

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

Ciência da Computação para Crianças

por

CLÁUDIA SANTOS FERNANDES

Dissertação submetida à avaliação,
como requisito parcial para a obtenção do grau de Mestre
em Ciência da Computação

Prof. Paulo Fernando Blauth Menezes
Orientador

Porto Alegre, dezembro de 2002.

CIP - CATALOGAÇÃO NA PUBLICAÇÃO

Fernandes, Cláudia Santos

Ciência da Computação para Crianças / por Cláudia Santos
Fernandes. - Porto Alegre : PPGC da UFRGS, 2002.

156 f.: il.

Dissertação (mestrado)-Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação, Porto Alegre, BR - RS, 2001. Orientador: Menezes, Paulo Fernando Blauth.

1. Teoria da Computação. 2. Informática Educacional. 3. Metodologia do Ensino de Ciência da Computação. I. Menezes, Paulo Fernando Blauth. II. Título.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitora: Profa. Wrana Panizzi

Pró-Reitor de Ensino: Prof. José Carlos Ferraz Hennemann

Pró-Reitor Adjunto de Pós-Graduação: Prof. Jaime Evaldo Fensterseifer

Diretor do Instituto de Informática: Prof. Philippe Olivier Alexandre Navaux

Coordenador do PPGC: Prof. Carlos Alberto Heuser

Bibliotecária-Chefe do Instituto de Informática: Beatriz Regina Bastos Haro

Agradecimentos

A todas as crianças que participaram tanto da pesquisa quanto da aplicação, por proporcionarem reflexão e alegria constantes.

Aos professores entrevistados, que forneceram subsídios fundamentais para o prosseguimento deste trabalho.

Às escolas pesquisadas, que cederam seu tempo sem interesse algum.

Ao Colégio de Aplicação da UFRGS por ser um centro de referência educacional no país e por nos acolherem tão bem, em especial aos professores Ítalo Modesto Dutra, Teresa Alice Rossell Malinsky, Adalberto Breier, Érica Becker e Maria Carmem de Moraes Gomes.

Um agradecimento especial ao professor Blauth pela orientação. Devo-lhe não só minha gratidão por sua vital contribuição, sem nunca vociferar ou tentado se impor, mas também por sua capacidade em me fazer progredir. Muito obrigada!

Ao excelente grupo de Teoria da Computação, pelo apoio e sugestões, especialmente à Simone Bavaresco e ao José Francisco Szucs, que conduziram a segunda aplicação e fizeram desta pesquisa também a sua.

À psicóloga Regina Gioconda, pelas orientações, sugestões e interpretação da pesquisa com as crianças.

Aos professores e funcionários do Instituto de Informática da Universidade Federal do Rio Grande do Sul, que sempre me auxiliaram, especialmente ao professor Tiarajú Asmuz Divério, pela atenção dispensada sempre com uma “pitada” de humor.

Aos meus companheiros de viagem Liliane, Daniela e Francisco, por compartilharem suas tristezas e alegrias que me incentivaram a prosseguir. Amigos, eu consegui!

Aos alunos, professores e funcionários da Faculdade de Informática de Presidente Prudente, da Universidade do Oeste Paulista pelo apoio fundamental, especialmente à profa. Melissa Palone Zanata pelas “dicas” de viagem.

Aos meus pais Maria Rosa e Jessé, não só por dividirem comigo minhas responsabilidades de mãe, mas por suas orgulhosas exibições de minhas conquistas. Bem, aqui vai mais uma!

Aos meus irmãos, sobrinhos e cunhados, por toda ajuda que obtive.

Ao Renê, que com seu carinho e apoio deu-me a força necessária para eu prosseguir.

Aos meus filhos Diego e Camila, que são a razão do meu viver e que várias vezes compreenderam a minha ausência, é para eles que ofereço este trabalho.

Sumário

Lista de abreviaturas.....	07
Lista de figuras.....	08
Lista de tabelas.....	10
Resumo.....	11
Abstract.....	12
1. Introdução.....	13
1.1 Contexto e motivação.....	13
1.2 Objetivos do estudo.....	14
1.3 Metodologia.....	14
1.3.1 Pesquisas de campo.....	14
1.3.2 Estudo de alguns dos conceitos fundamentais da Ciência da Computação selecionados.....	14
1.3.3 Elaboração da teoria e exercícios.....	15
1.3.4 Aplicação dos conceitos e exercícios com crianças.....	15
1.3.5 Alterações e correções na metodologia.....	15
1.4 Principal contribuição.....	15
1.5 Organização do estudo.....	16
2 O que é feito.....	17
2.1 Pesquisas em escolas.....	17
2.1.1 Detalhamento das pesquisas em escolas.....	17
2.1.2 Considerações sobre pesquisas em escolas.....	19
2.2 Pesquisas em escolas do Brasil.....	20
2.3 Considerações.....	20
3 O que deve ser ensinado de Ciência da Computação para crianças.....	21
3.1 Pesquisas com professores.....	21
3.2 Considerações da pesquisa com professores.....	25
3.3 Pesquisa com crianças.....	26
3.4 Considerações da pesquisa com crianças.....	27
3.5 Considerações.....	27
4 Metodologia do ensino de Ciência da Computação para crianças.....	28
4.1 Algoritmos.....	29
4.1.1 Lógica.....	29
4.1.2 Álgebra.....	30
4.1.3 Representação de algoritmos.....	34
4.2 Programas.....	38
4.2.1 Identificadores.....	39
4.2.2 Comandos	40
4.3 Máquinas.....	44
4.3.1 Máquina de Turing.....	44
4.4 Máquina de estados finitos ou autômatos finitos.....	49
4.5 Computabilidade e complexidade de problemas.....	51
5 Aplicação da metodologia.....	58
5.1 Pré-teste.....	58
5.1.1. Primeiro e segundo encontros.....	59

5.1.2. Terceiro encontro.....	60
5.1.3. Quarto encontro.....	61
5.1.4. Quinto encontro.....	63
5.1.5. Sexto e sétimo encontros.....	63
5.1.6. Oitavo encontro.....	64
5.1.7. Nono encontro.....	65
5.1.8. Décimo e décimo primeiro encontros.....	66
5.1.9. Décimo segundo e décimo terceiro encontros.....	68
5.2 Aplicação da metodologia ao Colégio de Aplicação da UFRGS.....	70
5.2.1 O Colégio de Aplicação da UFRGS e o Projeto Amora.....	70
5.2.1.1 Histórico do Colégio de Aplicação da UFRGS.....	70
5.2.1.2 Apresentação do Projeto Amora.....	71
5.2.2 Detalhamento das aulas da oficina de Ciência da Computação ao Colégio de Aplicação da UFRGS.....	72
5.2.2.1 Primeira aula.....	72
5.2.2.1 Segunda e terceira aulas.....	73
5.3 Considerações.....	75
6 Conclusão e trabalhos futuros.....	76
6.1 Conclusões.....	76
6.3 Principais contribuições e produção científica.....	77
6.2 Trabalhos futuros.....	78
Anexo 1 Ficha de pesquisa com escolas.....	79
Anexo 2 Ficha de pesquisa com os professores.....	83
Anexo 3 Ficha de pesquisa com as crianças.....	84
Anexo 4 Exercícios de lógica e aritmética (autores desconhecidos).....	85
Anexo 5 Exercícios de lógica símbolo-imagem (Piaget em [FUR 72]).....	86
Anexo 6 Exemplos de algoritmos e primitivas, para construção de um pássaro [BRO 2000] e de um balão em dobradura.....	97
Anexo 7 Exercícios para construção de algoritmos.....	100
Anexo 8 Exercícios para construção de primitivas e de algoritmos.....	101
Anexo 9 Exemplo de algoritmo para criação de um papel de carta na ferramenta Microsoft Word.....	102
Anexo 10 Software Matemático: Teoria dos Conjuntos.....	104
Anexo 11 Exercícios para criação de Identificadores.....	105
Anexo 12 Exercícios utilizando o comando de atribuição.....	106
Anexo 13 Exemplos e exercícios para desenvolvimento de programas.....	107
Anexo 14 Simulação da Máquina de Turing para duplicação de símbolos.....	109
Anexo 15 Exercícios sobre Máquina de Turing.....	116
Anexo 16 Produtos de software para simulação da Máquina de Turing.....	117

Anexo 17	Exercícios para construção de autômatos.....	118
Anexo 18	Exemplo de problemas solucionáveis/não solucionáveis.	120
Anexo 19	Telas dos jogos Simfarm e Show do Milhão.....	122
Anexo 20	Telas do Software de Treinamento Básico Sobre o Windows Utilizando o Jogo de Representação-RPG [FER 98].....	123
Anexo 21	Detalhamento dos Encontros.....	124
Anexo 22	Telas do software Microsoft Paint e do site www.altavista.com.br.....	125
Anexo 23	Especificação das aulas na Oficina do Colégio de Aplicação.....	126
Anexo 24	Produtos de software abordando lógica.....	131
Anexo 25	Software construído para o exercício 1 do anexo 7.....	133
Anexo 26	Artigo ICECE´2000.....	134
Anexo 27	Artigo SITE´2001 – AACE.....	140
Anexo 28	Artigo SBC´2001 – WIE.....	142
	Bibliografia.....	152

Lista de Abreviaturas

AACE	Association for the Advancement o Computing in Education
BD	Banco de Dados
CAP	Colégio de Aplicação
Cnpq	Conselho Nacional de Desenvolvimento Científico e Tecnológico
EducaDi	Projeto Educação a Distância
fig.	Figura
IEEE	Institute of Electrical and Electronics Engineers
ICECE	International Conference on Engineering and Computer Education
MEC	Ministério da Educação e Cultura
NTIC	Novas Tecnologias de Informação e Comunicação
PIBIC	Programa Institucional de Bolsas de Iniciação Científica
RPG	Role Playing Game
SBC	Sociedade Brasileira de Computação
SITE	Society for Information Technology & Teacher Education
tab.	Tabela
UFRGS	Universidade do Rio Grande do Sul
UNOESTE	Universidade do Oeste Paulista
WIE	Workshop de Informática na Escola
WWW	World Wide Web

Lista de Figuras

FIGURA 2.1 - Distribuição da quantidade de alunos nas escolas.....	17
FIGURA 2.2 - Distribuição das séries abrangidas pelas escolas pesquisadas.....	18
FIGURA 2.3 - Recursos utilizados pelas escolas na informática educacional.....	18
FIGURA 2.4 - Quantidade de professores responsáveis pelas aulas de informática..	18
FIGURA 2.5 - Formação dos professores responsáveis pelas aulas de informática...	19
FIGURA 4.1 - Exercícios sobre lógica e aritmética.....	30
FIGURA 4.2 - Exemplos de exercícios sobre lógica em [SER 2000].....	31
FIGURA 4.3 - Exemplos de exercícios de lógica em [SER 2000].....	32
FIGURA 4.4 - Exercícios de lógica símbolo-imagem, Piaget em [FUR 72].....	33
FIGURA 4.5 - Exemplo de primitiva para construção de um balão de dobradura....	34
FIGURA 4.6 - Exemplo de algoritmo para transformar-se um papel com formato de um retângulo em um papel com formato de um quadrado.....	34
FIGURA 4.7 - Exemplo de algoritmo, para construir-se um balão de dobradura.....	35
FIGURA 4.8 - Exercício para construção de algoritmo.....	36
FIGURA 4.9 - Exercício para construção das primitivas e algoritmo para o problema da "Torre de Hanói"	36
FIGURA 4.10 - Algoritmo, para se construir um papel de carta personalizado, por meio da ferramenta "Word".....	37
FIGURA 4.11 - Tela inicial do " <i>Software para Ensino Fundamental: Teoria dos Conjuntos</i> ".....	39
FIGURA 4.12 - Exemplo de um identificador e suas 3 características.....	39
FIGURA 4.13 - Exercício para definição de identificador.....	40
FIGURA 4.14 - Exercícios utilizando o comando de atribuição.....	41
FIGURA 4.15 - Representação da estrutura de seleção.....	42
FIGURA 4.16 - Exemplo da utilização do operador relacional \geq e estrutura de seleção.....	42
FIGURA 4.17 - Exemplo da utilização do operador lógico e e estrutura de seleção..	43
FIGURA 4.18 - Representação da estrutura de repetição.....	43
FIGURA 4.19 - Exercício utilizando estruturas de seleção.....	43
FIGURA 4.20 - Exercício utilizando estrutura de repetição.....	44
FIGURA 4.21 - Componentes de uma máquina de Turing.....	45
FIGURA 4.22 - Exemplo de simulação da máquina de Turing com a configuração inicial.....	46
FIGURA 4.23 - Exemplo de simulação da máquina de Turing.....	47
FIGURA 4.24 - Exemplo de simulação da máquina de Turing com a configuração final.....	48
FIGURA 4.25 - Exercício utilizando a máquina de Turing.....	49
FIGURA 4.26 - Tela do software que simula a máquina de Turing.....	49
FIGURA 4.27 - Representação dos estados.....	50
FIGURA 4.28 - Representação da função programa através de um grafo.....	50
FIGURA 4.29 - Exemplo de exercício e solução de autômatos.....	51
FIGURA 4.30- Relação entre classes de problemas [DIV 99], p. 169.....	52
FIGURA 4.31 - Exercício sobre computabilidade " <i>Problema do assassinato do bilionário Count Van Diamond</i> ".....	53
FIGURA 4.32 - Exercício sobre computabilidade " <i>Problema da ligação de serviços em 3 casas</i> ".....	53
FIGURA 4.33 - Exercício sobre computabilidade " <i>Problema do desenho da casa</i> "...54	54

FIGURA 4.34 - Exercício sobre computabilidade.....	54
FIGURA 4.35 - Classificação de problemas.....	55
FIGURA 4.36 - Problema do Caixeiro Viajante em [BRO 2000], p. 439.....	55
FIGURA 4.37 - Exercício sobre computabilidade " <i>Problema da Mochila</i> ".....	56
FIGURA 4.38 - Classificação de problemas.....	57
FIGURA 5.1- Tela Inicial do Software de Treinamento Básico sobre o Windows utilizando o Jogo de Representações (RPG).....	59
FIGURA 5.2 - Resultado do 3º encontro realizado com as crianças.....	61
FIGURA 5.3 - Resolução pelas crianças do exercício 6 do anexo 4.....	62
FIGURA 5.4 - Transcrição dos exercícios finais do anexo 5 resolvidos pelas crianças.....	64
FIGURA 5.5 - Algoritmo produzido por EMS e GT para construção de um copo em dobradura.....	65
FIGURA 5.6 - Um dos algoritmos produzidos pelas crianças para o exercício 1 do anexo 7.....	66
FIGURA 5.7 - Um dos algoritmos produzidos pelas crianças para o exercício 2 do anexo 7.....	67
FIGURA 5.8 - Um dos algoritmos produzidos pelas crianças para o exercício 3 do anexo 7.....	67
FIGURA 5.9 - Resolução pelas crianças do exercício 1 do anexo 8.....	68
FIGURA 5.10 - Resolução pelas crianças do exercício 2 do anexo 8.....	68
FIGURA 5.11 - Primitiva construída pela 1ª criança para o exercício 3 do anexo 8..	69
FIGURA 5.12 - Primitiva construída pela 2ª criança para o exercício 3 do anexo 8..	69
FIGURA 5.13 - Primitiva construída pela 3ª criança para o exercício 3 do anexo 8..	69
FIGURA 5.14 - Algoritmo para o exercício 3 do anexo 8.....	69
FIGURA 5.15 - Software de lógica o exercício 2 do anexo 7.....	74
FIGURA 5.16 - Software de lógica para o exercício 2 do anexo 8.....	74
FIGURA 5.17 - Software para construção de algoritmo - exercício 1 do anexo 7.....	74

Lista de Tabelas

TABELA 1.1 - Pesquisa com professores, contendo as disciplinas em que atuam e os conceitos fundamentais.....	21
TABELA 1.2 - Conceitos identificados pelos professores e organizados pela quantidade de respostas.....	24
TABELA 4.1 - Representação dos operadores relacionais, seu significado e exemplos de uso.....	42
TABELA 4.2 - Operadores lógicos, representação e uso.....	42
TABELA 4.3 - Lista de regras contendo o código da regra e sua descrição.....	45
TABELA 5.1 - Primitivas identificadas pelas crianças para o anexo 9.....	66
TABELA 5.2 - Respostas sobre o que gostariam de aprender do computador e de Ciência da Computação.....	73

Resumo

Este trabalho apresenta uma proposta metodológica para o ensino de Ciência da Computação para Crianças, elaborada, a partir de pesquisas e estudos, tendo, como objetivo, a aplicação dessa metodologia para correção e testes.

