de domínio público de execução de algoritmos, e os executa na forma de cálculos e pode ser baixado pelo endereço:

<http://www.guanabara.info/logica/Ferramentas/visualgv25.exe>

Criado pelo professor Cláudio Morgado de Souza, ele conta que a ideia para este programa nasceu da necessidade de uma ferramenta para os alunos iniciantes em programação exercitarem seus conhecimentos num ambiente próximo da realidade de uma linguagem de programação [14].

O programa trabalha com uma linguagem parecida com o Português, que é uma adaptação da língua portuguesa com algumas modificações, transformando-as em comandos e de grande popularidade nos meios acadêmicos e presente nos livros mais utilizados para a introdução no estudo de algoritmos.

Este programa de execução de algoritmos tem os princípios básicos da programação, que geralmente é usado para iniciar alunos em programação de computadores. Além disso, esta ferramenta deve também ser capaz de simular o que acontece na tela do computador com o uso dos famosos comandos “leia” e “escreva”, bem como possibilitar a verificação dos valores das variáveis, o acompanhamento passo a passo da execução de um algoritmo (pelo seu grande valor didático), e até mesmo suportar um modo simples de depuração. Aliado a tudo isto, existe um editor de texto com recursos razoáveis (tais como abrir e salvar arquivos) e que dispõe de todos os principais recursos de um ambiente gráfico [14].

Na figura abaixo é mostrada como é a interface do VisuAlg:

No trecho entre *algoritmo* e *var* (variável) é o local reservado para informações, como: título do algoritmo, função do algoritmo, autor do algoritmo e data de implementação do algoritmo. Entre *var* e *início* é o local onde devem ser feitas as declarações das variáveis que serão usadas no algoritmo. E entre *início* e *fimalgoritmo*, é onde se deve escrever o algoritmo.

Nas janelas debaixo são onde aparecem o algoritmo sendo executado, na da esquerda aparecem as variáveis e seus valores atribuídos a cada passo do

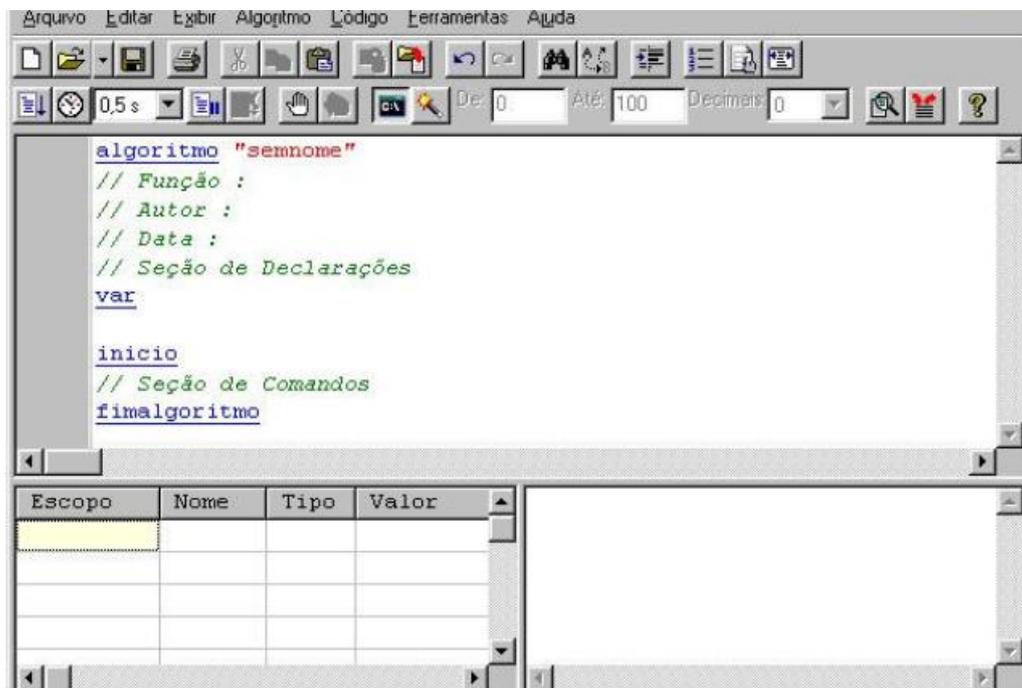


Figura 4.1: Interface VisuAlg

algoritmo e na da direita aparece o algoritmo já executado.

A partir de agora, todos os algoritmos apresentados nas atividades realizadas pelos alunos estarão escritos na forma da linguagem do programa VisuAlg.

1.1 Comandos do VisuAlg

Alguns comandos serão usados frequentemente na construção dos algoritmos usando o VisuAlg. Faremos agora uma apresentação desses comandos e suas estruturas.

- **Tipos de Variáveis**

Ao final das declarações das variáveis, digitamos dois pontos e especificamos as variáveis de acordo com o tipo, que podem ser, *inteiro*, *real*, *caracter* ou *logica*.

- **Imprimir na Tela**

Para que o computador imprima algo na tela, temos que usar o comando *escreva*, que escreve na mesma linha, ou *escreval* que escreve na linha seguinte. Além disso precisamos colocar tudo que será escrito entre parenteses e entre aspas duplas.

- **Comando *leia***

Serve para introduzir um dado externo para dentro do algoritmo. Esse dado externo é guardado em das variáveis já declaradas e que ficarão armazenadas na memória, para que a qualquer momento possa-se usá-las.

- **Comandos de decisão**

Escrevemos o comando *se*, logo em seguida as condições, depois o comando *entao*, em seguida a ordem a ser executada, depois o comando *senao* para executar a ordem quando as condições dadas são falsas e no final, para sairmos desta estrutura, colocamos *fimse*.

- **Comandos de repetição**

Temos a estrutura começando com o comando *para* seguido da variável, o comando *de* com um valor de início, em seguida o comando *ate* com um valor final e depois o comando *faca* seguido da ordem a ser executada e no final o comando *fimpara* para sair da estrutura.

Outra estrutura usa o comando *enquanto* seguido das condições, o comando *faca* seguido da ordem a ser executada e no final o comando *fimenquanto* para sair da estrutura.

E temos também a estrutura com o comando *repita* seguido da ordem a ser executada, logo após o comando *ate* seguido da condição de parada e no final o comando *fimrepita*.

O comando *//*, chamado de comentário, no visuAlg serve para explicarmos alguma situação em alguma linha do algoritmo, para que outras pessoas entendam o que foi feito nesta linha, pois tudo que foi escrito depois de *//* não será lido pelo programa.

- **Operadores matemáticos**

- *abs*(x): Retorna o valor absoluto do número x , $|x|$;
- *exp*(a,n): Encontra a potência de base a e expoente n , a^n ;
- *quad*(x): Retorna o valor de x ao quadrado, x^2 ;
- $x \bmod y$: Encontra o resto da divisão de x por y ;
- *raizq*(x): Calcula a raiz quadrada do número x , \sqrt{x} ;
- *pi*: retorna o valor de π .

Capítulo 5

DESENVOLVIMENTO DA PESQUISA

1 Escola

O desenvolvimento da pesquisa aconteceu em uma escola pública da rede estadual, localizada na cidade de Pedreira no estado de São Paulo. Esta escola foi escolhida por já existir uma aproximação entre nós, pois já tinha trabalhado nela como professor e também pela relação muito boa que construí com os seus gestores.

Foram selecionados 12 alunos do 1º ano do Ensino Médio, pois o objetivo deste trabalho é mostrar a viabilidade do ensino de matemática com o uso de algoritmos e programação primeiramente no Ensino Médio. A seleção dos alunos para participarem da pesquisa foi feita tendo em vista o interesse pela aquisição de novos conhecimentos. Além disso, os alunos convidados representam um grupo que possui um interesse especial pela Matemática.

Com a mudança de ciclo destes alunos, deixando o ensino fundamental e ingressando no ensino médio, percebe-se pouca experiência e uma defasagem destes alunos no desenvolvimento dos conteúdos de Matemática. Porém, podemos observar ainda uma grande expectativa destes para com aprendizado no ensino médio e também quanto à variação de estratégias utilizadas pelos mesmos no desenvolvimento das soluções de problemas.

De acordo com estas características descritas nessa transição de nível de ensino, podemos observar a assimilação dos conteúdos por parte dos alunos na série inicial do ensino médio, por vários ângulos de atuação, podendo assim pensar a viabilidade do ensino de algoritmos em qualquer momento deste segmento.

2 Materiais para a pesquisa

O desenvolvimento desta pesquisa depende da organização de atividades que envolvam a construção de algoritmos em situações que contemplem os conteúdos de matemática e a implementação dos mesmos em um programa de validação (VisuAlg). Para as atividades de construção de algoritmos foram necessários a acomodação dos alunos em uma sala de aula com computador, projetor e lousa, bem como cadernos, lápis e caneta.

Para a implementação do algoritmo construído, foram necessários uma sala de informática com computadores suficientes para todos os alunos e nos computadores o programa para a validação do algoritmo, VisuAlg.

3 Dados e algoritmos construídos

Com o objetivo de obter as informações necessárias para o desenvolvimento dessa pesquisa, deu-se a divisão da aplicação das atividades em três momentos.

No primeiro momento foi explicado aos alunos o objetivo da pesquisa, sem lhes informar os objetivos específicos e sim, conscientizando-os de que se tratava de componentes de uma pesquisa concreta, que visa investigar alguns parâmetros relacionados à estratégia do ensino de conhecimentos matemáticos.

A princípio foi apresentada a definição de algoritmo para situá-los dentro da concepção de sua construção, mostrar a sua importância no desenvolvimento de tecnologias e na organização e estruturação do conhecimento lógico matemático, assim como para Berlinski [4],

“...o algoritmo é a ideia que rege o mundo. Sem sua descoberta, a vida moderna seria muito diferente, já que o computador, a Internet, a realidade virtual e o correio eletrônico simplesmente não existiriam.”

Ainda para Berlinski,

“Um algoritmo é um procedimento eficaz, um modo de fazer uma coisa em um número finito de passos discretos. A Matemática clássica é, em parte, o estudo de determinados algoritmos.”

Após apresentar-lhes uma ideia sobre algoritmo, alguns alunos se mostraram surpresos com tamanha importância do tema nos dias atuais, fazendo alguns questionamentos do tipo: “Onde podemos encontrar os algoritmos no nosso dia-a-dia? Que tipo de situações os algoritmos estão presentes?” Depois foram apresentados aos alunos alguns conceitos e regras importantes para construção de algoritmos.

A partir da compreensão da definição e também do entendimento do que é um algoritmo e depois de trabalhados suas regras e conceitos de construção, foi proposta aos alunos a formulação de um algoritmo que serviu de inicialização para trabalhar com o programa VisuAlg, para entender alguns de seus comandos que serão muito usados, que são os comandos *escreva* ou *escreval* e o comando *leia*.

3.1 Atividade 1

Elaborar um algoritmo para um cadastro simples de uma pessoa, onde devem constar os dados pessoais como: nome, endereço e contato.

Para isso fizemos um exemplo na lousa de como seria este cadastro e tivemos o seguinte algoritmo:

Algoritmo: *Cadastro simples de uma pessoa*

<pre> var nome, rua, compl, email, cid, est: caractere num, tel, cep: real inicio escreval (" *Ficha de cadastro*") escreval(" ")//pula uma linha escreva(" Nome: ") leia (nome) escreval(" ") escreval (" Endereço ") escreval(" ") escreva (" Rua: ") leia(rua) escreva(" Num: ") leia (num) </pre>	<pre> escreva (" Complemento: ") leia (compl) escreva (" Cidade: ") leia (cid) escreva (" Cep: ") leia (cep) escreva (" Estado: ") leia (est) escreval (" ") escreval (" Tel.: ") leia (tel) escreva (" ") escreva (" email: ") leia (email) finalgoritmo </pre>
---	--

Este momento de construir um algoritmo usando um programa se mostrou um tanto quanto trabalhoso, porque mostrou a fragilidade que os alunos têm para desenvolver uma atividade no computador fora das situações que eles estão acostumados, como as redes sociais. Portanto esta primeira atividade teve um tempo de desenvolvimento acima do que se tinha programado.

Ainda dentro desta ideia de apresentar o programa VisuAlg para os alunos foram propostas mais duas atividades bem simples, mas agora envolvendo situações matemáticas, e descritas a seguir:

3.2 Atividade 2

Dados dois números reais realizar as quatro operações aritméticas, calcular a média aritmética entre os números e imprimir os resultados na tela.

A proposta desta atividade é mostrar para o aluno como podemos construir um algoritmo que resolva as quatro operações, os comandos que são

utilizados para que o aluno entenda como o programa realiza essas operações e com isso reforçar esses conteúdos básicos, porém agora com o auxílio de um programa desenvolvido por eles.

Nessa parte não tive maiores problemas, porque os alunos já estavam começando a entender o programa VisuAlg e seus comandos e com isso foi aproveitada mais a parte de se pensar algoritmicamente como realizar as quatro operações, o que envolve de fato o objetivo deste trabalho.

Neste momento começaram a surgir questões do tipo: o que acontece se fizer isso, o que acontece se fizer aquilo, e estas questões permitem ao professor explorar os conteúdos matemáticos envolvidos nos algoritmos e que são dinamizados com a utilização das tecnologias digitais. Assim tivemos o seguinte resultado na construção do algoritmo no VisuAlg:

```
Algoritmo: As quatro operações e a média aritmética entre dois números  
var  
num1, num2, soma, sub, mult, dv, media: real  
inicio  
escreva ( " Digite um número: " )  
leia (num1)  
escreva ( " Digite outro número: " )  
leia (num2)  
soma ← num1 + num2  
sub ← num1 - num2  
mult ← num1 * num2  
dv ← num1 / num2  
media ← soma/2  
escreva ( " A soma é ",soma," , a subtração é ",sub, " , a multiplicação é  
",mult," , a divisão é ", dv, " e a média é ",media)  
fimalgoritmo
```

3.3 Atividade 3

Dados dois números reais, encontrar qual deles é o maior.

Esta atividade proposta teve como elemento principal a estrutura de comandos de decisão, *se/então/senão*. O objetivo foi trabalhar questões de lógica, pois percebe-se que muitas das dificuldades apresentadas pelos alunos no desenvolvimento de conceitos matemáticos estão diretamente relacionadas com a deficiência que o aluno tem na sua formação em questões que envolvem valores lógicos, situações essas que são pouco exploradas na sua vida escolar.

Desenvolver a habilidade do raciocínio lógico permite que o aluno obtenha mais avanços na construção das habilidades e competências em matemática.

Isso ficou evidente na realização desta atividade, pois muitos alunos tiveram dificuldade na construção de um algoritmo que a permite identificar dentre dois números, qual número é o maior de uma maneira simples, usando conceitos matemáticos básicos e valores lógicos.