Devido à disseminação dos computadores e de seu uso como parte da formação dos alunos, muito se tem discutido sobre a união entre os recursos tecnológicos e a educação. A inclusão da informática no processo educativo vem sendo direcionada para a utilização e classificação de produtos de software educacional, em técnicas de comunicação à distância, no uso de recursos de multimídia ou de realidade virtual para apoio ao conteúdo abordado pelo professor e, também, para a construção de ambientes computacionais, que possam proporcionar uma mudança de paradigma educacional.

No entanto não adianta tornar disponível toda essa tecnologia, se, rapidamente, são abandonadas e são substituídas ferramentas e técnicas dentro da Computação. Por isto, tão importante quanto o ensino das tecnologias correntes é o ensino dos conceitos fundamentais da Ciência da Computação. Esses conceitos, além de proporcionarem um embasamento teórico para entendimento da ciência envolvida na computação, também propiciam o desenvolvimento de um raciocínio lógico e formal, assim como de habilidades que são exigidas no mundo atual.

Este trabalho apresenta pesquisas de campo sobre o uso da Informática Educativa em algumas escolas, a identificação, através de professores da graduação e pós-graduação, de alguns dos conceitos fundamentais da Ciência da Computação e uma pesquisa sobre o que crianças gostariam de aprender sobre Ciência da Computação. A partir dos resultados, são elaboradas teorias, demonstrações e exercícios para o ensino destes para crianças. Essa metodologia é aplicada a duas turmas heterogêneas de crianças para sua validação. Pretende-se que o estudo desses conceitos aguace o senso crítico e capacite as crianças não só a usarem as tecnologias mas também a entenderem seu funcionamento.

Palavras-chave: Teoria da Computação, Informática Educativa, Metodologia do Ensino de Ciência da Computação para Crianças.

TITLE: "COMPUTER SCIENCE FOR CHILDREN"**Abstract**

This work presents a methodological proposition for teaching Computer Science for Children. This proposition was elaborated from researches and studies, aiming objective the application of this methodology for correction and tests.

Due to the dissemination of computers and their use as part of students' formation, a lot of discussion about association between technology recourses and education has taken place. The inclusion of new technologies in teaching procedures has been aimed to use and classification of educational software products in distance communication techniques, in the use of multimedia and virtual reality resources to aid teaching, and also in the construction of computer environments that can change the educational paradigm.

However, it's not enough to make available all technology if tools and techniques are quickly abandoned and substituted in computers. Therefore, more important than teaching current technologies is teaching Computer Science fundamental concepts. These concepts besides providing an adequate theoretic basis needed to understanding the science involved in computers, also, provide the development of a logical and formal reasoning and develop skills required nowadays.

This work presents researches about the use of Educational Computers in schools, the identification, through college teachers, of the fundamental concepts of the Computer Science and a research about what children would like to learn about Computer Science. From the results theories and exercises are elaborated and provide subsidy for teaching fundamental concepts of Computer Science to children. This methodology is applied to two heterogeneous groups of children for validation. It is intended that the teaching of these concepts sharpens the critical sense and capacitate students to learn not only to use the technologies, but to understand how they work.

Keywords: Theory of Computation, Educational Computer, Methodology of Teaching Computer Science for Children.

1 Introdução

Este capítulo introdutório está dividido em 5 seções, que apresentam o contexto e motivação que o direcionaram, os objetivos geral e específicos propostos, a metodologia utilizada, sua principal contribuição e como foi organizado.

1.1 Contexto e motivação

Levando em consideração o uso disseminado de computadores e de todas as tecnologias envolvidas, a escola, como parte da formação total do indivíduo, está procurando utilizar as novas tecnologias para auxiliar a melhorar o ensino.

Diversos esforços são desenvolvidos, no sentido de se utilizar essa tecnologia, com o intuito de aplicá-la no processo de ensino-aprendizagem. Esses esforços estão direcionados para:

- utilização e classificação de produtos de software educacional em [MIS 2002], [PAN 97], [PAL 99] e [FER 98];
- técnicas de comunicação à distância em [ROS 2000] e [PEQ 2000];
- uso de recursos de multimídia ou de realidade virtual, para apoio ao conteúdo abordado pelo professor, como em [CRIS 98], [AND 99] e [OLI 2000];
- e, também, construção de ambientes computacionais, que possam proporcionar uma mudança de paradigma educacional em [SID 98] e [MIS 2002].

Porém, até o momento não se tem conhecimento de trabalhos que abordem os conceitos básicos envolvidos na Ciência da Computação. Todavia o mais próximo a que se chegou foi a descoberta de trabalhos envolvendo o ensino de linguagens de programação para crianças, [VIC 88].

Segundo [DIV 2001] e [CAR 94], as pessoas conseguem ler, interpretar e responder questões, contudo há dificuldade na transcrição do pensamento para a lógica formal.

De acordo com [DIV 99], a Teoria da computação é de fundamental importância para a Ciência da Computação, pois proporciona um adequado embasamento teórico, necessário para um correto e amplo entendimento da ciência envolvida na computação, que é a modelagem e computação corretas; igualmente propicia o desenvolvimento de um raciocínio lógico e formal e, ainda, introduz conceitos fundamentais, que são desenvolvidos em outras áreas.

Dentre outros aspectos envolvidos no processo de aprendizagem, o desenvolvimento do raciocínio lógico é de fundamental importância. Dessa feita, propõe-se que o ensino dos conceitos fundamentais da Ciência da Computação possa melhorar o aprendizado escolar e desenvolva habilidades que são exigidas no mundo atual.

Pesquisas voltadas para a área educacional por pessoas da área de Ciência da Computação são importantes, devido a dois panoramas: a nova geração de crianças que surge já incorporou o uso das tecnologias correntes e essas pesquisas servirão de base e referência, para que essa geração relacione conceitos entre a Computação e outras ciências, como a Filosofia, as Ciências Cognitivas e, principalmente, a Matemática.

A motivação para esse trabalho está na convicção de que o ensino dos princípios básicos das áreas de Ciência da Computação é tão ou mais importante que o ensino das tecnologias correntes.

1.2 Objetivos do estudo

1.2.1 Geral

O objetivo deste trabalho visa à elaboração de uma metodologia para o ensino de Ciência da Computação para Crianças, com aplicação dessa metodologia para testes e correções, sem a pretensão de um curso completo, mas com perspectivas de formação do pensamento instigante sobre problemas computacionais.

1.2.2 Específicos

- Construir uma proposta de currículo, que aborde alguns dos conceitos fundamentais da Ciência da Computação.
- Identificar uma faixa etária adequada para o trabalho com Ciência da Computação.
- Produzir a teoria e os exercícios específicos para cada um dos conceitos fundamentais selecionados.
- Produzir a teoria e os exercícios, de conformidade com alguns itens de algumas das atuais disciplinas.
- Verificar se existe a possibilidade de aplicação da metodologia em ambientes reais de aprendizagem.
- Aplicar a teoria e os exercícios, a fim de propiciar um trabalho interdisciplinar em um ambiente cooperativo e reflexivo.
- Verificar, através da aplicação da metodologia, se as crianças da faixa etária desta pesquisa conseguem expressar-se formalmente, compreender o comportamento de problemas e modelar, corretamente, as soluções para esses problemas.

1.3 Metodologia

Para se atingirem os objetivos propostos, diversas tarefas foram previstas. Em uma ordem cronológica aproximada, essas etapas são:

1.3.1 Pesquisas de campo

- Em escolas e na internet, para se saber o que está sendo feito no ensino de Ciência da Computação.
- Com professores dos cursos de Ciência da Computação e Sistemas de Informação, para identificação dos conceitos fundamentais da Ciência da Computação.
- Com crianças, a fim de se conhecerem as suas expectativas em relação ao aprendizado em Ciência da Computação.

1.3.2 Estudo de alguns dos conceitos fundamentais da Ciência da Computação selecionados

- Através de levantamento bibliográfico e seleção dessa bibliografia sobre os conceitos de Ciência da Computação, identificados na pesquisa com professores.

1.3.3 Elaboração da teoria e exercícios

- Adaptação dos conceitos para uma linguagem mais simples, clara e elaboração de demonstrações e exercícios, utilizando-se fatos comuns do cotidiano.
- Verificação da relação dos conceitos com os conteúdos de algumas das atuais disciplinas da grade curricular.

1.3.4 Aplicação dos conceitos e exercícios com crianças

- Através da seleção das crianças e da observação contínua do ambiente e das técnicas utilizadas na aplicação, modificando-o continuamente, visando, sempre, à motivação das crianças.

1.3.5 Alterações e correções na metodologia

- De acordo com os resultados esperados em cada exercício e as sugestões apresentadas pelas crianças.

1.4 Principal contribuição

Como principal contribuição desta pesquisa, tem-se o desenvolvimento de uma metodologia de ensino de Ciência da Computação para Crianças. Essa metodologia aborda alguns dos conceitos da Ciência da Computação, selecionados por profissionais atuantes na área do ensino, e foi elaborada de acordo com a faixa de idade sugerida por profissional de psicologia e identificada na pesquisa de campo.

No desenvolvimento da metodologia, procurou-se criticá-la, mediante aplicação em algumas crianças e submetê-la, por meio de artigos em congressos em que a informática educacional estivesse presente em [FER 2000], [FER 2000a] e [FER 2001]:

FERNANDES, C. S.; MENEZES, P. B.; ACCORSI, F. A Propose of Teaching Computer Science for Children. In: INTERNATIONAL CONFERENCE ON ENGINEERING AND COMPUTER EDUCATION, 2000. **Proceedings...** São Paulo: Senac, 2000.

FERNANDES, C. S.; MENEZES, P. B. Computer Science for Children. In: INTERNATIONAL CONFERENCE SOCIETY FOR INFORMATION TECHNOLOGY & TEACHER EDUCATION AND ASSOCIATION FOR THE ADVANCEMENT OF COMPUTING IN EDUCATION, 2001. **Proceedings...** Flórida: SITE, 2000.

FERNANDES, C. S.; MENEZES, P. B.; ACCORSI, F. Metodologia do Ensino de Ciência da Computação: uma proposta para crianças. In: WORKSHOP DE INFORMÁTICA NA ESCOLA - CONGRESSO DA SOCIEDADE BRASILEIRA DE COMPUTAÇÃO, 2001. **Proceedings...** Fortaleza: SBC, 2001.

Também foi iniciada a aplicação de parte dessa metodologia em um ambiente real de aprendizagem, para verificação de que estaria de acordo com os interesses e com o grau de abstração dessas crianças.

1.5 Organização do estudo

O desenvolvimento deste trabalho está apoiado na seguinte seqüência:

O capítulo 1 apresenta a introdução da dissertação com a definição dos objetivos da pesquisa, motivação, contribuição e metodologia utilizada.

No capítulo 2, são feitas pesquisas em escolas, como forma de se descobrirem aspectos da utilização da Informática no processo ensino-aprendizagem, dos recursos de hardware e software, da forma como as aulas são ministradas e como é feita a integração da informática com o contexto educacional dessas escolas. Também é detalhada uma pesquisa sobre a estrutura das escolas do Brasil.

No capítulo 3, são apresentadas outras duas pesquisas, que foram realizadas com o intuito de se encontrarem os conceitos fundamentais da Ciência da Computação. A pesquisa com professores foi realizada com profissionais dos cursos de Ciência da Computação e Sistemas de Informação, pertencentes a diversas e representativas instituições do país. A pesquisa com crianças visou a verificar suas expectativas em relação ao ensino de Ciência da Computação.

Após a identificação dos conceitos fundamentais, efetuada no capítulo 3, o capítulo 4 concentrou-se na elaboração da metodologia de ensino de Ciência da Computação para Crianças. Foram feitos estudos e propostos conceitos e exercícios nos itens algoritmos, programas, máquinas e autômatos, procurando, sempre, utilizar-se de uma linguagem simples e clara.

O capítulo 5 concentrou-se na aplicação da metodologia em duas turmas. A primeira aplicação ocorreu simultaneamente à elaboração da metodologia; e a segunda aplicação ocorreu em um ambiente real de aprendizagem.

O capítulo 6 conclui a dissertação, com apresentação de suas principais contribuições. Nele são feitas sugestões, para a continuação deste trabalho.

Por último, são anexados os modelos das pesquisas, os exercícios elaborados em sua totalidade, as telas dos softwares utilizados e os artigos publicados em congressos internacionais e nacional.

2 O que é feito

Para saber-se o que está sendo feito no ensino de Ciência da Computação e ampliar o cenário atual de Presidente Prudente-SP-Brasil, foi realizada uma pesquisa de campo em 5 escolas particulares das cidades de Presidente Prudente-SP e 4 escolas de Londrina-PR aplicando-se um questionário (anexo 1) aos responsáveis pela área de informática educacional. Foram pesquisados aspectos da utilização da Informática somente no processo ensino-aprendizagem, dos recursos de hardware e software, da forma como as aulas são ministradas e como é feita a integração da informática com o contexto educacional dessas escolas.

Apesar do valor de os equipamentos de informática terem caído muito nos últimos tempos, eles ainda não são acessíveis à maioria das escolas brasileiras. Por essa razão, a pesquisa concentrou-se em escolas particulares, que possuem condições de adquirir esses equipamentos e os utilizam para fins educacionais.

2.1 Pesquisas em escolas

As respostas dos questionários (anexo 1) foram preenchidas pela mestranda e validadas pelos profissionais responsáveis pela área de informática educacional de cada uma das escolas pesquisadas e estão detalhadas a seguir.

2.1.1 Detalhamento das pesquisas em escolas

A maioria das escolas são grandes, possuindo acima de 500 alunos. 33% das escolas pesquisadas possuem até 500 alunos; outros 33% possuem de 501 a 1.000 alunos; 11,11% possuem de 1.001 a 2.000 alunos; e 22,22% possuem acima de 1.000 alunos, como pode ser observado na fig. 2.1.

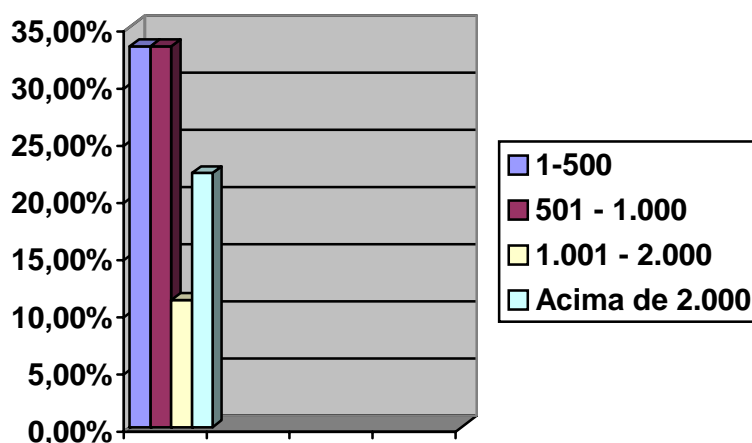


FIGURA 2.1 – Distribuição da quantidade de alunos nas escolas

A maioria das escolas pesquisadas atua desde o jardim da infância até os ensinos fundamental e médio: 44,44%, sendo que 22,22% atuam, também, nos cursos pré-vestibulares. Somente 33,33% atuam somente até o ensino fundamental, conforme a fig. 2.2.

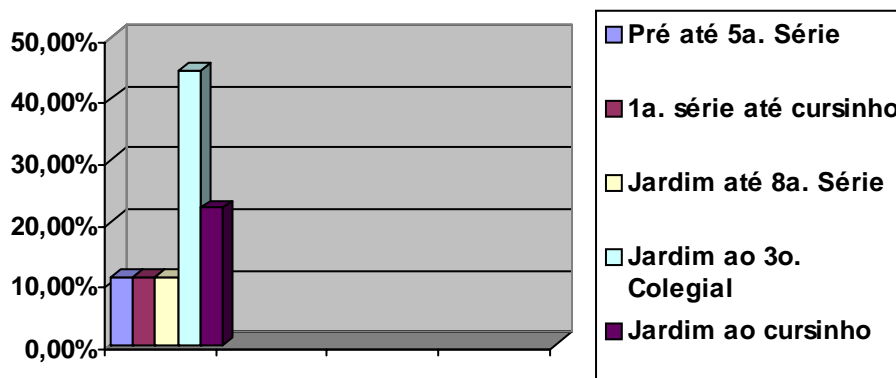


FIGURA 2.2 – Distribuição das séries abrangidas pelas escolas pesquisadas

Todas as escolas pesquisadas responderam que possuem informática educacional.