A dificuldade maior ficou por conta de que muitos alunos não entendem que quando é dado uma condição para algo acontecer espera-se que aconteça ou não aconteça. Isso fica fácil de perceber quando usamos a estrutura de comandos de decisão, *se/então/senão*, pois mostramos a condição, se isso acontecer então faça isso, senão (quer dizer que não aconteceu) faça aquilo ou não faça nada.

Existem também as dificuldades que os alunos carregam ao longo da sua formação matemática, uma delas é a identificação e utilização de símbolos. Nessa atividade o que me chamou a atenção, foi o fato de os alunos não possuírem a habilidade de trabalhar com os símbolos maior que ($>$) e menor que ($<$). Fato esse, que pode demandar uma atenção e um tempo maior para a realização da atividade.

Desta forma foi construído o seguinte algoritmo no VisuAlg:

Algoritmo: *Qual número é o maior*

```
var
num1, num2: real
inicio
escreva (" Digite um número: " )
leia (num1)
escreva (" Digite outro número: " )
leia (num2)
se (num1 > num2) entao
    escreva (" O maior número digitado foi ", num1)
senao
    escreva (" O maior número digitado foi ", num2)
fimse
finalgoritmo
```

3.4 Atividade 4

Ampliação da atividade anterior. Dados três números reais, encontrar qual deles é o maior.

De acordo com o observado na atividade anterior e percebendo o envolvimento dos alunos, foi sugerido dentro da mesma atividade que eles fizessem o mesmo, porém agora usando três números. Isso permitiu aumentar a complexidade do problema e reforçar mais ainda as ideias dos comandos de decisão, contribuindo para a compreensão dos conteúdos básicos de matemática envolvidos e também dos conceitos lógicos. Assim se obteve o seguinte algoritmo no VisuAlg:

```
Algoritmo: Qual número é o maior  
var  
num1, num2, num3: real  
inicio  
escreval (" Digite três números: " )  
leia (num1, num2, num3)  
se (num1 > num2) e (num1 > num3) entao  
    escreval (" O maior número digitado foi ", num1)  
senao  
    se (num2 > num3) entao  
        escreval (" O maior número digitado foi ", num2)  
    senao  
        escreval (" O maior número digitado foi ", num3)  
fimse  
fimse  
fimalgoritmo
```

Essa situação, que foi provocada, também teve o objetivo de mostrar a eles o quanto é importante entendermos um problema menor para construirmos as soluções de problemas maiores. Isso realmente foi verificado nas atitudes dos alunos para resolver essa situação, pois como já tinham resolvido a atividade com dois números, estavam seguros de que não seria tão difícil chegar a um algoritmo para resolver o problema para três números. O resultado foi que muitos conseguiram realmente desenvolver essa atividade sozinhos, ou seja, sem o auxílio do professor.

Além disso, podemos explorar mais situações que no momento da aplicação da atividade não foram observadas, como por exemplo, se os dois números ou três números propostos fossem iguais. No caso de três números, como ficaria o algoritmo se dois dos números fossem iguais? Todas essas situações e algumas outras podem ser problematizadas durante a aplicação desta atividade.

Por outro lado essas atividades também tinham como intenção prepará-los para as atividades que seriam trabalhadas no segundo momento.

No segundo momento, dado que os alunos já estavam familiarizados com os conceitos e aplicações na construção de algoritmos e o programa VisuAlg, passamos a analisar conteúdos matemáticos com um pouco mais de dificuldade e raciocínio, valendo-se do uso do processo algorítmico e do programa VisuAlg para desenvolver e dinamizar a construção destes conhecimentos.

Começamos assim pela seguinte situação matemática que foi colocada aos alunos e propondo que tentassem apresentá-la a partir de um algoritmo implementado no VisuAlg.

3.5 Atividade 5

Dados três números positivos, verificar se é possível formar um triângulo. Caso se confirme que é um triângulo, calcular seu perímetro e área.

Para isso foi mostrada a eles a relação que existe entre os lados de um triângulo qualquer (desigualdade triangular), como se calcula o perímetro e como se calcula a área usando a fórmula de Herón.

Desigualdade Triangular: *Dadas três medidas, é possível se obter um triângulo se e somente se, as medidas obedecerem à seguinte regra: uma medida deve ser sempre menor que a soma das outras duas medidas.*

Perímetro de uma figura plana: *É a soma das medidas dos lados de uma figura.*

Área de um triângulo (Fórmula de Herón):

$$A = \sqrt{p \cdot (p - a) \cdot (p - b) \cdot (p - c)},$$

onde p é o semiperímetro do triângulo, metade da medida do perímetro, e a , b e c são as medidas dos lados do triângulo.

Dada esta situação, podemos resolvê-la com o seguinte algoritmo com implementação no VisuAlg:

Algoritmo: *Construindo Triângulos*

var

a, b, c, p, area: real

inicio

escreval (" Digite o valores das medidas dos lados do triângulo ")

leia (a,b,c)

se (a < b + c) e (b < a + c) e (c < a + b) entao

escreval (a," ",b," e ",c," formam um triângulo ")

p ← (a + b + c)/2 // Semiperímetro do triângulo

*area ← raizq(p * (p - a) * (p - b) * (p - c)) // Fórmula de Herón*

*escreval (" Com perímetro igual a " ,2*p)*

escreval (" Com área igual a " ,area)

senao

escreval (a," ",b," e ",c," não formam um triângulo ")

fimse

fimalgoritmo

Importante observar, o que podemos concluir se os números fornecidos forem negativos? Apesar de na atividade serem pedidos três números positivos, nada impede de o aluno querer testar com números negativos. De qualquer forma, a condição de existência não falha. Sendo assim, se os números não obedecerem à desigualdade triangular, o algoritmo vai retornar que não é possível formar um triângulo com os números dados. Uma opção interessante é pedir para o aluno testar a desigualdade manualmente, para ver o que acontece.

Assim como as atividades propostas no primeiro momento, as atividades do segundo momento também promovem possibilidades de desenvolver questões que ampliem os conteúdos matemáticos envolvidos. Por exemplo: problematizar com os alunos como tornar mais eficiente o algoritmo para que este não só verifique se é um triângulo, mas que também, se for um triângulo, que classifique-o de acordo com seus ângulos, ou seja, retângulo, acutângulo ou obtusângulo, ou também, que classifique-o de acordo com seus lados, ou seja, escaleno, isósceles ou equilátero. Com isso se consegue trabalhar em

uma mesma atividade outros conteúdos matemático que estão relacionados com a atividade proposta e ao mesmo tempo desafiar os alunos a encontrarem os caminhos mais lógicos e eficientes para a ampliação dos algoritmos.

3.6 Atividade 6

Ampliação do algoritmo da atividade 5, de forma que este também retorne se o triângulo formado é retângulo ou não.