A fig. 2.3 apresenta os recursos de informática utilizados, que são computadores com recursos de multimídia, conectados à internet e a impressoras. Somente 11,11% possuem recursos de robótica, e nenhuma possui lousa eletrônica.

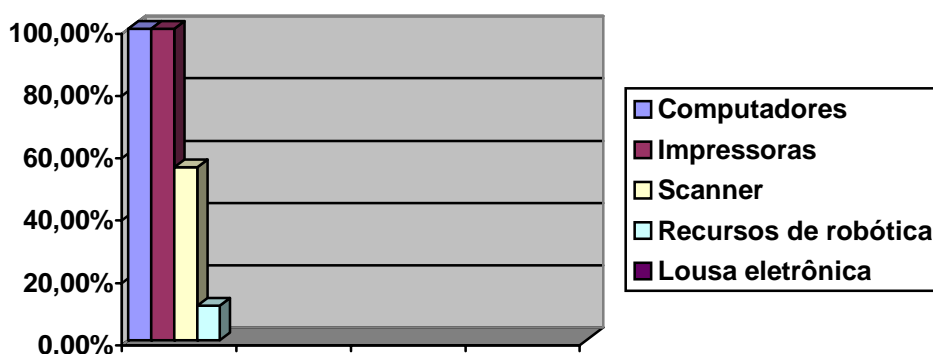


FIGURA 2.3 – Recursos utilizados pelas escolas na informática educacional

Em todas as escolas, os recursos de informática estão dispostos em linha ou em "L" e as salas onde se encontram estão separadas, em relação às salas onde são ministradas outras disciplinas. Porém, em uma das escolas, há um projeto de serem colocados equipamentos nas salas onde são ministradas as disciplinas tradicionais.

A maioria das escolas (66,66%) possui 2 professores, que são responsáveis por todas as aulas de informática. Somente 11,11% das escolas possuem 3 professores e 22,22% possuem somente 1 professor, conforme a fig. 2.4.

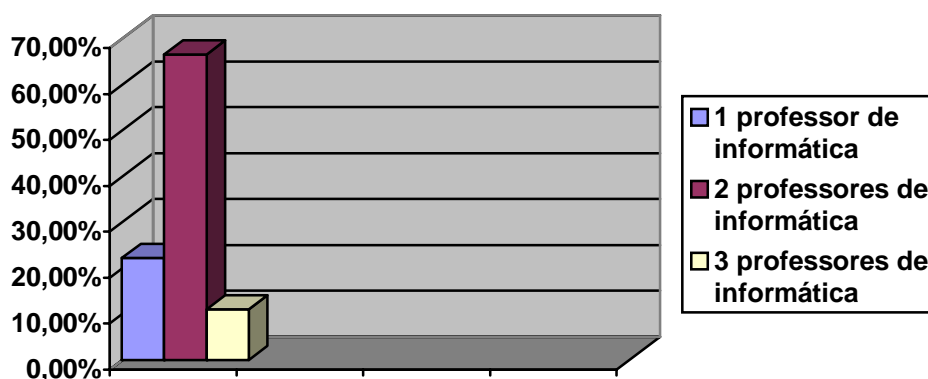


FIGURA 2.4 – Quantidade de professores responsáveis pelas aulas de informática

De acordo com a fig. 2.5, 44,44% dos professores responsáveis por coordenar as aulas de informática possuem, em sua formação, cursos de treinamento em informática. Outros 44,44% têm, como formação, Ciência da Computação ou Analista de Sistemas e 11,11% são formados em Matemática.

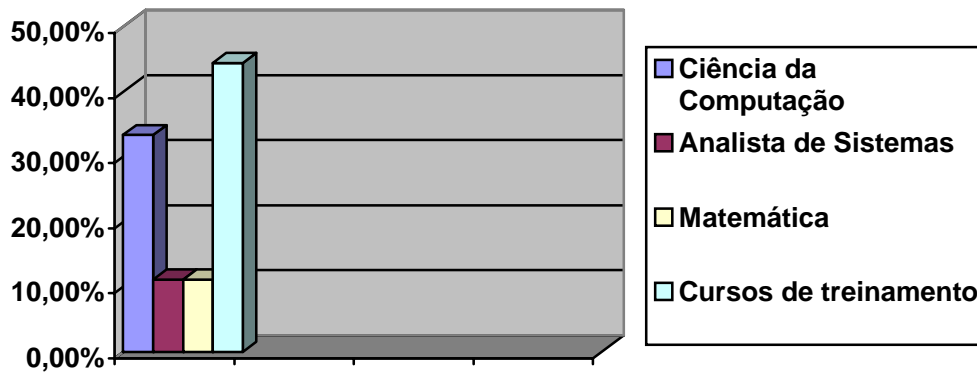


FIGURA 2.5 – Formação dos professores responsáveis pelas aulas de informática

A forma de integração com outras disciplinas ocorre, igualmente, para todas as escolas, ou seja, através de produtos de software educacionais específicos para cada uma das disciplinas. Porém, antes de adquirirem os softwares, são realizadas pesquisas e discussões entre os professores de informática e das outras disciplinas, conseguindo, desse modo, consenso na compra desses produtos.

2.1.2 Considerações sobre as pesquisas em escolas

Entre as escolas pesquisadas, 75% trabalham de modo convencional, possuem laboratórios de informática contendo vários recursos de informática, como: impressoras, scanner e computadores com recursos de multimídia, redes e internet, dispostos em linha.

Nesses laboratórios, as aulas são ministradas por profissionais de informática periodicamente, abrangendo quase todas as séries dos ensinos fundamental e médio, de acordo com o objetivo da escola.

As aulas são, sempre, separadas das outras disciplinas. Limitam-se a ensinar alguns aplicativos de uso geral, como editores de texto, planilhas eletrônicas e produtos para elaboração e apresentações multimídia. Em conjunto com os professores das disciplinas básicas, trabalham com produtos voltados para fins educacionais, abrangendo alguns tópicos das disciplinas da grade curricular.

Algumas escolas trabalham de forma um pouco distinta e são alvo de considerações complementares.

Uma das escolas somente utiliza os computadores tanto para os professores quanto para os alunos na criação de *Home Pages*. Foram feitos treinamentos com os professores, de modo que cada um constrói a *Home Page* de sua disciplina. Os alunos também criam páginas e o professor incorpora as melhores páginas na sua disciplina.

Outra escola visitada possui a idéia de instalar computadores em sala de aula, a fim de que o professor possa fazer uso durante suas aulas expositivas, utilizando, dessa feita, o computador como pesquisa e como apoio para ilustração e exemplos do conteúdo abrangido.

A última escola que trabalha diferente não possui treinamento básico, pois acredita que o aluno já deveria saber manusear o computador. O aluno se inscreve para as aulas e os computadores são utilizados como apoio para montagem de robôs, por

exemplo, um robô que se movimenta com energia solar. Há produtos de software para fins educacionais, ainda que, pouco utilizados. Quem seleciona esses produtos é o próprio profissional de informática. Segundo ele, esses programas são escolhidos, por se tratar de programas de simulação, onde é proposto um desafio para o aluno.

2.2 Pesquisas em escolas do Brasil

Em uma pesquisa realizada pelo Jornal “*Folha de São Paulo*” [INF 2000] sobre a estrutura das escolas do Brasil, foi verificado que, em 1997, apenas 28% das escolas possuíam laboratórios de informática, passando para 54% em 1999.

Esta mesma pesquisa aponta que as escolas de ensino médio estão mais bem equipadas do que as de educação fundamental; mas, ainda assim, apresentam números que revelam o nível brasileiro em tecnologia educacional: apenas 22% dos estabelecimentos do antigo 2º grau possuem acesso à Internet.

A pesquisa conclui que no nível de ensino médio, do qual, segundo as diretrizes curriculares do Ministério da Educação e Cultura [MEC 2002], o aluno já deveria sair preparado para ingressar no mercado de trabalho ou na vida acadêmica, poucas escolas estaduais (5%) e quase a metade das particulares (49%) oferecem aos estudantes a possibilidade de se conectarem à rede.

2.3 Considerações

De acordo com [JUR 97], os educadores podem desmistificar o uso de computadores e *alfabetizar, tecnologicamente*¹ os alunos; e, ao mesmo tempo, assegurar que as tecnologias da informação sejam usadas, tendo em vista desenvolver e aprofundar a aprendizagem, de forma que a *alfabetização em tecnologias da informação* aguace o senso crítico e capacite os alunos a aprenderem não só a usar essas tecnologias mas também a entender seu funcionamento.

Essa visão é corroborada por vários autores, como em [ESM 99] e [SIL 97], em que há a constatação de que os métodos tradicionais de ensino da Matemática levam o aluno a empregar muito tempo na elaboração de cálculos em detrimento da construção de conceitos e desenvolvimento do raciocínio abstrato.

Portanto, como, em matemática, não se ensina a manusear uma calculadora, o ensino de Ciência da Computação não pode limitar-se a um treinamento ou ensinar uma ferramenta como um editor de texto específico. De fato, se acontecer uma mudança do tipo de editor, esse treinamento terá de ser recomeçado. Adicionalmente, como não cabe nesse contexto, o ensino de datilografia ou taquigrafia.

¹ Conceito utilizado na Inglaterra, pelo qual fazem uma taxonomia com os vários níveis de alfabetização, por exemplo, tem menor índice de alfabetização a pessoa que não sabe utilizar o computador, para se comunicar com outras pessoas.

3 O que deve ser ensinado de Ciência da Computação para Crianças

Para se saber quais tópicos de Ciência da Computação ensinar a crianças, foi realizado um questionário (anexo 2) com professores de cursos de Ciência da Computação, Sistemas de Informação e Processamento de Dados, pertencentes às seguintes instituições: UFRGS (Universidade Federal do Rio Grande do Sul); UEM (Universidade Estadual de Maringá); USP-São Carlos (Universidade de São Paulo – Campus São Carlos); UNOESTE (Universidade do Oeste Paulista); UNIPAR (Universidade do Paraná); UFPel (Universidade Federal de Pelotas); e UNOPAR (Universidade Norte do Paraná).

3.1 Pesquisas com professores

Os professores participantes atuam em disciplinas diversas, como: Engenharia de Software, Banco de Dados, Linguagens de Programação, Matemática, Teoria Geral de Sistemas etc. Foi-lhes solicitado, que independentemente da aplicação, respondessem, de forma intuitiva e objetiva, quais os conceitos fundamentais da Ciência da Computação e estão dispostos na tab. 1.1, exatamente como foram respondidos.

TABELA 1.1 – Pesquisa com professores contendo as disciplinas que atuam e os conceitos fundamentais

Disciplinas em que Atuam	Conceitos identificados
<ul style="list-style-type: none"> – Fundamentos de Sistemas de Informação – Teoria Geral dos Sistemas – Prática e Gerência de Projetos – Tópicos em Sistemas de Informação I e II – Empreendedorismo 	<ul style="list-style-type: none"> – Pensamento Sistêmico – Amplo Conhecimento da Tecnologia da Informação – Facilidade de Comunicação e Trabalho em Equipe – Agente de Produção e Disseminação de Pesquisa em Tecnologia
<ul style="list-style-type: none"> – Arquitetura de Computadores – Circuitos Digitais – Lógica – Matemática Discreta 	<ul style="list-style-type: none"> – Arquitetura de Computadores – Engenharia de Software – Banco de Dados – Computação Gráfica
<ul style="list-style-type: none"> – Fundamentos BD – Projeto BD 	<ul style="list-style-type: none"> – Lógica e Álgebra – Teoria da computação, computabilidade – Linguagens de programação e algoritmos – Linguagens formais e compilação – Estrutura de dados, estruturas de arquivos, BD – Sistemas operacionais, sistemas distribuídos
<ul style="list-style-type: none"> – Algoritmos e Técnicas de Programação I e II – Estrutura de Dados – Ferramentas Computacionais I – Informática Básica 	<ul style="list-style-type: none"> – Lógica de desenvolvimento – Transmissão de dados – Estruturas de Dados (tipos e aplicações) – Armazenamento de Dados – Matemática computacional aplicada – Métodos de busca (acesso às informações)

Disciplinas em que Atuam	Conceitos identificados
<ul style="list-style-type: none"> - Teoria Geral de Sistemas - Introdução à Administração - Empreendedorismo - Organização e Métodos 	<ul style="list-style-type: none"> - Sistemas - Organizações - Algoritmos - Hardware/software - Linguagens de Programação - Dados/Informações
<ul style="list-style-type: none"> - Algoritmos e Técnicas de Programação I - Inglês Instrumental - Informática Básica 	<ul style="list-style-type: none"> - Informação - Formalismo - Lógica - Semântica - Batch e on-line - Transmissão de dados
<ul style="list-style-type: none"> - Laboratório I e II - Teoria dos Grafos - Estágio Supervisionado 	<ul style="list-style-type: none"> - Linguagens de Programação - Lógica computacional - Matemática
<ul style="list-style-type: none"> - Pesquisa e Ordenação - Estrutura de Dados - Programação de Computadores I 	<ul style="list-style-type: none"> - Algoritmos - Hierarquia de Chomsky - Máquinas Universais - Computabilidade - Funções Computadas - Linguagens Formais
<ul style="list-style-type: none"> - Linguagens Formais e Autômatos 	<ul style="list-style-type: none"> - Programa - Máquina - Gramática - Linguagem - Autômatos
<ul style="list-style-type: none"> - Linguagens e Técnicas de Programação - Algoritmos e Estrutura de Dados - Linguagens Formais e Autômatos 	<ul style="list-style-type: none"> - Linguagens formais (Geradores e Reconhedores) - Computabilidade/Algoritmos - Estrutura de Dados (Listas, Árvores e Grafos)
<ul style="list-style-type: none"> - Laboratório de Informática (Histórico, Conceitos de hardware, software, SO, Editor de textos) - Ferramentas Computacionais (Delphi) - Estágio Supervisionado 	<ul style="list-style-type: none"> - Conhecimento em Matemática - Conceitos de Lógica - Conceitos de Algoritmo (principal)
<ul style="list-style-type: none"> - Programação de Computadores II e IV - Ferramentas Computacionais I - TTC - Projeto e Estágio Supervisionado 	<ul style="list-style-type: none"> - Iniciação Científica - Projeto de Software não Comercial - Técnica de Programação - Arquitetura de Software e Hardware
<ul style="list-style-type: none"> - Banco de Dados 	<ul style="list-style-type: none"> - Gerenciamento de dados/informações
<ul style="list-style-type: none"> - Sistema distribuído - Modelagem de Empresas - Ensino à Distância 	<ul style="list-style-type: none"> - Algoritmos - Modelagem Conceitual - Abstração - Lógica Formal - Estrutura de Dados - Arquitetura de Sistemas

Disciplinas em que Atuam	Conceitos identificados
<ul style="list-style-type: none"> - Engenharia de Software - SGDB - Projeto de Graduação 	<ul style="list-style-type: none"> - Algoritmos - Estrutura de dados - Linguagens formalismos e Autômatos - Sistemas Centralizados e Distribuídos - Interface Homem-Máquina - Hiperdocumentos e Multimídia - Sistemas de Informação - Engenharia de Software
<ul style="list-style-type: none"> - Engenharia de Software - Linguagens de Programação 	<ul style="list-style-type: none"> - Algoritmos
<ul style="list-style-type: none"> - Banco de Dados - Redes de Computadores 	<ul style="list-style-type: none"> - Lógica - Algoritmos
<ul style="list-style-type: none"> - Programação III - Banco de Dados 	<ul style="list-style-type: none"> - Lógica de Programação - Conceitos Matemáticos - Sistema Operacional - Redes
<ul style="list-style-type: none"> - Computação Numérica - Teoria da Computação - Processamento de Alto Desempenho 	<ul style="list-style-type: none"> - Lógica e Estruturas Discretas - Teoria dos Algoritmos - Teoria da Computabilidade & Complexidade - Métodos Formais - Tecnologias da Computação
<ul style="list-style-type: none"> - Algoritmos e Técnicas de Programação - Estrutura de Dados - Compiladores 	<ul style="list-style-type: none"> - Algoritmos - Programas
<ul style="list-style-type: none"> - Engenharia de Software 	<ul style="list-style-type: none"> - Algoritmos - Abstração - Tipos Abstratos de Dados - Modelagem
<ul style="list-style-type: none"> - Computação Gráfica 	<ul style="list-style-type: none"> - Algoritmo - Estrutura de Dados - Tipo Abstrato de dados/objeto/Classe - Complexidade de espaço e tempo - Especificação X Implementação
<ul style="list-style-type: none"> - Tolerância a falhas - Arquitetura de Computadores - Sistemas Operacionais 	<ul style="list-style-type: none"> - Álgebra booleana - Aritmética binária - Modelo procedural - Concorrência - Processo, Thread, Objetos - Complexidade
<ul style="list-style-type: none"> - Pesquisa e Ordenação - Teoria da Computação - Linguagens Formais - Engenharia de Software 	<ul style="list-style-type: none"> - Algoritmos - Máquinas - Lógica - Estrutura de Dados
<ul style="list-style-type: none"> - Arquitetura e Organização de Computadores - Modelagem e Simulação 	<ul style="list-style-type: none"> - Modelo de Von Newmann (CPU, memória, programa, dado, instrução) - Modelos de linguagens de programação

Disciplinas em que Atuam	Conceitos identificados
	(procedural, funcional, lógico)
	– Estruturas de dados (listas, filas, pilhas, grafos)
	– Algoritmos (sequências, controle de fluxo, laços e iterações, recursividade)
	– Computabilidade
	– Linguagens formais e autômatos
– Informática Teórica	– Máquinas, autômatos
– Inteligência Artificial	– Programas
– Informática Educacional	– Computação
	– Computador
– Engenharia de Software	– Algoritmos
– Linguagens de programação	

A seguir, foram somados os conceitos, em relação à quantidade de respostas, e organizados na tabela 1.2. Alguns critérios foram adotados, como: o conceito identificado como *linguagens* foi somado com *linguagens de programação*; os conceitos *álgebra* e *aritmética binária* foram deixados juntos, com a *matemática*; o conceito identificado como *máquinas* foi somado ao de *autômatos*, assim como as lógicas *computacional*, de *desenvolvimento* e *algébrica*, que estão colocadas somente no conceito *lógica*. Porém a *lógica formal*, por se tratar de significado diferente foi separada das demais.

TABELA 1.2 – Conceitos identificados pelos professores e organizados pela quantidade de respostas.

	Conceitos	Quantidade de respostas
1	Algoritmos	15
2	Lógica	10
3	Estrutura de dados	7
4	Linguagens de programação	6
5	Autômatos e máquinas	6
6	Computabilidade	5
7	Matemática, álgebra e aritmética binária	5
8	Programas	5
9	Complexidade	4
10	Linguagens formais	4
11	Redes	3
12	Abstração	2
13	Banco de dados	2
14	Dados/Informações	2
15	Formalismos	2
16	Modelagem conceitual	2
17	Objetos	2
18	Sistemas distribuídos	2
19	Sistemas operacionais	2
20	Tipos de dados abstratos	2
21	Agente de produção	1
22	Ampla Conhecimento da tecnologia da informação	1
23	Armazenamento de dados	1
24	Arquitetura de computadores	1
25	Arquitetura de sistemas	1

	Conceitos	Quantidade de respostas
26	Arquitetura de software e hardware	1
27	Batch e on-line	1
28	Classes	1
29	Compilação	1
30	Computação	1
31	Computação Gráfica	1
32	Computador	1
33	Concorrência	1
34	Disseminação de pesquisa em tecnologia	1
35	Engenharia de software	1
36	Especificação	1
37	Estrutura de arquivos	1
38	Estruturas discretas	1
39	Facilidade de comunicação	1
40	Funções Computadas	1
41	Gerenciamento de Dados	1
42	Gramática	1
43	Hardware/software	1
44	Hierarquia de chomsky	1
45	Hiperdocumentos e multimídia	1
46	Implementação	1
47	Iniciação Científica	1
48	Interface homem-máquina	1
49	Matemática computacional aplicada	1
50	Métodos de busca	1
51	Métodos Formais	1
52	Modelo de Von Newmann	1
53	Modelo procedural	1
54	Organizações	1
55	Pensamento sistêmico	1
56	Processo	1
57	Projeto de software não comercial	1
58	Semântica	1
59	Sistemas	1
60	Sistemas centralizados	1
61	Sistemas de informação	1
62	Técnicas de computação	1
63	Técnicas de programação	1
64	Teoria da computação	1
65	Thread	1
66	Trabalho em equipe	1

3.2 Considerações sobre a pesquisa com professores

Algoritmo foi o conceito que obteve maior votação, seguido de Lógica, Estrutura de Dados, Linguagens de programação, Autômatos e Máquinas, Computabilidade, Matemática e Álgebra.

Baseado nessa votação, percebe-se que esses conceitos podem ser reorganizados em três grupos:

- **Matemática fundamental:** Lógica; Abstração; Matemática e Álgebra.
- **Informática teórica:** Algoritmos (conceito); Programa (conceito); Linguagens Formais e Autômatos; Computabilidade; Complexidade; Formalismos e Máquinas (conceito).
- **Informática aplicada:** Algoritmos (técnicas de programação); Programa (disciplinas práticas); Linguagens de Programação; Estrutura de Dados;

Redes; Banco de Dados (dados/informações); Modelagem Conceitual; Máquinas (arquitetura de computadores); Sistemas Distribuídos e Sistemas Operacionais.

Esse agrupamento pode ser, também, encontrado nas diretrizes curriculares dos cursos da área de computação e informática propostos pelo Ministério da Educação e Cultura [MEC 2002], onde os currículos são separados em quatro grandes áreas de formação:

- **Formação básica:** compreende os princípios básicos da área, a Ciência da Computação, a matemática necessária para defini-los formalmente, a física e a eletricidade necessária para permitir o entendimento e o projeto de computadores viáveis tecnicamente e a formação pedagógica que introduz os conhecimentos básicos da construção do conhecimento, necessário ao desenvolvimento da prática do ensino de computação.
- **Formação tecnológica:** aplica os conhecimentos básicos no desenvolvimento tecnológico da computação.
- **Formação complementar:** permite uma interação dos egressos dos cursos com outras profissões.
- **Formação humanística:** dá ao egresso uma dimensão social e humana.

Ainda, segundo as diretrizes, a Ciência da Computação é a área mais importante na composição dos currículos dos cursos. Seu ponto central está nos conceitos de máquina e algoritmo e suas sub-áreas são: Programação; Computação e Algoritmos; Arquitetura de Computadores; Matemática; Física e Eletricidade; e Pedagogia.

3.3 Pesquisa com crianças

A última pesquisa (anexo 3) foi realizada com 25 crianças, pertencentes a escolas pública e particular da cidade de Presidente Prudente-SP, com idades de 9 a 11 anos, idades essas sugeridas pela psicóloga que auxiliou na pesquisa.

Depois de perguntas sobre o nome, idade e série de cada criança seguiu-se uma conversa sobre o que cada uma achava da escola, suas preferências a respeito das disciplinas e o que a criança gostava de fazer quando não estava na escola. Após, foi pedido que respondessem o que gostariam de aprender do Computador e de Ciência da Computação.

Para que as crianças pudessem efetuar distinção entre esses dois conceitos, foram dados dois exemplos:

- “Quando a professora ensina a somar, subtrair, ela ensina a mexer com a calculadora?”
- “Sobre um carro, pode-se aprender a dirigir-lo ou a construí-lo.”

Além dessa conversa particular, foi pedido que desenhassem um ambiente que elas considerassem ideal para o ensino de Ciência da Computação.

As crianças foram colocadas em grupos (de 5 a 10), em uma grande mesa, contendo materiais para desenho. Cada uma fez um ou mais desenhos, de acordo com seu desejo.

3.4 Considerações sobre pesquisa com crianças

As respostas sobre o que gostariam de aprender sobre o computador foram: aprender a jogar, conversar na internet, escrever textos, desenhar, aprender sobre assuntos das diversas disciplinas da escola. E quando indagadas sobre Ciência da Computação, responderam: construir o computador, entendendo suas partes, e seu próprio jogo.

Durante a conversa, percebeu-se que as crianças ficaram impressionadas de saber que podem fazer seu próprio jogo e que podem construir o computador. Todavia uma das crianças acrescentou que só queria construir o computador com alguém ao seu lado.

O ambiente imaginado pelas crianças é bem alegre, contém móveis, janelas e vasos de flores. Na tela do computador, sempre há um desenho colorido, e em vários desenhos, há uma criança sorrindo.

Em alguns desenhos, foi verificado que algumas crianças possuem uma noção maior do funcionamento do computador, pois algumas o ligavam à energia elétrica.

Foram verificados desenhos muito parecidos em sua estrutura, concluindo-se que várias crianças preferiram copiar o desenho de seu amigo.

3.5 Considerações

Neste capítulo, procurou-se identificar alguns dos conceitos fundamentais da Ciência da Computação, identificar uma faixa de idade em que poderiam ser aplicados e verificar se esses conceitos seriam bem aceitos entre as crianças dessa faixa de idade.

As próximas etapas serão o levantamento bibliográfico, seleção dessa bibliografia e adaptação desses conceitos para uma linguagem mais simples, clara e elaboração de demonstrações e exercícios, utilizando-se fatos comuns do cotidiano.

4 Metodologia do Ensino de Ciência da Computação para Crianças

Identificados os conceitos fundamentais da Ciência da Computação, faz-se necessário um estudo sobre como serão introduzidos esses conceitos e as formas de abordá-los de conformidade com o currículo das disciplinas existentes. Para tanto, tomar-se-á, como base, a história da educação e como as atuais disciplinas foram introduzidas no currículo escolar atual.

A história é o entendimento das transformações provocadas pelo homem no tempo. A Pedagogia é a arte e a ciência de ensinar, isto é, a ação do homem quando transmite seus conhecimentos. Tem, como preocupação, o homem em sua totalidade e é centrada na formação do pensamento. Assim, a Pedagogia, no ensino de Ciência da Computação, pode auxiliar na formação do pensamento computacional, aproveitar métodos usados em outras ciências e identificar e eliminar obstáculos da aprendizagem.

O caminho para o ensino de Ciência da Computação deve ter, como preocupação, a solução de problemas e deve ser centrada na experiência do aluno, além de atuar em domínio específico do conhecimento.

Entender e estudar a história da computação é, antes de mais nada, entender a evolução da matemática. De acordo com [BOY 91], o resultado da construção de técnicas que permitiram, no passado, calcular, por exemplo, áreas e tangências coincide com o próprio desenvolvimento tecnológico da humanidade. Dessa feita, o primeiro ano de Ciência da Computação deve ser fundamentado com o ensino da matemática. Exemplo: noções de lógica da Matemática e noção de condicional da Ciência da Computação.

A sólida fundamentação para a prática da Ciência da Computação tem como objetivos desenvolver habilidades necessárias para pensar, para se expressar clara e precisamente, a fim de solucionar problemas e criar conceitos.

Agora são feitas considerações gerais sobre os conceitos identificados como fundamentais e, posteriormente, serão organizados, estudados, definidos e, também, selecionados e elaborados exercícios sobre eles.

Um *algoritmo* é a descrição de um conjunto de ações que, obedecidas, resultam numa sucessão finita de passos discretos, atingindo um objetivo. Uma vez que se tenha o algoritmo, deve-se ter uma forma de representá-lo, de modo que a *máquina* entenda, assim, os algoritmos, em Ciência da Computação, são representados, por meio de *programas*. Essa representação deve ocorrer mediante um conjunto de comandos facilmente compreensíveis e sem ambigüidades. Existe uma grande diversidade de esquemas para a representação de algoritmos, conhecidos como *linguagens de programação*. Para [NET 87], existem duas maneiras básicas para a formalização de linguagens de programação: *gramática* (mecanismos de geração) e reconhedores (*autômatos*).

A busca de algoritmos, visando a controlar tarefas cada vez mais complexas, conduz à investigação da existência ou não de algoritmos que solucionem determinados problemas. Segundo [DIV 99], o objetivo do estudo para a solução de problemas é investigar a existência ou não de algoritmos que solucionem determinada classe de problemas. Ou seja, investigar os limites da *computabilidade* e, conseqüentemente, os limites do que pode efetivamente ser implementado em um computador.

Dessa feita, se não existir um algoritmo que leve à solução de um problema, este não poderá ser resolvido por uma máquina. Portanto, são problemas não computáveis. Agora, são detalhados cada um dos conceitos identificados como fundamentais no estudo de Ciência da Computação.

4.1 Algoritmos

A palavra algoritmo, na matemática [FON 98], designa um procedimento geral de cálculo, que se desenvolve, por assim dizer, automaticamente, poupando esforço mental durante o seu curso.

Segundo [FOR 93], *"Algoritmo é uma seqüência de passos que visam atingir um objetivo bem definido"*.

Para [GUI 85], *"Um algoritmo é a descrição de um padrão de comportamento, expressado em termos de um repertório bem definido e finito de ações primitivas, das quais damos por certo que elas podem ser executadas"*.

Em [BRO 2000], algoritmo é definido da seguinte forma: *"Um algoritmo é um conjunto ordenado de passos executáveis não ambíguos, definindo um processo que tem um término"*.

Nota-se que, em todas estas definições, as características principais dos algoritmos, como também em [AHO 74] e [TER 91], são: não podem ser ambíguos, devem ser finitos e, se os passos definidos forem seguidos, deverão conduzir à solução de um determinado problema.

O objetivo de um algoritmo é o de ensinar a resolver problemas, ou seja, não se está preocupado com fornecer o resultado da solução do problema. Exemplos clássicos desse fato são as receitas de bolos. Quando se possui uma receita de bolo, têm-se os passos, para que seja possível obter-se o resultado final: o bolo. Nenhuma receita traz o bolo pronto.

4.1.1 Lógica

A lógica fundamenta os raciocínio as ações; o pensamento lógico geralmente é criativo inovador. A cabeça humana é uma máquina notável que não pode nem deve ser robotizada. O raciocínio lógico lubrifica torna mais produtivo o pensa e m direção ao porvir. É dos hábitos da reflexão que brota o aprender. [SER 2000] pág. 21.

Uma primeira etapa para o ensino de algoritmo para crianças deve ser a introdução do conceito de *lógica*, que, de acordo com [FOR 93], é a matéria-prima para elaboração de algoritmos.

Isto é verdadeiro, pois a lógica fornece os instrumentos mentais necessários, para enfrentar-se qualquer tipo de investigação.

... a Lógica é a arte de pensar corretamente e, visto que a forma mais complexa do pensamento é o raciocínio, a Lógica estuda ou tem em vista a "correção do raciocínio". [FOR 93] pág. 1

Segundo [FON 98], a lógica foi considerada, na tradição clássica e medieval, instrumento indispensável ao pensamento científico. Da Ciência Lógica, nasceu a Lógica Matemática e, dentro desta, várias filosofias da lógica, que interpretam os cálculos simbólicos e sua sistematização axiomática. Para a Computação, interessa abordar-se, em particular, a questão do pensamento dedutivo e matemático, seus limites, o problema da relativa mecanização do pensamento quantitativo e o problema da Inteligência Artificial.

Com Aristóteles (384 a.C. – 322 a.C.), é que se dá o verdadeiro nascimento da Lógica.

De acordo com [JOS 95], Aristóteles estudou as relações de causa e efeito. Causa: como acontecem, todas coisas têm uma finalidade, causa material (substancial), causa formal (forma), causa atuante (eficiente). Exemplo: por que chove? Vapor d'água esfria (causa material); a nuvem estava bem no lugar onde o ar se esfriou (causa formal); as plantas e os animais precisam de chuva (causa final).

Concluiu que, atrás de todos os fenômenos naturais, há uma finalidade: Chove, para que as plantas cresçam, dêem frutos e alimentem o homem. Poder-se ia dizer, simplesmente, que Deus criou as águas (rios, chuvas) para o homem, não que a água goste deles!

A diferença entre forma e substância é imprescindível para o homem reconhecer as coisas do mundo. Quanto mais se conhece, mais se tem capacidade de ordenar, classificar, agrupar-se em categorias. Distinguem-se coisas vivas de coisas mortas: pedra, animal, plantas. Tudo da natureza pertence a grupos e subgrupos. Exemplo: cachorro (animal, mamífero, macho etc).

Se encontrar algo e não souber o que é, não a tocará com medo. A brincadeira de se isolar uma pessoa, enquanto um grupo escolhe um objeto para adivinhar o que é, foi inventada por Aristóteles. Quem conhecer sua filosofia facilmente adivinhará o objeto escolhido.

Foi querendo organizar os conceitos do homem que Aristóteles criou a "Lógica". Premissas: premissa maior: "toda criatura viva é mortal"; premissa menor: "o cão é vivo", conclusão: "o cão é mortal".

Já se sabia que o cão era mortal, mas Aristóteles organizou, formou conceitos que podem ser usados em outras situações. Sabendo-se que um rato é mamífero sabe-se que ele se alimenta da mãe, apesar de nunca ter sido visto mamando.