Para isso foi apresentado aos alunos como podemos identificar se um triângulo é retângulo. Esta problematização possibilitou incluir nesta atividade o Teorema de Pitágoras e a recíproca do Teorema de Pitágoras:

Teorema de Pitágoras: *Se um triângulo é retângulo, então o comprimento de um de seus lados ao quadrado é igual à soma dos quadrados dos comprimentos dos outros dois lados.*

Recíproca do Teorema de Pitágoras: *Dado um triângulo, se o quadrado do comprimento de um dos lados é igual a soma dos quadrados dos comprimentos dos outros dois lados, então o triângulo é retângulo.*

Feito isso, o algoritmo anterior ampliado ficou:

Algoritmo: *Construindo Triângulos*

```

var
a, b, c, p, area: real
inicio
escreval (" Digite os valores dos lados do triângulo " )
leia (a,b,c)
se (a < b + c) e (b < a + c) e (c < a + b) entao
    escreval (a," ",b," e ",c," formam um triângulo " )
    se (a*a = b*b + c*c) ou (b*b = a*a + c*c) ou (c*c = a*a +
b*b) entao// Recíproca do Teorema de Pitágoras
        escreval(" E também é um triângulo retângulo " )
        p ← (a + b + c)/2 // Semiperímetro do triângulo
        area ← raizq(p*(p - a) * (p - b) * (p - c))// Fórmula de Herón
        escreval (" Com perímetro igual a ",2*p)

```

```

    escreval ( " Com área igual a " ,area)
senao
    escreval ( " Que não é um triângulo retângulo " )
    p ← (a + b + c)/2
    escreval ( " Com um perímetro igual à " ,2*p)
    area ← raizq(p * (p - a) * (p - b) * (p - c))
    escreval ( " E uma área igual à " ,area)
fimse
senao
    escreval (a," ,",b," e ",c," não formam um triângulo " )
fimse
finalgoritmo

```

3.7 Atividade 7

Construir um algoritmo que resolva uma equação quadrática, ou seja, resolver $ax^2 + bx + c = 0$, para a, b e c números reais e $a \neq 0$.

Esta atividade proposta aos alunos, tem uma característica mais algébrica. Primeiramente foi apresentado aos alunos uma das maneiras de resolver uma equação quadrática, sendo esta a mais praticada nos currículos das propostas pedagógicas para matemática das escolas, que tem a seguinte forma de resolução:

Seja $ax^2 + bx + c = 0$, com $a, b, c \in \mathbb{R}$ com $a \neq 0$, então

$$x = \frac{-b + \sqrt{b^2 - 4ac}}{2a} \text{ ou } x = \frac{-b - \sqrt{b^2 - 4ac}}{2a}$$

Aqui temos mais uma atividade que permite uma abordagem de conteúdos matemáticos, bem dinâmica, utilizando algoritmos e programação. Até por que, a forma de se resolver uma equação quadrática apresentada na trajetória escolar dos alunos na maioria das instituições de ensino básico, é exatamente algorítmica. Portanto, desenvolver esse conteúdo usando as ferramentas propostas nesta dissertação pode melhorar ainda mais esse entendimento.

Na sua aplicação percebeu-se claramente essa análise, pois os alunos foram verificando passo a passo o que seria necessário na resolução. A partir desta análise foi elaborado o seguinte algoritmo com o auxílio do programa VisuAlg:

Algoritmo: *Equação quadrática*

var

a, b, c, d, r, x1, x2: real

inicio

escreval (" Digite os valores de a, b e c da sua equação quadrática: ")

leia (a,b,c)

*d ← b*b - 4*a*c*

se (d >= 0) entao

r ← raizq(d)

*escreval (" x1 = ", (- b + r)/(2*a)," e ")*

*escreval (" x2 = ", (- b - r)/(2*a))*

senao

escreval (" A equação não possui valores de x reais para equação com coeficientes ", a," ,b," e ",c)

fimse

fimalgoritmo

Com esse algoritmo surgiram algumas situações para serem discutidas com os alunos. O trecho em que as variáveis **d** e **r** recebem os cálculos para serem usados depois, isto envolve uma ideia algorítmica de se guardar os cálculos. Pode-se entender como eficiência da operação, pois não precisa calcular as operações repetidas vezes, basta calcular uma vez só e guardar o valor para ser usado mais tarde. Mostrar esta estratégia ao aluno desenvolve nele habilidades de sistematização, para não ficar fazendo várias vezes os mesmos cálculos.

Uma outra situação que pode ser abordada, é o que acontece com o algoritmo se o valor do coeficiente **a** for zero. Se isso acontecer, o programa vai retornar uma mensagem, *operação inválida*, ele entende que dividir por

zero é uma operação inválida. Com isso reforçamos ainda mais o conceito de que não existe divisão por zero.

Pode-se assim propor alguma alternativa para que quando se for usar o algoritmo ou programa desenvolvido, ele avise que o valor do coeficiente **a** da equação quadrática $ax^2 + bx + c = 0$ deve ser diferente de zero.

Avançando agora para o terceiro e último momento do trabalho de campo com os alunos. Foram preparadas algumas atividades mais elaboradas do ponto de vista matemático e algorítmico. Nesse momento já percebemos uma maior habilidade do aluno com o programa VisuAlg, proporcionando assim mais foco na construção dos algoritmos e nos conteúdos matemáticos envolvidos. Assim como nos dois momentos anteriores, foram propostas duas atividades, detalhadas a seguir:

3.8 Atividade 8

Construir um algoritmo que calcule e explore as particularidades de uma função quadrática, que são: zeros da função, concavidade da parábola e ponto de máximo ou mínimo.

Novamente, primeiro foram trabalhados os conceitos destas particularidades, apresentando as definições e forma de cálculo, começando primeiro por definir uma função quadrática.

Função quadrática: É uma função $f : \mathbb{R} \rightarrow \mathbb{R}$, tal que, $f(x) = ax^2 + bx + c$ com a, b, c números reais e $a \neq 0$.

Zeros de uma função quadrática: Os zeros de uma função quadrática, são os valores de x que satisfazem $f(x) = 0$, ou seja, os valores de x que resolvem a seguinte equação: $ax^2 + bx + c = 0$.

Os zeros de uma função quadrática são as raízes da equação anterior.

Concavidade de uma Parábola: O gráfico de uma função quadrática é uma parábola. Sua concavidade é definida conforme o valor de a . Se $a > 0$, concavidade voltada para cima, ou se $a < 0$, concavidade voltada para baixo.

Ponto de máximo ou ponto de mínimo de uma função quadrática: O ponto de maior ordenada no gráfico da função quadrática é definido como ponto de máximo, e isso acontece quando a parábola tem concavidade voltada

para baixo, ou seja, quando $a < 0$. Do contrário, de forma análoga, definimos ponto de mínimo. Tal ponto é chamado em ambos os casos, de vértice da parábola

Vamos representar as coordenadas do ponto de máximo ou de mínimo como (x_v, y_v) , que são as coordenadas do vértice da parábola que representa esse ponto e podem ser calculados da seguinte forma:

$$x_v = -\frac{b}{2a} \text{ e } y_v = -\frac{\sqrt{b^2 - 4ac}}{4a}.$$

As definições e informações acima foram suficientes para que os alunos começassem e concluíssem o algoritmo proposto para esse momento e que ao final teve o seguinte resultado:

Algoritmo: *Função quadrática*

var

a, b, c, d, r, x1, x2, xv, yv: real

inicio

escreval (" Digite os valores de a, b e c da função quadrática: ")

leia (a,b,c)

se (a = 0) entao

escreval (" O valor de 'a' não pode ser zero para uma função quadrática

")

senao

*d ← b*b - 4*a*c*

se (d >= 0) entao

r ← raizq(d)

se (a > 0) entao

escreval (" A função quadrática tem sua parábola com concavidade voltada para cima então, tem ponto de mínimo ")

*x1 ← (- b + r) / (2*a)*

*x2 ← (- b - r) / (2*a)*

*xv ← - b / (2*a)*

*yv ← - d / (4*a)*

escreval (" Os zeros da função são $x_1 =$,x1," e $x_2 =$,x2)
 escreval (" O ponto de mínimo tem coordenadas (" ,xv," , " ,yv,"
) ")

senao

escreval (" A função quadrática tem sua parábola com con-
 cavidade voltada para baixo então, tem ponto de máximo. ")

$$x1 \leftarrow (-b + r)/(2*a)$$

$$x2 \leftarrow (-b - r)/(2*a)$$

$$xv \leftarrow -b/(2*a)$$

$$yv \leftarrow -d/(4*a)$$

escreval (" Os zeros da função são $x_1 =$,x1," e $x_2 =$ "
 ,x2)

escreval (" O ponto de máximo tem coordenadas (" ,xv," , " ,yv,"
) ")

fimse

senao

escreval (" A função quadrática não possui zeros no con-
 junto dos Reais.")