4.1.2 Álgebra

Segundo [FON 98], a *álgebra* é ciência que fundamenta o desenvolvimento da Computação. De fato, o computador e todos os instrumentos que o precederam (régua de cálculo, máquina de Pascal, a calculadora de Leibniz, a máquina analítica de Babbage etc.) são manifestações práticas que foram surgindo, com naturalidade, como resultado da busca pelo homem de reduzir os problemas a equações matemáticas, resolvendo-as, segundo regras.

O termo álgebra vem do árabe al-jabr ou, literalmente, a "reunião de partes quebradas". Álgebra é a arte dos raciocínios perfeitos.

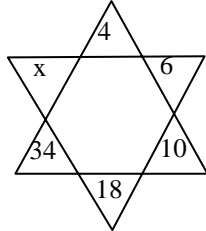
São exemplos de exercícios usuais a alguns livros constantes na fig. 1.1, que podem ser trabalhados com os alunos, e o anexo 4 contém o restante desses exercícios.

1. Se o salário de um homem é de 2.000,00 por semana, se ele gasta 1.400,00 por semana, quanto demorará, para juntar 3.000 em economia?
2. Se 2 lápis custam 5,00, quantos lápis poderão ser comprados com 60,00?
3. Tenho 1 caixa: dentro dela 2 outras caixas; dentro de cada uma delas, outras 2. Quantas caixas tenho?
4. Tendo-se 2 vasilhas de 5 e 3 litros, como se conseguem exatamente 7 litros de água?
5. Tendo-se 2 vasilhas de 7 e 5 litros, como se conseguem exatamente 8 litros de água?

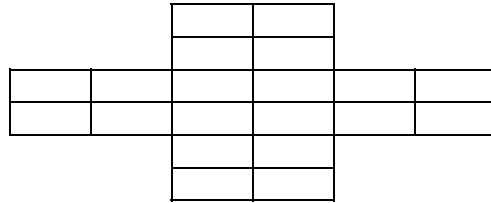
FIGURA 4.1 - Exercícios sobre lógica e aritmética

[SER 2000] contém vários exercícios em que são trabalhados os raciocínios lógicos matemático, quantitativo, numérico, analítico e crítico, como nos exemplos das fig. 4.2 e 4.3.

1. Considere os números escritos nos pequenos triângulos das pontas da figura estrelada e determine o valor de "x".



2. Dispor, nos quadradinhos da figura abaixo em forma de crucifixo, os números de 1 até 20, de modo que as somas nas horizontais e verticais das hastes horizontais e verticais da cruz seja, sempre, igual a 71.



3. Quatro amigas vão ao teatro e uma delas resolve entrar sem pagar. Aparece um vigilante e quer saber qual delas entrou sem pagar.

"Eu não fui", diz a Gabriela.

"Foi na Graciela", diz a Manuela.

"Foi a Daniela", diz a Graciela.

"A Manuela não tem razão", diz a Daniela.

Só uma delas mentiu. Quem não pagou o bilhete?

4. Analise, quanto à validade, o seguinte argumento:

Um número inteiro é par ou ímpar.

3 é um número inteiro.

3 é par.

Logo, 3 não é ímpar.

5. Era um caracol tão pequeno, que quase sumia. Iniciando, no chão, na grande palmeira subia. Usando, sempre, do máximo de energia, todos os dias 6 metros para cima fazia; mas, à noite, sempre 4 metros descia. Ao anoitecer do 15^o dia, a subida teve fim. Diga baixinho, apenas para mim, qual a altura da palmeira do jardim?



FIGURA 4.2 - Exemplos de exercícios sobre lógica em [SER 2000]

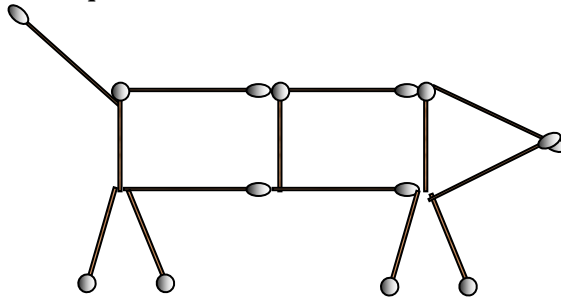
6. Considere 5 casinhas e indique, em cada uma delas, quem é o proprietário, qual a cor, que bebida usa, qual o tipo de carro e qual o animal que cria, orientando-se nos seguintes dados:

- o sergipano mora na casa azul;
- o paraense tem um carro Monza;
- o gaúcho possui um gato;
- o carioca bebe cerveja;
- os canários estão a mesma distância do Omega e do vinho;
- o sabiá não bebe whisky e não mora na casa branca;
- na casa rosa bebe-se vodka;
- o curió é vizinho da casa onde se bebe cerveja;
- a casa rosa é vizinha da casa da direita, cinza;
- o paraense e o gaúcho são vizinhos;
- o proprietário do Vectra cria canários;
- a Pampa pertence à casa de cor verde;
- bebe-se guaraná na terceira casa;
- o brasileiro é vizinho da casa branca;
- o proprietário do carro Escort bebe vinho;
- o proprietário do curió é vizinho do dono do Omega;
- o proprietário da Pampa é vizinho do dono do cachorro;
- o paraense e o gaúcho são vizinhos;



Proprietário: _____, _____, _____, _____, _____,
 Cor da casa: _____, _____, _____, _____, _____,
 Bebida: _____, _____, _____, _____, _____,
 Carro: _____, _____, _____, _____, _____,
 Animal: _____, _____, _____, _____, _____,

7. O "porquinho abaixo foi construído com 14 palitos de fósforo. Convém observar que ele está olhando para o lado direito. Deslocar 2 palitos de modo a fazê-lo olhar para o lado esquerdo.



8. Quantos triângulos há em cada uma das seguintes figuras?

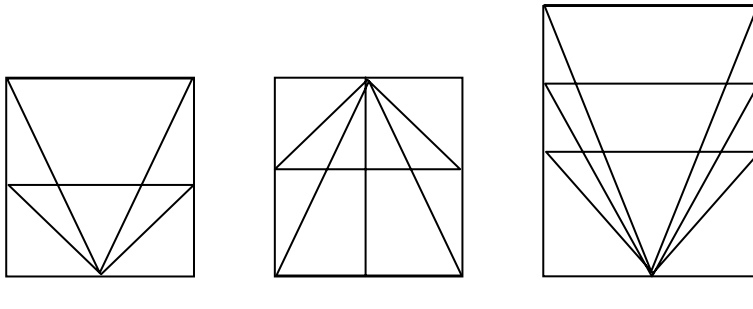






FIGURA 4.3 - Exemplos de exercícios de lógica em [SER 2000]




Na finalização desses conceitos, tem-se, na fig. 4.4, um exemplo de exercícios sugeridos por Piaget em [FUR 72], específicos de lógica simbólica-imagem fundamental para a Ciência da Computação. Todo o conjunto de exercícios se encontra no anexo 5, onde são abordados os conceitos relativos à negação, conjunção, disjunção, conjunção com negação e disjunção com negação.

Exemplo: $A \not\Rightarrow$  ($\not\Rightarrow$ novo símbolo)

A { ?
?
?
?
 $M \rightarrow$?
 $M \not\Rightarrow$ { ?
?
?
?

Exemplo: (\neg novo símbolo)

$\neg M \rightarrow$ { 


 $\neg C \rightarrow$? ?

$\neg C$?   

Exemplo: (\wedge novo símbolo)







$A \wedge C$ { \rightarrow  
 \rightarrow  
 $\not\Rightarrow$ 
 $\not\Rightarrow$ 

FIGURA 4.4 - Exercícios de lógica símbolo-imagem, Piaget em [FUR 72]

4.1.3 Representação de Algoritmos

Para se escreverem algoritmos, é necessário saber-se qual a linguagem a ser utilizada. Se for uma comunicação homem-homem, utilizar-se-á a linguagem natural (português, inglês etc.). Também poderá ser utilizada a linguagem das figuras, como em [BRO 2000], onde há o algoritmo, para se construir um pássaro, por meio de dobradura (anexo 6).

Todavia essas formas de comunicação podem levar a enganos, como na frase "O artesão produz formas perfeitas". Nesta frase, poder-se-ia achar que o artesão produz esculturas com f(ó)rmas perfeitas ou fabrica f(ô)rmas para bolos.

Segundo [BRO 2000], no caso do algoritmo para a construção do pássaro, é provável que alguns não conseguirão construí-los, somente seguindo as instruções. Ao contrário, necessitariam da definição precisa de uma linguagem. Essa definição precisa é chamada de primitiva. As fig. 4.5, 4.6 e 4.7 contêm as primitivas, assim como o algoritmo, a fim de se construir um balão por meio de dobraduras.

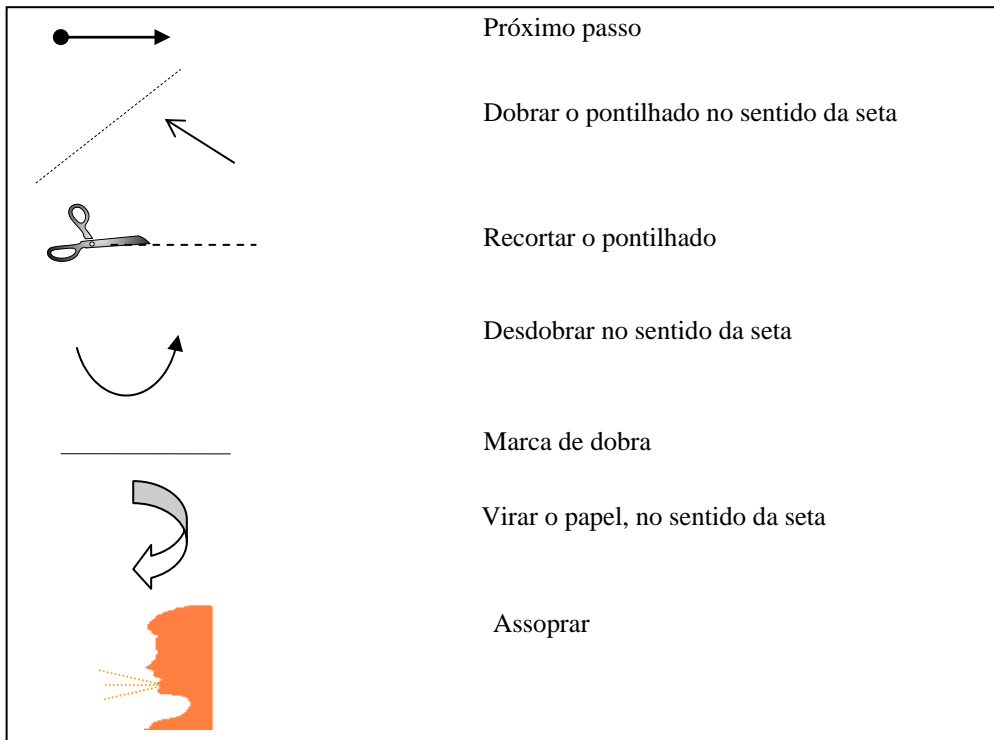


FIGURA 4.5 - Exemplo de primitiva para construção de um balão de dobradura.

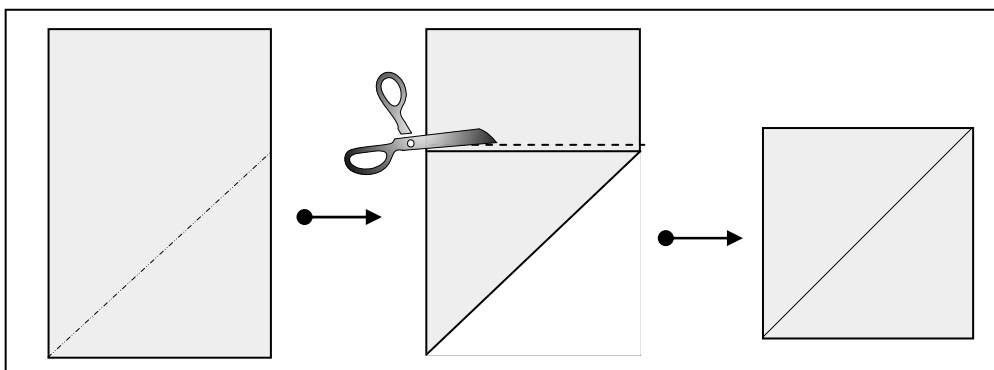


FIGURA 4.6 - Exemplo de algoritmo para transformar-se um papel com formato de um retângulo em um papel com formato de um quadrado.

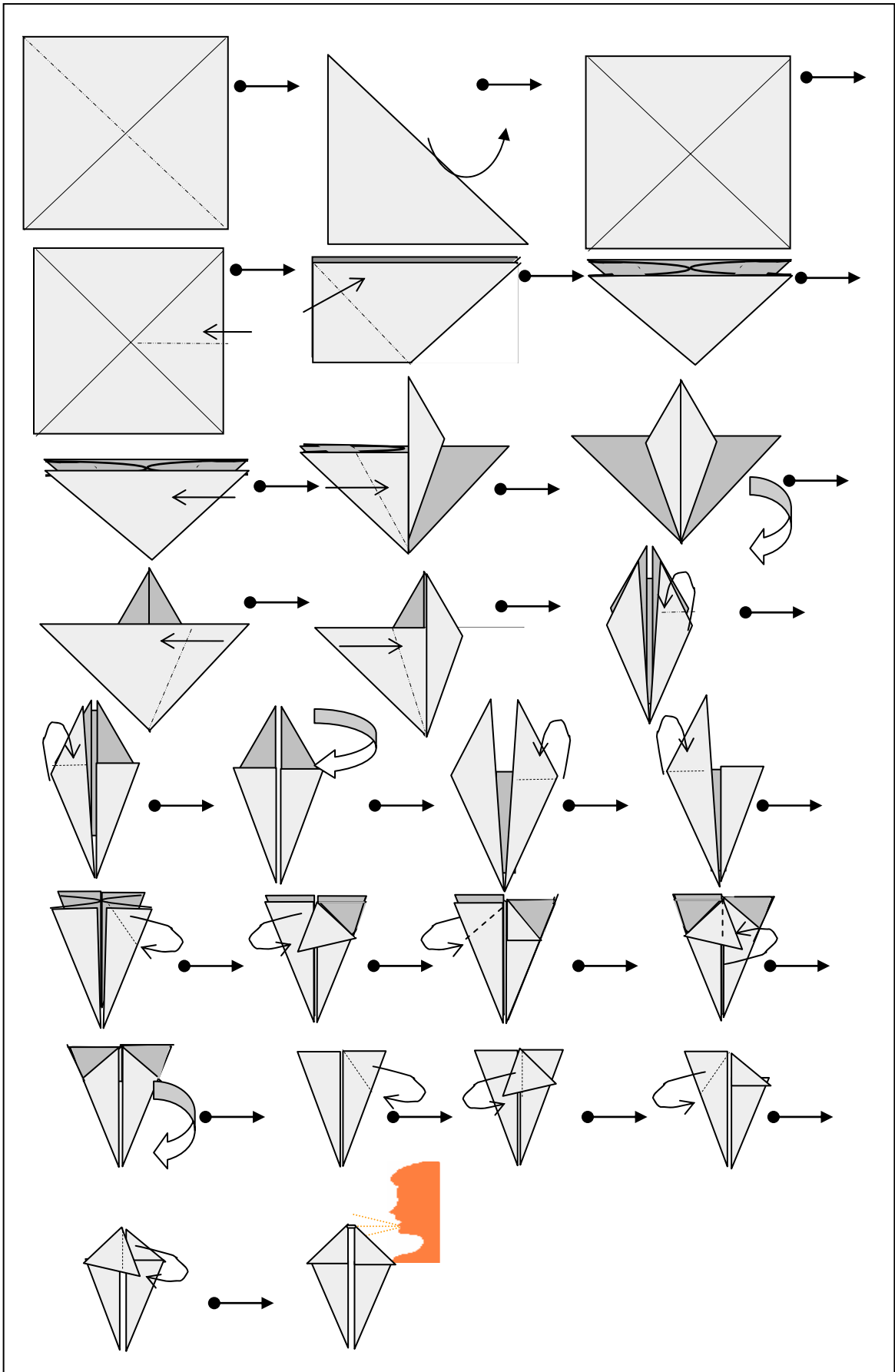


FIGURA 4.7 - Exemplo de algoritmo, para construir-se um balão de dobradura

A fig. 4.8 apresenta um exercício tomando-se, por base, as primitivas dadas, deve ser construído o algoritmo, retirado de [FOR 93]. O anexo 7 contém, além desse exercício, outros que, utilizando-se das primitivas dadas, construa-se algoritmos, para 3 jesuítas e 3 canibais atravessarem um rio em um barco e para o jogo da velha. No jogo da velha, é possível que se utilizem de estruturas de seleção e/ou repetição.

Utilizando as seguintes primitivas construa o algoritmo para o seguinte problema:

Suponha que você possua um robô e queira fazê-lo trocar uma lâmpada, sendo que o mesmo foi programado para obedecer às seguintes primitivas:

- pegue <objeto>
- pressione <objeto>
- gire garras 180° no sentido horário
- gire garras 180° no sentido anti-horário
- mova <objeto> para <lugar>
- desloque-se para <lugar>

e ainda é capaz de:

- perceber quando algum comando não mais pode ser executado
- sentir alguma fonte de calor

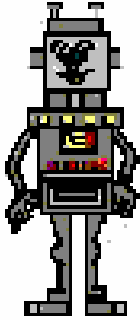


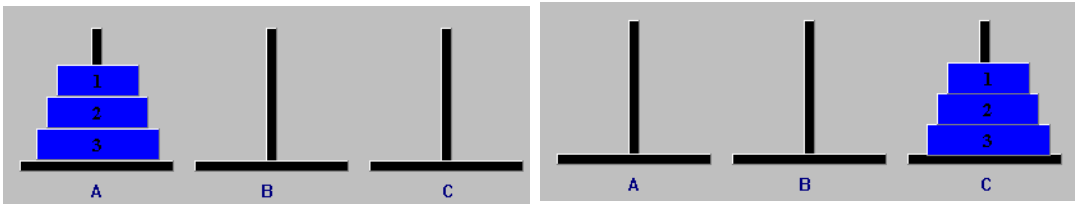
FIGURA 4.8 - Exercício para construção de algoritmo

O próximo exercício solicita que, além do algoritmo, sejam construídas suas primitivas, fig. 4.9. No anexo 8, encontram-se exercícios para a construção de primitivas e algoritmos que resolvam os problemas, a fim de um homem atravessar um rio com três cargas, visando a resolver uma equação do primeiro grau; e, por último, atravessar um cruzamento de ruas.

Construir as primitivas e o algoritmo para o seguinte problema:

[GRA 94] Problema da Torre de Hanói

São dados três pinos com 3 discos, inicialmente empilhados, por tamanho decrescente, em um dos três pinos. O objetivo é transferir a torre inteira de discos para um dos outros pinos, movendo apenas um disco de cada vez, e nunca colocando um disco maior em cima de um menor. Apenas um disco pode ser movido por vez e cada disco deve, sempre, ser colocado em um dos pinos.



Configuração inicial Configuração final

FIGURA 4.9 - Exercício para construção das primitivas e algoritmo para o problema da "Torre de Hanói"

Também pode ser considerado um algoritmo um tutor dos passos a serem seguidos na ferramenta *Word*, para a construção nessa ferramenta de uma folha

personalizada. A figura 4.10 contém a primeira etapa desse algoritmo; e o anexo 9 contém todo o algoritmo com a segunda parte onde é colocada uma figura no fundo do papel. O interessante desse exercício seria pedir que as crianças identificassem quais as primitivas.

**Algoritmo para construir um papel de carta personalizado (com borda e figura)
Quais seriam as primitivas?**

Siga os seguintes passos:
1ª Parte: Colocar uma Borda decorada:

1. Clique no Menu **Formatar** e escolha a opção **Bordas e Sombreamento**
2. Escolha a opção **Borda de Página**
3. Na caixa **Arte**, escolha o desenho que deseja para sua borda
4. Clique em **OK** e terá uma folha com bordas decoradas

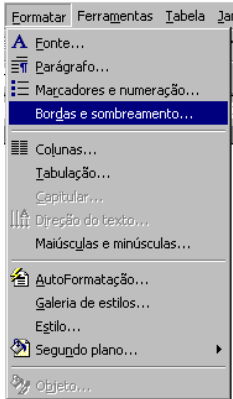
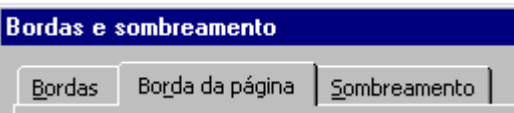
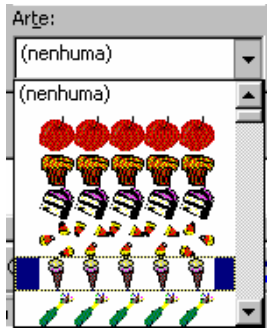
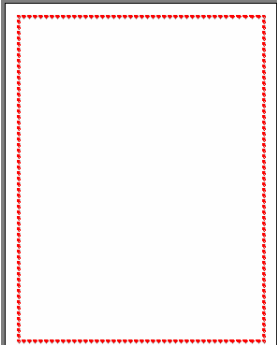





FIGURA 4.10 – Algoritmo, para se construir um papel de carta personalizado, por meio da ferramenta "Word"

Têm-se, como sugestões, outros exercícios, que podem ser trabalhados:

- propor aos alunos que, utilizando as primitivas de origami (anexo 6), construam algoritmos, para a fabricação de avião, barco, copo etc.;
- como já sugerido, propor a construção das primitivas para o algoritmo que ensina a fazer um papel de carta (anexo 8);
- utilizando as primitivas acima, propor a construção de outros algoritmos, para completar o papel de carta, com, por exemplo, a inserção de Cabeçalho e Rodapé;
- propor que os alunos construam primitivas e algoritmos diferentes dos já utilizados.

4.2 Programas

A definição precisa da linguagem a ser utilizada é de fundamental importância, para que se evitem as várias interpretações.

A Ciência da Computação trata estes problemas estabelecendo um conjunto bem definido de elementos funcionais básicos com os quais podem ser construídas representações de algoritmos. Cada um destes blocos construtivos é dito uma operação primitiva, ou simplesmente, primitiva. [BRO 2000] pág. 151

Para aproximação com a linguagem usual da Ciência da Computação, em vez de se utilizar o termo *primitivas*, utilizar-se-ão *comandos*. Cada um dos comandos deverá ter, claramente definidos, sua sintaxe e semântica.

O conjunto de comandos utilizados, para se escreverem algoritmos, constitui uma *linguagem de programação*. Os algoritmos escritos, utilizando-se os comandos de uma linguagem de programação, são chamados de *programas*.

Devido à diversidade de máquinas e de linguagens de programação em [DIV 99], a formalização de conceitos de programas e de máquina não é baseada em qualquer linguagem ou computador real. Suas definições são feitas por meio dos modelos matemáticos. Esses são simples, para um fácil e rápido entendimento, e também facilitam a demonstração de resultados.

A distinção entre programa e máquina é importante na Ciência da Computação, uma vez que o programa (ou algoritmo) independe da máquina e possui uma complexidade estrutural e computacional (quantidade de trabalho necessário para resolver o problema). [DIV 99] p. 192.

Em [DIV 99] o conceito de programa é visto como um conjunto de operações e testes que são compostos de acordo com uma estrutura de controle. Os tipos de programas são determinados, segundo a estrutura de controle, e são três:

- Monolítico, com apenas desvios condicionais e incondicionais;
- Iterativo, com estruturas de iteração de trechos de programas;
- Recursivo, baseado em sub-rotinas recursivas.

O programa do tipo *monolítico* é estruturado, utilizando-se desvios condicionais (dependendo da resposta, é executada uma tarefa) e incondicionais (após uma instrução, segue-se para outra). Pode ser especificado na forma de texto, por meio do uso de instruções rotuladas e na forma diagramática: fluxograma.

Para o desenvolvimento de programas, este trabalho utiliza-se dos programas do tipo monolítico em forma de fluxograma. Não se trabalha com os programas do tipo iterativo e recursivo, por três razões: o programa monolítico é o tipo mais fundamental de programas; apesar de não possuir mecanismos explícitos de repetição, permite simular-se uma repetição, utilizando-se suas estruturas; e recursão é um conceito importante e difícil de ser trabalhado, por possibilitar níveis diversos de abstração, sendo recomendado como trabalho futuro.

4.2.1 Identificadores

Para se trabalhar com os programas, é necessário o trabalho com os identificadores; e, para poder-se entendê-los e defini-los, é necessário o entendimento da "Teoria dos Conjuntos".

A fig. 4.11 e o anexo 10 contêm informações sobre um software que trabalha com conceitos e exercícios que podem ser utilizados para reforço sobre a Teoria dos Conjuntos em [LEO 97].

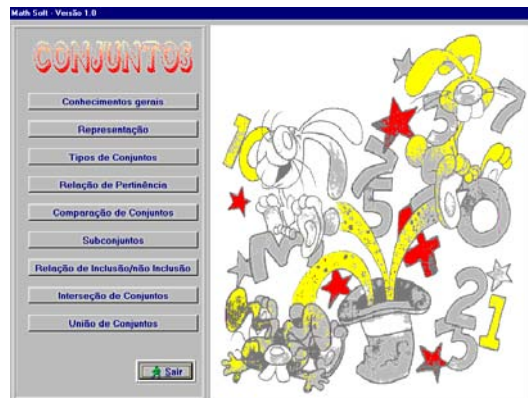


FIGURA 4.11 - Tela inicial do "Software para Ensino Fundamental: Teoria dos Conjuntos"

Identificador é o nome utilizado para incógnitas. De acordo com [MAN 88] e [COR 90], o Identificador possui 3 características que devem ser muito bem trabalhadas: nome, conteúdo e o tipo, exemplificados na fig. 4.12.

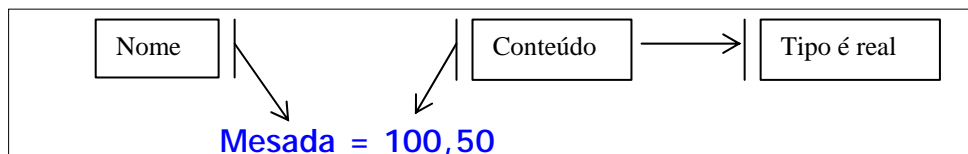


FIGURA 4.12 - Exemplo de um identificador e suas 3 características

O nome e o conteúdo do identificador são conceitos que devem ser claramente dissociados. Buscando aproximação na maneira como o computador trabalha, utilizam-se 4 conjuntos, para caracterizar os tipos dos identificadores de acordo com [AMS 2000]:

– **Inteiro:** Informação numérica pertencente ao conjunto dos inteiros.

$$\mathbf{Z} = \{ \dots, -2, -1, 0, 1, 2, 3, \dots \}$$

Exemplos: Mariana possui **3** irmãos.

As pessoas possuem **2** olhos e **1** nariz.

– **Real:** Informação numérica pertencente ao conjunto dos reais.

Exemplos: Camila pesa **30,500** gramas e tem **1,35** de altura.

A mesada de Lucas é de R\$ **100,50**.

- **Caracter:** Informação composta por um conjunto alfanumérico (a..z, A...Z, 0..9) e/ou caracteres especiais (!,",#,\$,%,&/,(),= etc).

Exemplos: O primeiro imperador do Brasil foi Dom Pedro I.

Diego mora na rua 12 de outubro, nº 234.

- **Lógico:** Informação que pode assumir somente dois valores, porém não ambos ao mesmo tempo.

Exemplos: O ventilador está *ligado* ou *desligado*.

O livro está *aberto* ou *fechado*.

O identificador do tipo lógico deverá assumir somente os valores *verdadeiro* ou *falso*.

A fig. 4.13 e o anexo 11 contêm exercícios de se completarem, para definição de identificadores.

Defina o nome, o tipo e dê exemplos de conteúdos para definir o identificador para a seguinte situação.

Exemplo: Definir o identificador para o conjunto de nome das frutas.

Nome	Tipo	Exemplos de conteúdos
Frutas	Caracter	"maça", "pêra", "goiaba"

Definir o identificador para o conjunto de valores de sua mesada.

Nome	Tipo	Exemplos de conteúdos

FIGURA 4.13 - Exercício para definição de identificador

4.2.2 Comandos

Como definido anteriormente, os comandos são as primitivas utilizadas, para se definirem os algoritmos. Os comandos iniciais a serem utilizados para a construção de algoritmos serão: atribuição e seleção (que serão utilizadas para simular-se uma repetição).

- **Atribuição:** Como em [HAR 92] o comando que associa um valor ou conteúdo a uma variável.

Exemplos:

Tem-se o identificador de nome **idade**, que é do tipo *inteiro*.

Idade

Depois de se aplicar o comando **Idade ← 10**.

Idade

No lado direito do comando, podem ser utilizadas expressões aritméticas.

Depois de se aplicar o comando $Idade \leftarrow 10 + 2$.

Idade

12

Como exercício, pode-se utilizar a fig. 4.14, tem em vista atribuírem-se valores nos identificadores definidos e o anexo 12 contém todo o conjunto de exercícios.

Completar o conteúdo dos identificadores de acordo com os comandos de atribuição.

Definir os conteúdos do identificador *Número* e *X*, que são do tipo *Inteiro*, após os comandos de atribuição.

Comandos	Número	X
Número \leftarrow 10 X \leftarrow 5		
Número \leftarrow 2 X \leftarrow 4		
Número \leftarrow 10 + 2 Número \leftarrow Número + 1		
Número \leftarrow X - 2 Número \leftarrow Número - X		

Definir os conteúdos do identificador *Nome1* e *Nome2*, que são do tipo *Caracter*, após os comandos de atribuição.

Comandos	Nome1	Nome2
Nome1 \leftarrow 'Maria'		
Nome2 \leftarrow 'Ana'		
Nome1 \leftarrow Nome1 + Nome2		

FIGURA 4.14 - Exercícios utilizando o comando de atribuição

– **Seleção ou Teste:** Como em [HAR 92], o comando, para seguir caminhos diferentes, dependendo de uma resposta a uma pergunta.

Deve-se saber que as respostas a uma pergunta assumem, somente, os valores *verdadeiro* ou *falso*.

A estrutura *se então senão* permite fazer-se uma escolha entre duas alternativas exclusivas, exemplificada através da fig. 4.15.

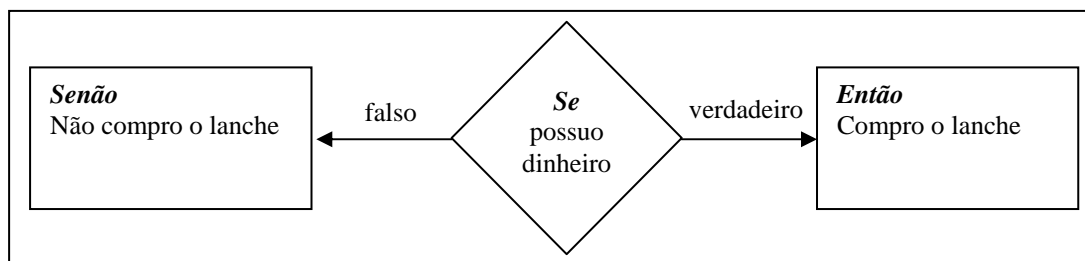
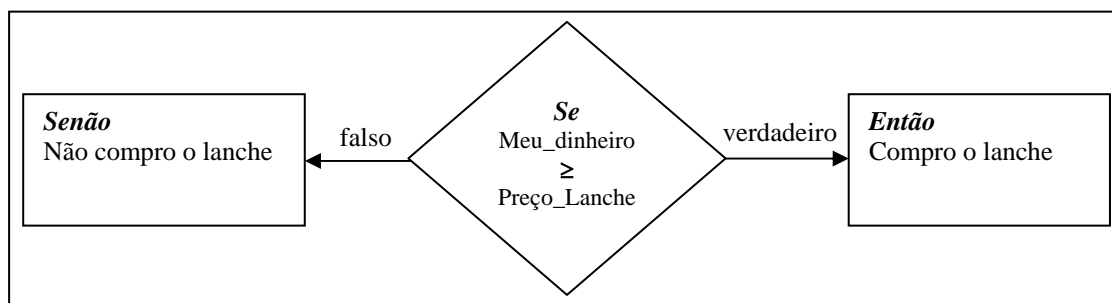


FIGURA 4.15 – Representação da estrutura de seleção

- **Operadores relacionais:** Utilizados para comparação entre 2 identificadores do mesmo tipo. O resultado de uma operação relacional será, sempre, um valor do tipo lógico (*verdadeiro* ou *falso*). Os operadores relacionais a serem utilizados estão na tabela 4.1 e na fig. 4.16 e contêm um exemplo utilizando estrutura de seleção e operador relacional.