se ($a > 0$) entao

escreval (" A função quadrática tem sua parábola com con-
 cavidade voltada para cima então, tem ponto de mínimo.")

$$xv \leftarrow -b/(2*a)$$

$$yv \leftarrow -d/(4*a)$$

escreval (" O ponto de mínimo tem coordenadas (" ,xv," , "
 ,yv,") .")

senao

escreval (" A função quadrática tem sua parábola com con-
 cavidade para baixo então tem ponto de máximo.")

$$xv \leftarrow -b/(2*a)$$

$$yv \leftarrow -d/(4*a)$$

escreval (" O ponto de máximo tem coordenadas (" ,xv,
 " , " ,yv,") .")

fimse

fimse

fimse

fimalgoritmo

Para esta atividade foi necessário um tempo maior no seu desenvolvimento, até porque envolve várias situações e também por ser trabalhada em várias etapas, como foi sugerido aos alunos.

Começou-se primeiro pela definição de função quadrática, para que os alunos entendessem o que é importante ser estudado.

A discussão neste início com eles ficou por conta dos valores que podem ser atribuídos ao coeficiente a , pois na definição de função quadrática colocamos que a não pode ser zero, e isto pode ser problematizado de algumas maneiras com os alunos. Uma delas, foi mostrar que se o valor do coeficiente a for zero temos que a função deixa de ser quadrática e se torna uma função afim, mostrando para o aluno uma das diferenças entre estas duas funções.

A partir disso, foi sugerido colocar no algoritmo uma parte que retorna que o coeficiente a não pode ser zero, colocando uma mensagem para que se digite outro valor para ele.

Cada etapa do algoritmo promove um estudo detalhado das situações que são evidenciadas em uma função quadrática, possibilitando ao professor mostrar as condições sobre os valores do coeficiente a e sobre os valores do discriminante. Esses interferem diretamente na análise das situações encontradas em uma função quadrática com isso, pode-se proporcionar várias inserções por parte do professor com objetivo de fazer o aluno refletir sobre as implicações encontradas em cada caso.

Para finalizar esse terceiro momento foram apresentadas aos alunos as ideias de repetição em um algoritmo, que dentro do processo algorítmico é conhecido como laço, e que foram as últimas situações algorítmicas trabalhadas com eles nesta dissertação.

A proposta é desenvolver conteúdos matemáticos que possibilitam as estruturas de repetição, como por exemplo as sequências numéricas ou progressões numéricas, situações matemáticas muito recorrentes na vida escolar dos alunos.

Este momento do trabalho ficou um pouco comprometido por conta de algumas particularidades da escola. Uma delas foi o final de bimestre que fez com que os alunos ficassem preocupados com as provas oficiais e consequentemente ficavam indisponíveis para as atividades desenvolvidas por este trabalho. Lembrando que as práticas de campo foram realizadas em horário de aula dos alunos com a autorização da gestão escolar.

Nesse período os professores no final de bimestre trabalham revisões de provas e atividades para complementação de notas com seus alunos, dificultando assim a participação destes neste trabalho.

Apesar disso, três alunos ainda se dispuseram a desenvolver algumas atividades e declararam que a participação neste trabalho os ajudou a compreender alguns conteúdos matemáticos que antes ainda não estavam muito esclarecidos.

3.9 Atividade 9

Construir algoritmos que produzam sequências de números, aplicando as estruturas de repetição.

Para iniciar, foi pedido a eles que pensassem em um algoritmo que produzisse a sequência de números inteiros de 0 a 100. No entanto, desenvolvi antes com eles três maneiras de fazer um algoritmo e produzir uma sequência menor, de 0 a 10.

Começamos com o seguinte algoritmo usando o comando para-até-faça:

Algoritmo: *Números inteiros de 0 a 10*

var

x: inteiro

inicio

para x de 0 ate 10 faça

escreva (x)

fimpara

fimalgoritmo

Uma outra forma de produzir um algoritmo que execute a mesma situação é usando o comando enquanto-faça, como mostrado abaixo:

Algoritmo: *Números inteiros de 0 a 10*

var

x: inteiro

inicio

$x \leftarrow 0$

enquanto $x \leq 10$ *faca*

escreva (x)

$x \leftarrow (x + 1)$

fimenquanto

fimalgoritmo

Agora usando outro comando de repetição que é o repita-até, temos o seguinte algoritmo:

Algoritmo: *Números inteiros de 0 a 10*

var

x: inteiro

inicio

$x \leftarrow 0$

repita

escreva (x)

$x \leftarrow x + 1$

ate $x > 10$

fimrepita

fimalgoritmo

Depois de apresentadas estas maneiras de construir um algoritmo para gerar uma sequência de números inteiros de 0 a 10, os alunos fizeram as alterações necessárias para se produzir a sequência proposta no início desta

atividade, que era a dos números inteiros de 0 a 100. Depois foi problematizado com eles outras situações que podem ser desenvolvidas usando as mesmas ideias neste algoritmo como, por exemplo, produzir uma sequência de números inteiros pares de 0 a 100, ou de números ímpares, deixando a escolha de qual formato de algoritmo usar por conta deles.

Esse momento proporcionou aos alunos uma análise sobre as construções de sequências, tendo eles que identificar como alterar os algoritmos acima para produzir estas sequências.

No segundo e no terceiro exemplo não tiveram dificuldade, pois perceberam logo que bastava alterar o número inicial da sequência e o valor somado ao número inicial. No caso da sequência par, o número inicial é o zero e somando dois; na sequência ímpar, o valor inicial é o um, também somando dois.

No primeiro exemplo precisava acrescentar antes do **faca** o comando **passo**, que indica de quanto em quanto a sequência vai aumentando, neste caso de dois em dois. No caso da sequência ser ímpar, alterar também o número inicial. Portanto, são vários os desdobramentos que esta atividade permite, e fornece uma situação muito rica para o professor na sua prática docente.

3.10 Atividade 10

Desenvolver dois algoritmos, um deles para produzir um Progressão Aritmética (P.A.) e a soma dos seus termos, dados o primeiro termo, a razão e a quantidade de termos. O outro, para produzir uma Progressão Geométrica (P.G.) e também a soma dos seus termos, dados o primeiro termo, a razão e a quantidade de termos.

Esta proposta de atividade não pôde ser trabalhada com os alunos devido ao término do segundo bimestre, quando os alunos estavam encerrando suas atividades avaliativas e, conseqüentemente, preparando-se para as férias, dificultando assim o acesso a eles.