TABELA 4.1 – Representação dos operadores relacionais, seu significado e exemplos de uso

Operador	Significado	Exemplos de uso
=	Igual a	Mesada de Valentina = Mesada de Elis
≠	Diferente de	Mesada de Franco ≠ Mesada de Enrico
<	Menor que	Quantidade de dentes de Aninha < Quantidade de dentes de Joaquim
>	Maior que	Quantidade de dentes de Joaquim > Quantidade de dentes de Aninha
≤	Menor ou igual a	Mesada de Paulo é ≤ Mesada de Antonio
≥	Maior ou igual a	Mesada de Mariana ≥ Mesada de Joana

FIGURA 4.16 – Exemplo da utilização do operador relacional \geq e estrutura de seleção.

- **Operadores lógicos:** Utilizados, para unir, separar ou negar os valores lógicos identificados na tab. 4.2. São utilizados nas tomadas de decisões e nos processos de repetições, como na fig. 4.17.

TABELA 4.2 – Operadores lógicos, representação e uso

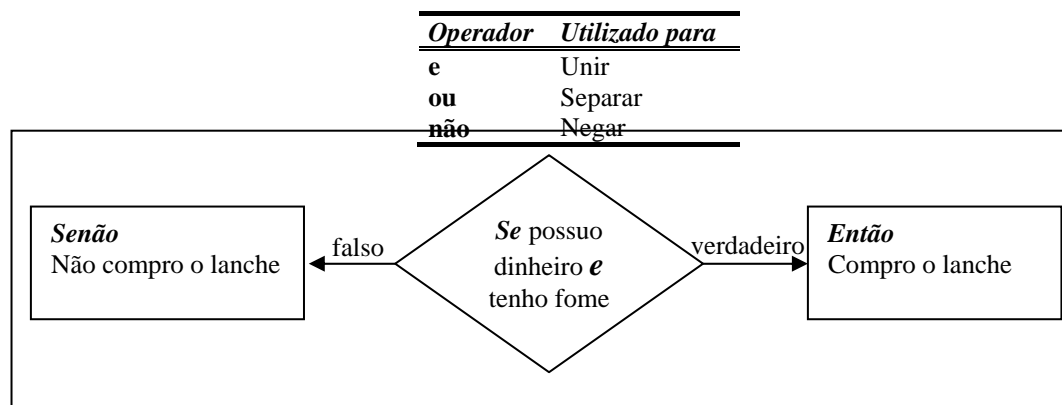


FIGURA 4.17 – Exemplo da utilização do operador lógico e e estrutura de seleção

- **Repetição:** É necessária, quando um determinado trecho do algoritmo, representando o mesmo comportamento, deve ser escrito, seqüencialmente, mais de uma vez. Escreve-se esse trecho apenas uma vez, mas o mesmo pode ser executado várias vezes. A estrutura de repetição a ser utilizada está exemplificada na fig. 4.18.

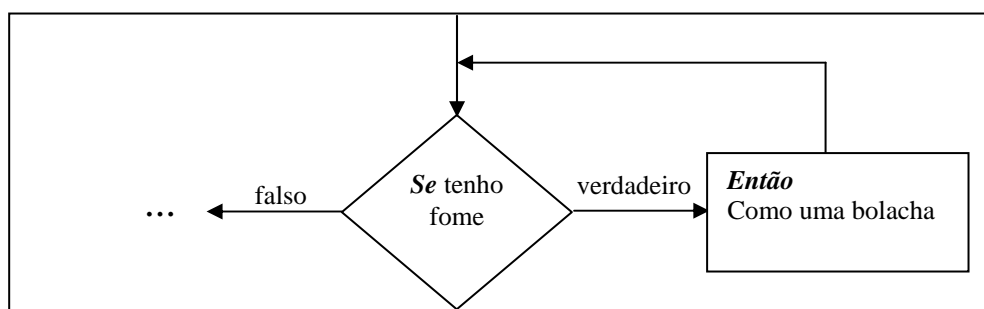


FIGURA 4.18 – Representação da estrutura de repetição

Nas fig. 4.19 e 4.20 são apresentados o início da solução de um exercício para a utilização da estrutura de seleção e um exemplo de exercício, utilizando a estrutura de repetição respectivamente.

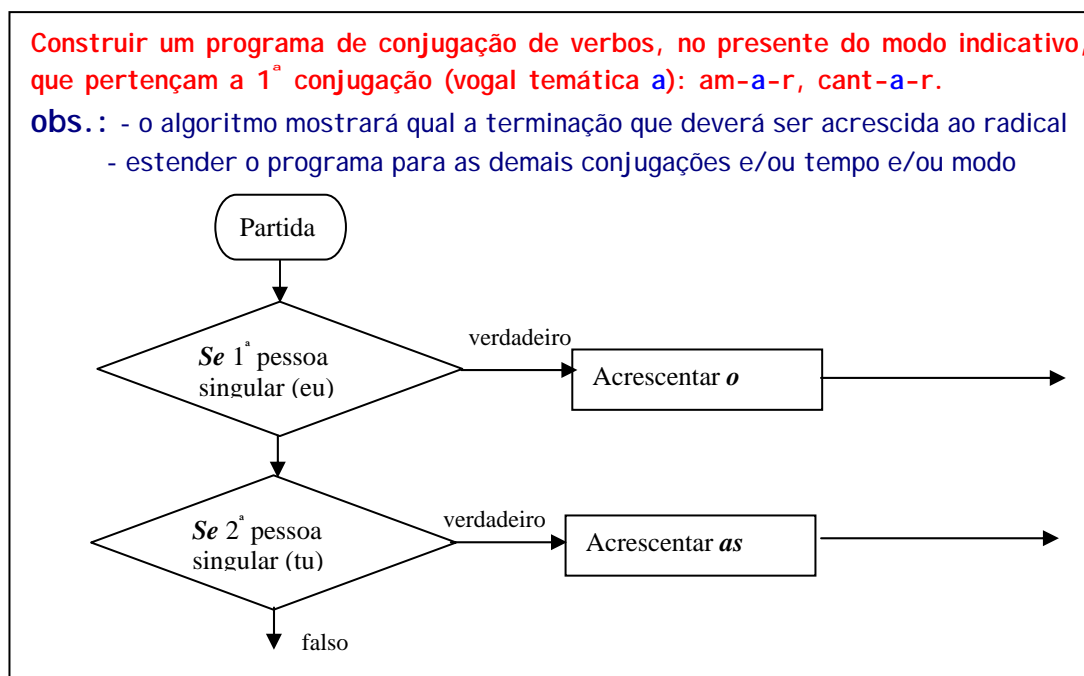


FIGURA 4.19 - Exercício utilizando estruturas de seleção

Construir um programa, para eleger um representante da classe.

Obs.:- I rão ser fornecidos os nomes de 3 candidatos

- Fazer uma repetição, para ir totalizando os votos de cada candidato



- Exibir qual candidato ganhou (fez o maior número de votos)

FIGURA 4.20 - Exercício utilizando estrutura de repetição

No anexo 13, encontram-se exemplos que, utilizando-se de fatos rotineiros, simulam as estruturas de seleção e repetição, servindo-se de operadores relacionais e aritméticos.

4.3 Máquinas

Diversas pesquisas foram realizadas por diversos matemáticos (Leibniz, Boole, Frege, Peano, Hilbert, Newman, Godël, Turing entre outros), com o objetivo de reduzir todo raciocínio a um processo mecânico.

A lógica simbólica, também conhecida como lógica matemática, tem o mesmo objeto que a lógica formal tradicional: estudar e fazer explícitas as formas de inferência, deixando de lado – por abstração – o conteúdo de verdades que essas formas possam transmitir, buscando, com o cálculo simbólico, a automatização de algumas operações do pensamento.

Os primeiros sistemas lógicos foram definidos por Boole e Frege. Segundo [FON 98], foram os que forneceram uma idéia clara de formalismo, concebendo a lógica como uma construção formal, para a qual se buscou, posteriormente, uma interpretação.

Dessa maneira, segundo [FON 98], a matemática se liberta dos conceitos de medida e contagem e torna-se o equivalente a um jogo, como o xadrez, onde regras arbitrárias são utilizadas, para manipular símbolos. No entanto, essa maior abstração implica um novo problema: se os axiomas não são verdadeiros em relação a nada, o que prova sua existência?

Um sistema formal pode ser visto como uma espécie de "jogo" rigorosamente definido, que especifica regras para manipulação de símbolos. O que caracteriza um sistema formal é muito semelhante às regras dispostas para um determinado "jogo". Para dizer a alguém como jogar e para estabelecer as regras que qualificam de formal um sistema, três aspectos desse jogo devem ser estabelecidos: a natureza dos símbolos, a descrição da situação inicial do jogo e uma lista de quais movimentos são permitidos a uma dada posição. [FON 98] p. 40.

Se se tivesse um jogo de xadrez, dever-se-ia saber quais são as peças e qual o seu formato, os movimentos dessas peças e como elas estão dispostas, inicialmente, no tabuleiro.

4.3.1 Máquina de Turing

A Máquina de Turing, em [FON 98], é um conceito abstrato, que, efetivamente, deu início à era dos computadores, baseou-se no princípio de que a simples aplicação de regras permite passar, mecanicamente, de uns símbolos a outros.

Turing analisou: Qual a forma utilizada pelos homens, para executarem um cálculo? Partindo de regras definidas, vão-se modificando os estados dos números, até que se chegue à solução. Dependendo da quantidade de dígitos, necessita-se guardar os estados intermediários, uma memória, que pode ser papel e lápis.

A Máquina de Turing, em [HOP 79], [LEW 98], [BIR 79] e [SIP 97], é composta de uma fita dividida em células, uma cabeça que lê e grava dados na fita, um

conjunto de regras que definem o movimento da cabeça e o estado inicial da fita, como na fig. 4.21.

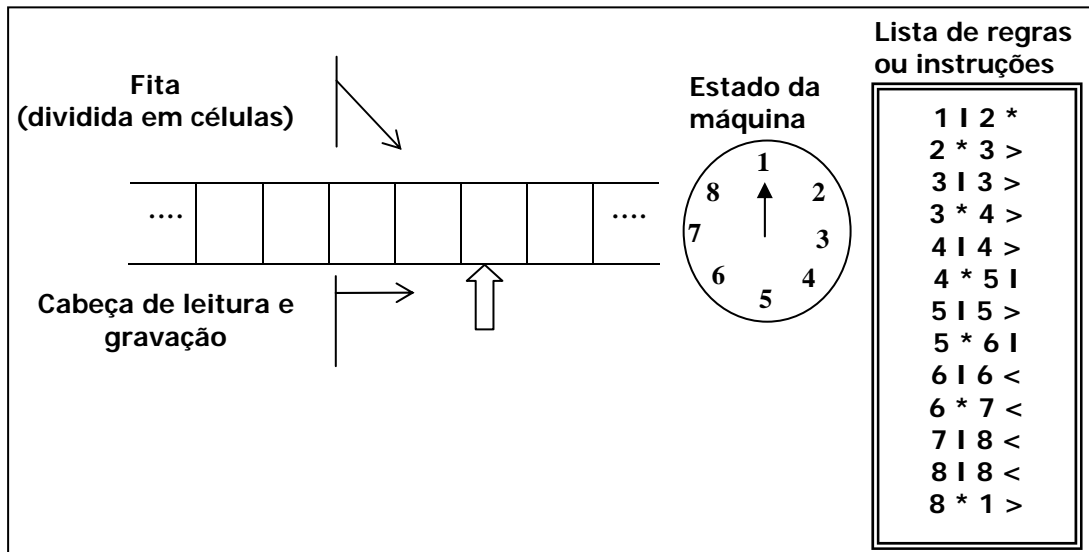


FIGURA 4.21 - Componentes de uma máquina de Turing

Em [FON 98], há um exemplo específico onde estão definidas as seguintes regras ou primitivas ou comandos:

- Substituir branco (*) por símbolo (**I**).
- Substituir símbolo (**I**) por branco (*).
- Ir um quadrado para a **direita** (>).
- Ir um quadrado para a **esquerda** (<).

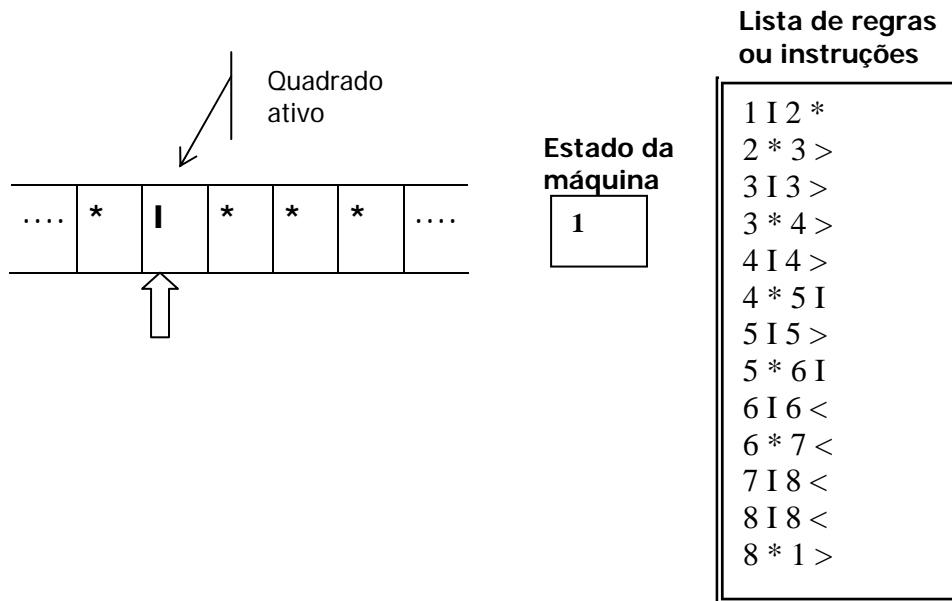
Através dessas regras, cria-se o conjunto de instruções; tabela 4.3.

TABELA 4.3 – Lista de regras contendo o código da regra e sua descrição

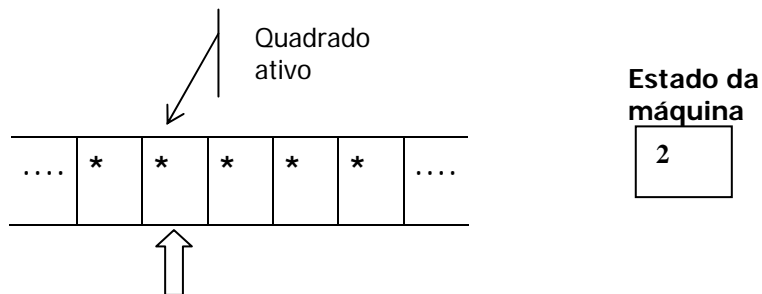
Código	Descrição
1 I 2 *	Estado 1: se há um I no quadrado ativo, substitua-o por * e vá para Estado 2
2 * 3 >	Estado 2: se há um * no quadrado ativo, vá para Estado 3 e ande um quadrado para direita
3 I 3 >	Estado 3: se há um I no quadrado ativo, vá para Estado 3 e ande um quadrado para direita
3 * 4 >	Estado 3: se há um * no quadrado ativo, vá para Estado 4 e ande um quadrado para direita
4 I 4 >	Estado 4: se há um I no quadrado ativo, vá para Estado 4 e ande um quadrado para direita
4 * 5 I	Estado 4: se há um * no quadrado ativo, substitua-o por I e vá para Estado 5
5 I 5 >	Estado 5: se há um I no quadrado ativo, vá para Estado 5 e ande um quadrado para direita
5 * 6 I	Estado 5: se há um * no quadrado ativo, substitua-o por I e vá para Estado 6
6 I 6 <	Estado 6: se há um I no quadrado ativo, vá para Estado 6 e ande um quadrado para esquerda
6 * 7 <	Estado 6: se há um * no quadrado ativo, vá para Estado 7 e ande um quadrado para esquerda
7 I 8 <	Estado 7: se há um I no quadrado ativo, vá para Estado 8 e ande um quadrado para esquerda
8 I 8 <	Estado 8: se há um I no quadrado ativo, vá para Estado 8 e ande um quadrado para esquerda
8 * 1 >	Estado 8: se há um * no quadrado ativo, vá para Estado 1 e ande um quadrado para direita

A simulação da máquina de Turing, de acordo com este conjunto de comandos e a fita iniciada com o valor **I** e o estado inicial **1**, ficaria das formas especificadas nas fig. 4.22, 4.23 e 4.24.

1. Configuração inicial da máquina de Turing



2. A máquina está no estado **1** e, de acordo com a lista de regras, se tiver um **I** no quadrado ativo, deve ser substituído por ***** e ir ao estado **2**.



3. A máquina está no estado **2** e, de acordo com a lista de regras, se tiver um ***** no quadrado ativo, deve ir ao estado **3**; e a cabeça de leitura/gravação deve andar um quadrado à **direita**.