Contudo, o desenvolvimento desta atividade seguiria as mesmas ideias das atividades propostas anteriormente, realizando os mesmos passos, porém agora trabalhando outros conteúdos matemáticos. A atividade poderia ter

sido realizada da seguinte forma:

Primeiro seriam trabalhadas com os alunos as definições das progressões aritméticas e geométricas e as respectivas somas dos seus termos, como segue:

Progressão aritmética P.A.: *É uma sequência numérica, onde dado o 1º termo desta sequência, os seus sucessores são obtidos somando sempre ao anterior uma mesma quantidade, chamada de razão.*

Soma dos termos de uma P.A.: *É a soma de n (n = número de termos da P.A.) termos desta sequência.*

Progressão geométrica P.G.: *É uma sequência numérica, onde dado o 1º termo desta sequência, os seus sucessores são obtidos multiplicando o anterior sempre por uma mesma quantidade, chamada de razão.*

Soma dos termos de uma P.G.: *É a soma de n (n = número de termos da P.G.) termos desta sequência.*

A segunda etapa seria trabalhar com os alunos os possíveis algoritmos utilizando o VisuAlg. Para produzir estas situações, os algoritmos possíveis seriam:

```
Algoritmo: Progressão Aritmética e soma dos seus termos
var
a, r, n, k, soma: inteiro
inicio
escreval ( "Digite os respectivos valores do 1º termo, da razão e da quantidade de termos da P.A." )
leia (a,r,n)
soma ← 0
para k de 1 ate n faça
    escreval (a)
    soma ← soma + a
    a ← a + r
fimpara
escreval ( " " )
escreval ( "A soma é: " ,soma)
finalgoritmo
```

Algoritmo: *Progressão Geométrica e soma dos seus termos*

var

a, r, n, k: inteiro

soma: real

inicio

escreval ("Digite os respectivos valores do 1º termo, da razão e da quantidade de termos da P.G.: ")

leia (a,r,n)

soma ← 0

para k de 1 ate n faça

escreva (a)

soma ← soma + a

*a ← a*r*

fimpara

escreval (" ")

escreval ("A soma é: ",soma)

fimalgoritmo

Como esses algoritmos não foram trabalhados com os alunos, não foi possível observar seus desdobramentos na prática, porém as atividades propostas acima promovem várias situações que podem ser exploradas pelo professor no momento do seu desenvolvimento com os alunos.

No caso de uma P.A., propor que os alunos observem o que acontece se a razão for negativa, como ficariam seus termos e sua soma. Numa P.G., o que acontece quando a razão é negativa ou quando for maior que zero e menor que um, como ficariam seus termos e sua soma, são algumas das muitas situações que podem ser trabalhadas.

Uma situação que pode ser interessante e não pensada na atividade 10, que os alunos construíssem um algoritmo que calculasse a soma dos termos de uma P.A. ou P.G. usando a fórmula que eles aprendem nos livros, pode-se então comparar os resultados dos dois algoritmos e verificar que as somas são iguais.

As atividades desenvolvidas fornecem muitas propostas que promovem

uma dinâmica de trabalho que procura valorizar as construções algorítmicas como estratégias de ensino para fortalecer os conhecimentos matemáticos sugeridos no Ensino Básico. Ao mesmo tempo, propiciam inserir o algoritmo na vida escolar do aluno, como uma forma de potencializar o seu raciocínio lógico e também promover o acesso a essa grande ideia milenar, que hoje é responsável pelas tecnologias computacionais encontradas no nosso dia a dia.

Os resultados encontrados no desenvolvimento das atividades, permitem projetar um otimismo bastante promissor de acordo com os objetivos desta pesquisa.

Por fim, encerra-se este desenvolvimento de campo, de forma que muitas outras atividades poderiam ter sido desenvolvidas, porém por questão de tempo e disponibilidade dos alunos não foram trabalhadas, mas estas atividades serão deixadas como propostas de trabalho nos anexos desta dissertação.

Capítulo 6

CONSIDERAÇÕES FINAIS

Esta pesquisa procurou apresentar uma estratégia de construção do conhecimento matemático e do desenvolvimento do raciocínio lógico, usando da elaboração de algoritmos e sua implementação no programa VisuAlg.

Foram apresentadas várias situações onde o professor pode trabalhar com o aluno muitos conceitos matemáticos que estão diretamente relacionados com o desenvolvimento de algoritmos. Também com o uso do programa de implementação VisuAlg, estas situações podem ser trabalhadas de forma mais dinâmica, proporcionando uma aula mais atrativa ao aluno.

Cabe destacar que este tipo de trabalho depende bastante da estrutura escolar, onde necessariamente a escola precisa estar equipada com um laboratório de informática, com computadores suficientes, para que os alunos estejam no mínimo organizados em duplas, que é o recomendado, para proporcionar uma aprendizagem efetiva.

Por fim, esse trabalho procurou mostrar uma forma de abordagem de conteúdos, que pode contribuir em muito para a construção do conhecimento matemático do aluno, além disso promove o desenvolvimento do raciocínio lógico e também um primeiro contato com um conhecimento importantíssimo no mundo das tecnologias digitais, que basicamente tem como princípio o desenvolvimento de algoritmos, que é a programação.

Referências Bibliográficas

- [1] *Algorithms – Business by numbers*, The Economist, (2007).
www.economist.com/node/9795140.
- [2] *The national curriculum in England*, (2014).
www.gov.uk/government/publications/national-curriculum-in-england-framework-for-key-stages-1-to-4.
- [3] R. S. ALMEIDA, *Aprendendo Algoritmos com o VisuAlg.*, Ciência Moderna Ltda, 2013.
- [4] D. BERLINSKI, *O advento do algoritmo: a ideia que governa o mundo*, Editora Globo, 2002.
- [5] S. CHAMBERS, *Escolas da Inglaterra ensinam alunos de 5 anos a programar*, Revista Exame, (2014).
<http://exame.abril.com.br/tecnologia/noticias/escolas-da-inglaterra-ensinam-alunos-de-5-anos-a-programar>.
- [6] R. S. A. FRANÇA AND H. COSTA, *Ensino de computação na educação básica no brasil: Um mapeamento sistemático*, in XXI Workshop sobre Educação em Computação, 2013.
- [7] V. GIRALDO, P. CAETANO, AND F. MATTOS, *Recursos computacionais no ensino de matemática*, SBM, 2012.
- [8] A. P. JAHN AND N. ALLEVATO, *Tecnologias e Educação Matemática*, SBEM, 2010.

- [9] D. E. KNUTH, *The art of Computer Programming*, vol. 1, Addison-Wesley, 1973.
- [10] W. S. LEAL, *O ensino de algoritmos no ensino médio: por que não?*, Universidade do Grande Rio, 2009.
- [11] P. NOGUEIRA, *Abracadabra: os algoritmos estão dominando o mundo*, Revista do Brasil, (2013). www.redebrasilatual.com.br/revistas/85/abracadabra-eletronico-861.html.
- [12] P. RIBENBOIM, *Números primos, amigos que causam problemas*, SBM, 2015.
- [13] O. TAVARES, *O uso cotidiano do algoritmo*, Gazeta do Povo, (2011). www.gazetadopovo.com.br/vida-e-cidadania/o-uso-cotidiano-do-algoritmo-4x3n9sw4bkhoam6fzqcp27mfi.
- [14] B. TONET AND C. KOLIVER, *Introdução aos Algoritmos*, NAPRO – Universidade de Caxias do Sul, 2013.

Apêndice A

PROPOSTA DE ATIVIDADES

1 MDC e MMC

Algoritmo de Euclides - calcular MDC e MMC entre dois números inteiros positivos.

Euclides viveu por volta de 300a.C. na Grécia, e deve ter morrido por volta de 295a.C., talvez em Alexandria, uma cidade grega, hoje no Egito. Não se sabe onde ele nasceu, nada sobre a sua vida pessoal, exceto que ele fora influenciado pelos ensinamentos de Platão. Sabe-se também que ele tinha uma escola de matemática em Alexandria, que foi dirigida por seu discípulo Apolônio após a sua morte. É uma pena que se saiba tão pouco sobre sua vida. Mas sabe-se o mais importante, que é sobre o seu trabalho. Ele inclui os *Elementos*, dividido em 13 partes. Neste livro, Euclides usou o “método axiomático”. Algumas afirmações sobre os conceitos da teoria, considerados “evidentes” foram tomados como axiomas. Por dedução lógica, executada com (quase) infalível rigor, Euclides provou seus Teoremas. Esse método tornou-se fundamental para o desenvolvimento da matemática e é mundialmente usado até hoje. As partes 1 a 6 dos *Elementos* são sobre geometria plana, as partes 7 a 9 são sobre números, onde entra o *algoritmo de Euclides*, o livro 10 é sobre o conjunto dos números irracionais, as outras partes são sobre geometria espacial e contêm, entre outras coisas, uma descrição dos cinco poliedros convexos regulares que Platão descreveu. Esses livros foram traduzidos para o latim e árabe e foram cuidadosamente estudados pelos matemáticos [12].

Máximo Divisor Comum - MDC: *Sejam a e b inteiros positivos. Um inteiro positivo d é o máximo divisor comum de a e b se e somente se verifica:*

- (i) $d|a$ e $d|b$.
- (ii) Se $d'|a$ e $d'|b$, então $d'|d$.

Mínimo Múltiplo Comum - MMC: *Sejam a e b inteiros positivos. Um inteiro positivo m é o mínimo múltiplo comum de a e b se e somente se verifica:*

- (i) $a|m$ e $b|m$.

(ii) Se $a|m'$ e $b|m'$, então $m|m'$.

O algoritmo de Euclides consiste num método algorítmico para se obter o MDC (Máximo Divisor Comum) de dois números. O método algorítmico é o seguinte:

Sejam a e b dois números inteiros positivos com $a > b$. Então temos as seguintes divisões sucessivas

$$a = b.q_1 + r_1$$

$$b = r_1.q_2 + r_2$$

$$r_1 = r_2.q_3 + r_3$$

$$r_2 = r_3.q_4 + r_4$$

$$r_3 = r_4.q_5 + r_5$$

.....

Assim sucessivamente até que se chegue em um $r_n = 0$, e portanto, o último resto diferente de zero, ou seja, r_{n-1} será o $mdc(a, b)$.

Proposta de atividade:

a) Construa, usando o VisuAlg, um algoritmo que retorne o mdc dados os números a e b de entrada.

b) Existe uma relação muito importante entre o mdc e o mmc de dois inteiros positivos a e b , que é a seguinte:

$$mmc(a, b).mdc(a, b) = a.b.$$

Use esta relação para que o algoritmo construído para calcular o mdc também calcule o mmc de dois inteiros positivos.

Algoritmo sugerido: *Calcular MDC e MMC entre dois números inteiros positivos*

var

a, b, maior, mdc, aux: inteiro

mmc: real

inicio

escreval ("Digite os números para o cálculo do MDC: ")

leia (a,b)

se (a > b) entao

maior ← a

mdc ← b

senao

maior ← b

mdc ← a

fimse

enquanto (maior mod mdc > 0) faca

aux ← mdc

mdc ← maior mod mdc

maior ← aux

fimenquanto

*mmc ← (a*b)/mdc*

escreval ("O MDC entre ",a," e ",b," é: ",mdc)

escreval ("O MMC entre ",a," e ",b," é: ",mmc)

fimalgoritmo

2 Sequência de Fibonacci e número áureo

Sequência de Fibonacci - Construir um algoritmo que retorne os primeiros termos da sequência de Fibonacci e também uma aproximação para o número de ouro.

O matemático Leonardo Pisa (1170 - 1250), conhecido como Fibonacci, propôs no século XIII a sequência numérica abaixo:

(1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, ...).

Essa sequência tem uma lei de formação simples: cada elemento, a partir do terceiro, é obtido somando-se os dois anteriores. No caso da sequência de Fibonacci os dois primeiros são iguais a 1. Veja:

$1 + 1 = 2$, $2 + 1 = 3$, $3 + 2 = 5$ e assim por diante.

Desde o século XIII, muitos matemáticos, além do próprio Fibonacci, dedicaram-se ao estudo da sequência que foi proposta, e foram encontradas inúmeras aplicações na análise de mercados financeiros, na ciência da computação e na teoria dos jogos.

Também aparece em configurações biológicas, como, por exemplo, na disposição dos galhos das árvores ou das folhas em uma haste, no arranjo do cone da alcachofra, do abacaxi, ou no desenrolar da samambaia.

Proposta de atividade:

a) Construa um algoritmo no VisuAlg que imprima a sequência de Fibonacci para uma quantidade n de termos dada.

b) Existe um número irracional chamado de número de ouro ou número áureo, representado pela letra grega φ (fi), e podemos achar uma aproximação para ele dividindo o termo sucessor pelo anterior da sequência de Fibonacci. Por ser encontrado em estudos de crescimento, o número de ouro ganhou um status de “ideal”, sendo alvo de pesquisadores, artistas e escritores. O fato de ser apoiado pela matemática é que o torna fascinante.

Use o algoritmo que foi construído para a sequência de Fibonacci, e encontre uma aproximação do número de ouro, para uma quantidade n de termos dada, ou seja, se temos 10 termos na sequência de Fibonacci, então uma aproximação para o número de ouro é a divisão do 10º termo pelo 9º termo.

Algoritmo sugerido: *Sequência de Fibonacci e número de ouro aproximado.*

```
var
n, f1, f2, aux, k: inteiro
ouro: real
inicio
escreval ("Digite a quantidade n de termos da sequência de Fibonacci." )
escreval (" " ) // Pula linha na hora da impressão.
leia (n)
f1 ← 0
f2 ← 1
escreval ("A sequência de Fibonacci com ",n," termos é:" )
para k de 1 ate n faça
    aux ← f1 + f2
    escreva (f2)
    f1 ← f2
    f2 ← aux
fimpara
ouro ← f2/f1
escreval (" " )
escreval ("O número áureo aproximado para esta sequência de ",n," termos é: ",ouro)
finalgoritmo
```

3 Raiz quadrada

Método Babilônico da raiz quadrada - Construir algoritmo que retorne a raiz quadrada de um número usando o método babilônico.

A Babilônia era uma das cidades da Mesopotâmia, região ao sul da Ásia entre o rio Tigre e o Eufrates, hoje Iraque e terras vizinhas. Em grego Mesopotâmia significa entre os rios.

Há cerca de 10 000 anos, o povo desta região começou a viver da agricultura e da pecuária. Viviam em casas feitas de canas e tijolos de barro, construíram celeiros onde guardavam os seus cereais e desenvolveram um sistema simbólico, com pequenos objetos em argila, com diferentes formas geométricas, para fazerem o registro do seu comércio e dos seus bens. Por exemplo, um cilindro de argila podia representar um animal, duas esferas dois bushel (medida de capacidade) de cereal.

Entre 3500 e 3300 a.C. deu-se um grande desenvolvimento dos povos do sul da Mesopotâmia, em especial os da cidade de Ur e Uruk. Nesse período o comércio entre as populações intensificou-se e o sistema de registro se desenvolveu, começaram a aparecer objetos com formas mais diversificadas e com riscos e buracos.

Os babilônicos também desenvolveram muita matemática, inclusive já sabiam calcular a raiz quadrada de um número positivo, conhecido como método babilônico da raiz quadrada, que consiste no seguinte algoritmo:

$\sqrt{a} = n$, vamos arriscar um valor para \sqrt{a} , que chamaremos de x_1 , vamos dividir a por x_1 e somarmos o resultado a x_1 e depois dividir por 2, o resultado final chamaremos de x_2 , ou seja,

$$x_2 = \frac{1}{2}\left(x_1 + \frac{a}{x_1}\right).$$

Agora repetiremos o mesmo raciocínio usando o x_2 e chamaremos o resultado final de x_3 , então teremos,

$$x_3 = \frac{1}{2}\left(x_2 + \frac{a}{x_2}\right).$$

E assim sucessivamente, teremos uma relação de recorrência da seguinte forma:

$$x_n = \frac{1}{2} \left(x_{n-1} + \frac{a}{x_{n-1}} \right).$$

Agora precisamos definir um critério de parada, e esse critério se dá a partir do módulo da diferença entre o valor de a com o valor de x_n ao quadrado, que podemos majorar de acordo com a precisão que queremos para a raiz quadrada, ou seja,

$$|a - (x_n)^2| < \epsilon,$$

onde ϵ pode ser definido como, $\epsilon = 10^{-k}$, com $k \in \mathbb{N}$.

Portanto, de acordo com o valor definido para o algoritmo parar, x_n será o valor aproximado para \sqrt{a} , com k casas decimais iguais ao valor procurado.

Usando os conhecimentos babilônicos, construa um algoritmo no VisuAlg que imprima um valor aproximado para \sqrt{a} com uma precisão de k casas decimais.

Algoritmo sugerido: Raiz quadrada de um número usando o método babilônico

var

a, k, x: real

início

escreval (" Digite o valor que será calculado a raiz quadrada, a quantidade k de casas decimais de precisão desejada e um chute inicial")

leia (a,k,x)

enquanto abs (a - quad(x)) > exp (10,-k) faça

$x \leftarrow (x + a/x)/2$

fimenquanto

escreval (" A raiz quadrada de ",a," é igual ou aproximadamente: ",x)

fimalgoritmo

4 Calculando o valor de π

Calculando o valor aproximado de π (pi), usando o perímetro de polígonos regulares, com número de lados igual a uma potência de 2, inscritos em uma circunferência unitária.

Os matemáticos antigos, até o século XVI, tentaram obter valores de π (pi) usando polígonos regulares inscritos na circunferência com número de lados cada vez maior. Vamos mostrar como faziam isto.

A ideia se baseava em tomar um polígono regular inscrito em uma circunferência de raio 1, e ir dobrando o número de lados, de modo que o perímetro do polígono se aproxime mais e mais ao perímetro da circunferência à medida que o número de lados vai dobrando.

Essa ideia é puramente algorítmica, então podemos construir um algoritmo e depois implementá-lo no VisuAlg.

Na figura A.1 a seguir temos $l_n = AC = CE$ são lados do polígono regular de n lados inscrito em uma circunferência de raio 1. Se B é o ponto médio do arco AC , então $AB = BC = CD = DE = l_{2n}$ são lados do polígono regular de $2n$ lados inscrito na mesma circunferência.

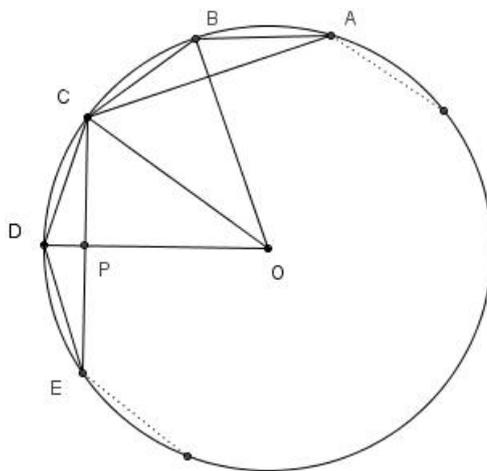


Figura A.1:

Sendo O o centro da circunferência, OD o raio que vale 1 e P o ponto de interseção de CE com OD fazendo um ângulo reto, temos os triângulos

retângulos CPO e CPD . Assim pelo triângulo CPO temos

$$(OP)^2 + \left(\frac{l_n}{2}\right)^2 = 1.$$

$$OP = \sqrt{1 - \frac{(l_n)^2}{4}} = \frac{1}{2}\sqrt{4 - (l_n)^2}.$$

E pelo triângulo CPD obtemos

$$(1 - OP)^2 + \left(\frac{l_n}{2}\right)^2 = (l_{2n})^2.$$

Daí,

$$(l_{2n})^2 = 2\left(1 - \frac{1}{2}\sqrt{4 - (l_n)^2}\right)^2 = 2 - \sqrt{4 - (l_n)^2}.$$

Assim,

$$l_{2n} = \sqrt{2 - \sqrt{4 - (l_n)^2}}.$$

Esta fórmula permite calcular o lado de um polígono regular de $2n$ lados inscrito em uma circunferência de raio 1 em função do lado do polígono regular de n lados inscrito na mesma circunferência. Como o lado do quadrado inscrito na circunferência de raio 1 é $l_4 = \sqrt{2}$ podemos facilmente prosseguir e encontrar:

$l_8 = \sqrt{2 - \sqrt{2}}$, $l_{16} = \sqrt{2 - \sqrt{2 + \sqrt{2}}}$, $l_{32} = \sqrt{2 - \sqrt{2 + \sqrt{2 + \sqrt{2}}}}$
e assim por diante.

Repare que $32 = 2^5$ e a expressão que calcula o lado do polígono de 32 lados possui 4 radicais. Lá dentro o primeiro sinal é negativo e todos os outros são positivos. Como cada vez que dobramos o número de lados acrescentamos mais um radical, o lado de um polígono regular de 2^n lados é:

$$l_{2^n} = \sqrt{2 - \sqrt{2 + \sqrt{2 + \sqrt{2 + \dots}}}},$$

com $(n - 1)$ radicais.

O perímetro do polígono de 2^n lados é igual a $2^n \cdot l_{2^n}$, que tende a 2π

quando n cresce. Assim, aproximações de π podem ser obtidas por:

$$p_n \approx 2^{n-1} \cdot l_{2^n} \approx 2^{n-1} \cdot \sqrt{2 - \sqrt{2 + \sqrt{2 + \sqrt{2 + \dots}}}},$$

com $(n - 1)$ radicais na expressão e um polígono de 2^n lados.

Infelizmente não podemos calcular por este método qualquer número de casas decimais de π . Seria possível se houvesse uma calculadora que trabalhasse com infinitas casas decimais. Mas esta calculadora ainda não foi inventada. Observe que, na expressão acima, temos um produto, onde o primeiro fator é muito grande e o segundo muito pequeno. Se a sua calculadora trabalha com, por exemplo, 12 dígitos, não é possível aumentar muito o valor de n . O segundo fator vai perdendo precisão e o resultado idem. Os matemáticos antigos calculavam essas raízes manualmente, com um número absurdo de casas decimais, para conseguirem obter umas poucas casas decimais precisas de π .

Usando as ideias acima, construa um algoritmo no VisuAlg que imprima um valor aproximado de π .

Algoritmo sugerido: *Calculando um valor aproximado para π .*

```
var
m, p, r: real
n, k: inteiro
inicio
  escreval ("Digite n > 1 para aproximar o valor de pi, usando um polígono
regular de 2n lados inscrito em uma circunferência de raio 1 ")
  leia (n)
  r ← 0
  para k de 1 ate n-2 faça
    r ← raizq(2+r)
  fimpara
  m ← exp (2,n-1)
  p ← m*raizq(2 - r)
  escreval ("O valor aproximado de pi encontrado pelo algoritmo é ", p)
  escreval (" ")
  escreval ("O valor aproximado de pi dado pelo programa é ",pi)
finalgoritmo
```