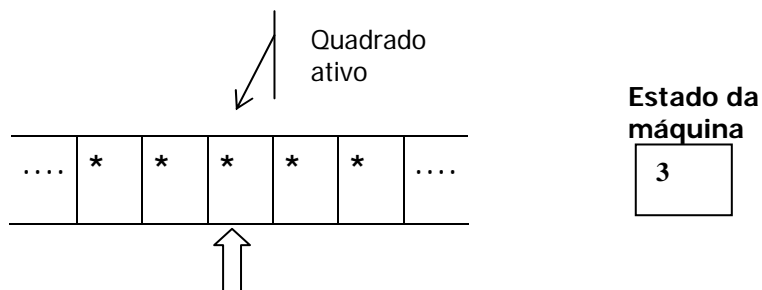
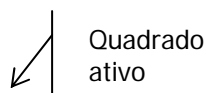
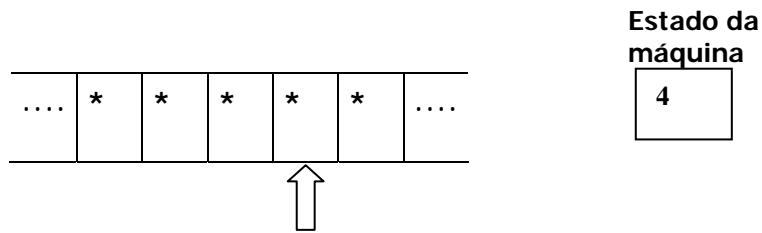


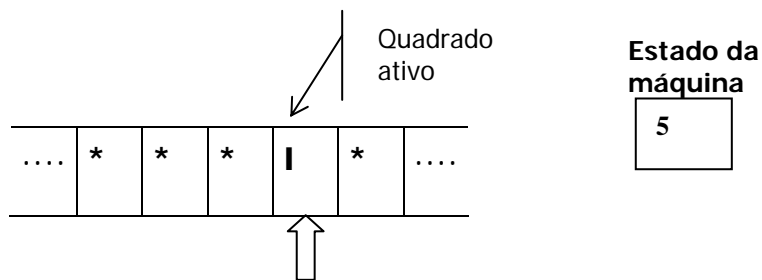
FIGURA 4.22 - Exemplo de simulação da máquina de Turing com a configuração inicial

4. A máquina está no estado **3** e, de acordo com a lista de regras, se tiver um ***** no quadrado ativo, deve-se ir ao estado **4**; e a cabeça de leitura/gravação deve andar um quadrado à **direita**.

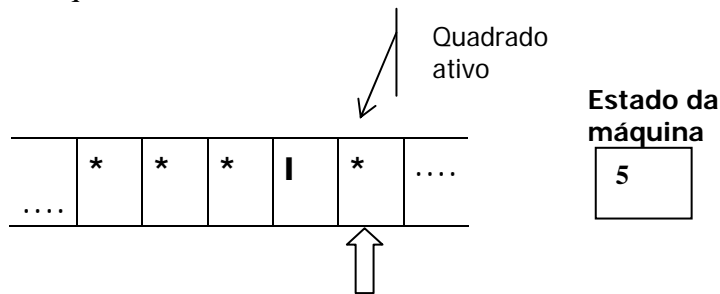




5. A máquina está no estado 4 e, de acordo com a lista de regras, se tiver um * no quadrado ativo, deve-se ir substituí-lo por I e ir ao estado 5.



6. A máquina está no estado 5 e, de acordo com a lista de regras, se tiver um I no quadrado ativo, deve-se ir ao estado 5 e a cabeça de leitura/gravação deve andar um quadrado à direita.



7. A máquina está no estado 5 e de acordo com a lista de regras, se tiver um * no quadrado ativo, deve-se substituí-lo por I e ir ao estado 6.

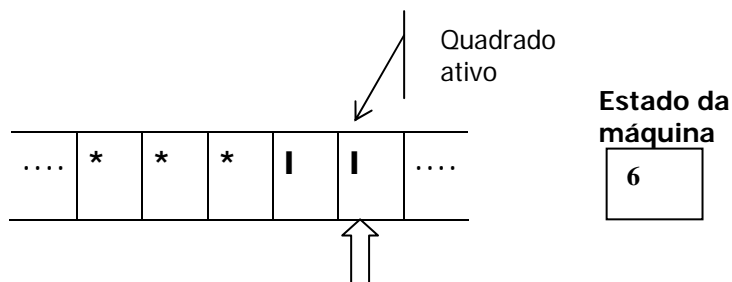
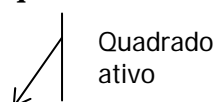
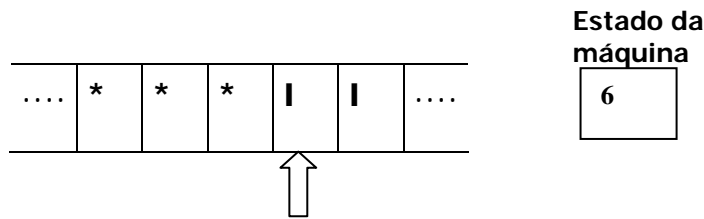


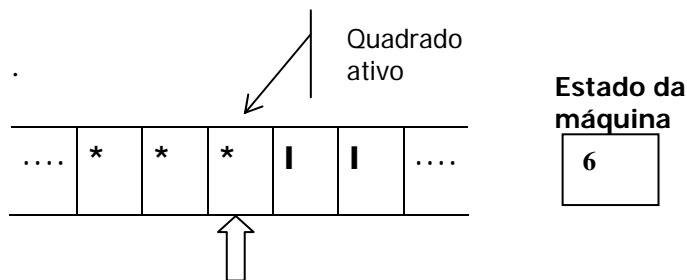
FIGURA 4.23 - Exemplo de simulação da máquina de Turing

8. A máquina está no estado 6 e, de acordo com a lista de regras, se tiver um I no quadrado ativo, permanece no estado 6 e a cabeça de leitura/gravação deve andar um quadrado à esquerda.

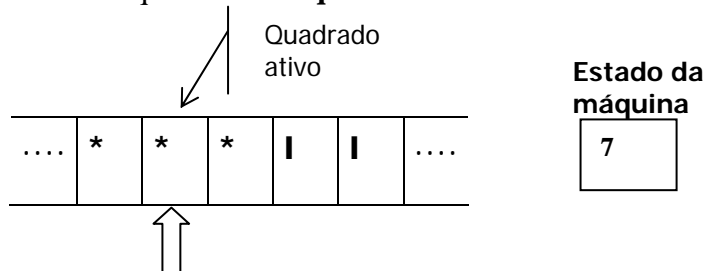




9. A máquina está no estado **6** e de acordo com a lista de regras, se tiver um **I** no quadrado ativo, permanece no estado **6** e a cabeça de leitura/gravação deve andar um quadrado à **esquerda**.



10. A máquina está no estado **6** e de acordo com a lista de regras, se tiver um ***** no quadrado ativo, deve-se ir ao estado **7** e a cabeça de leitura/gravação deve andar um quadrado à **esquerda**.



11. A máquina está no estado **7** e, de acordo com a lista de regras, não há instrução para o estado **7**; se há um *****, portanto, a máquina pára, pois não sabe o que fazer.

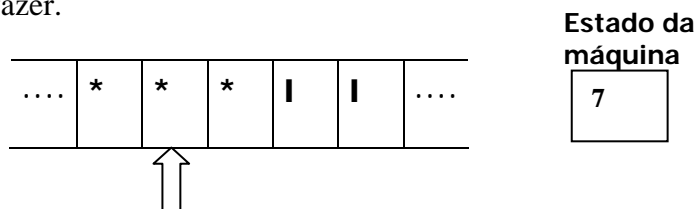


FIGURA 4.24 - Exemplo de simulação da máquina de Turing com a configuração final

A configuração final da máquina, ilustrada na figura 4.24 no número 11, contém o dobro de símbolos (**I**) iniciais.

O anexo 14 contém a demonstração, para, quando a fita, inicialmente contiver **II**, na configuração final, a fita terá **IIII**. O anexo 15 possui exercícios para construção de outras máquinas de Turing, como na fig. 4.25.

Descreva uma máquina de Turing, que substitua uma cadeia de **III** por uma de *******.

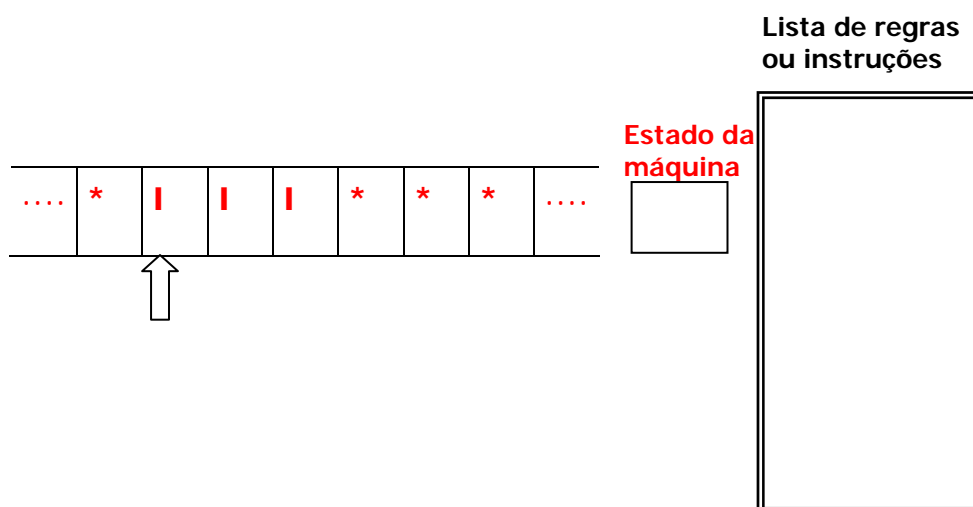


FIGURA 4.25 - Exercício utilizando a máquina de Turing

Na fig. 4.26, tem-se a tela pertencente a um software encontrado em <http://www.cic.unb.br/tutores/turing/MaqTur.html>, que simula o funcionamento de uma Máquina de Turing. Pode-se ver a fita, a cabeça de leitura e gravação, assim como as regras que determinam o movimento da máquina em um determinado momento; anexo 16.

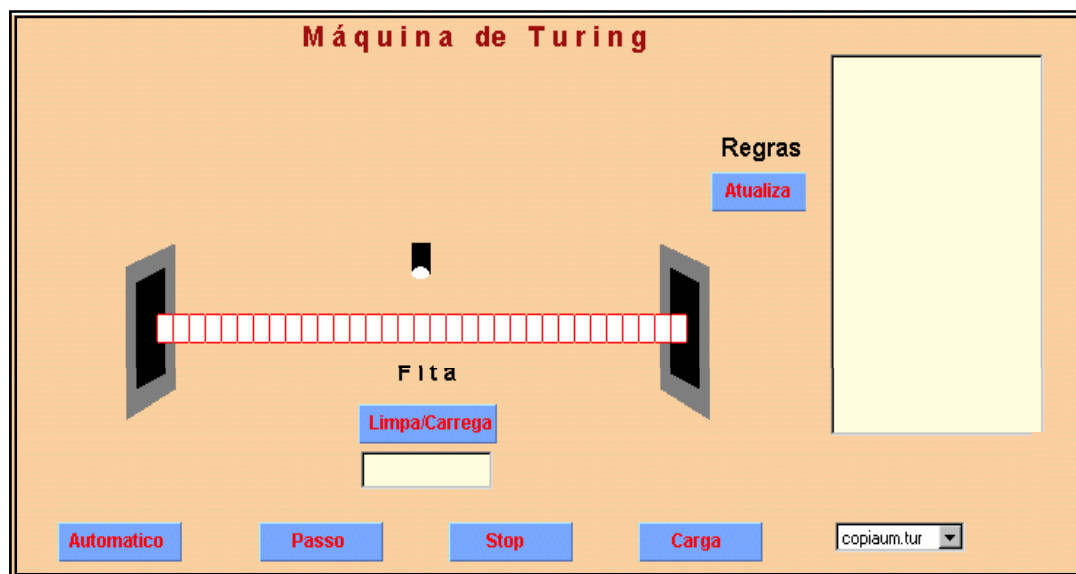


FIGURA 4.26 - Tela do software que simula a máquina de Turing

No anexo 16, também é apresentado um produto de software, que edita e simula a máquina de Turing; porém somente poderá ser utilizado, quando for abordado, com as crianças, o conceito de máquinas de estados finitos, que será trabalhado a seguir.

4.4 Máquina de estados finitos ou autômato finito

O estudo dos autômatos finitos permite projetarem-se vários tipos comuns de computador, algoritmos e programas.

Em [LEW 98], é descrito como um modelo de computação restrito, capaz de executar tarefas muito especializadas de manipulação de *string*, como dizer se determinada cadeia de caracteres aparece em um texto.

Segundo [HOP 79], é um modelo matemático de um sistema, com entradas e saídas discretas. Em [MEN 98], um *Sistema de Estados Finitos* é exemplificado, por meio do funcionamento de um elevador, onde cada estado possui as informações relativas ao *andar corrente* e *direção do próximo movimento*, e não é capaz de memorizar as informações anteriores.

O autômato finito é especificado formalmente, através do seu conjunto de estados, do conjunto dos terminais, do conjunto das produções construídas, do estado inicial do autômato e do conjunto de seus estados finais.

Em [HUN 87], autômato finito é descrito como um dispositivo, para se reconhecerem cadeias de uma dada linguagem. Possui um conjunto finito de estados, alguns dos quais são designados estados finais, e o controle passa de um estado para outro, à medida que cada caracter da cadeia é lido de acordo com um dado conjunto de transições. A cadeia será aceita, se, ao ler-se seu último símbolo, o autômato se encontrar em um dos estados finais; caso contrário, a cadeia não pertencerá à linguagem aceita pelo autômato.

Em [MOR 98], [SUD 98], [DAV 94] e [SIP 97] um autômato finito é descrito, mediante 5 elementos: (alfabeto, conjunto de estados, estado inicial, estado final, função programa). A função programa pode ser descrita por meio de uma tabela ou de um grafo.

Como em [MEN 98], neste trabalho, a representação da função programa será feita, através de um grafo, onde os estados são representados, mediante circunferências, e os arcos contêm o símbolo lido; e, de acordo com esse símbolo, o próximo estado a ser atingido exemplificado nas fig. 4.27 e 4.28.

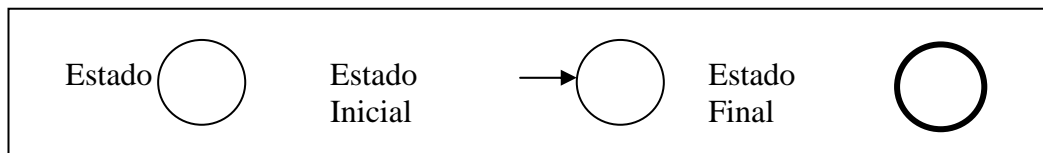


FIGURA 4.27 - Representação dos estados

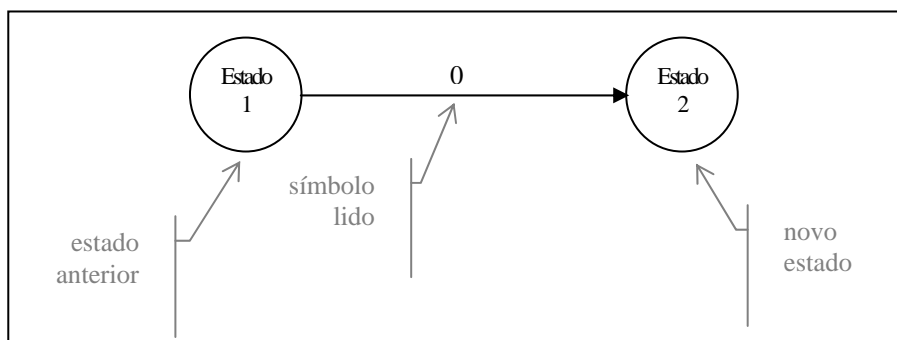


FIGURA 4.28 - Representação da função programa através de um grafo

A fig. 4.29 contém o exercício e o autômato que simulam os estados civis de uma pessoa.

Exemplo de construção de autômatos para definição dos estados civis de uma pessoa

obs.: - pode-se completar o exemplo
- descobrir quais são estados finais e iniciais

Estados (civis)

- solteiro(a)
- estar namorando
- noivo(a)
- casado(a)
- divorciado(a)
- viúvo(a).

As ações

- Encontrar um namorado(a)
- Marcar a data do casamento
- Casar
- Marido/esposa falecer
- Brigar com marido/esposa

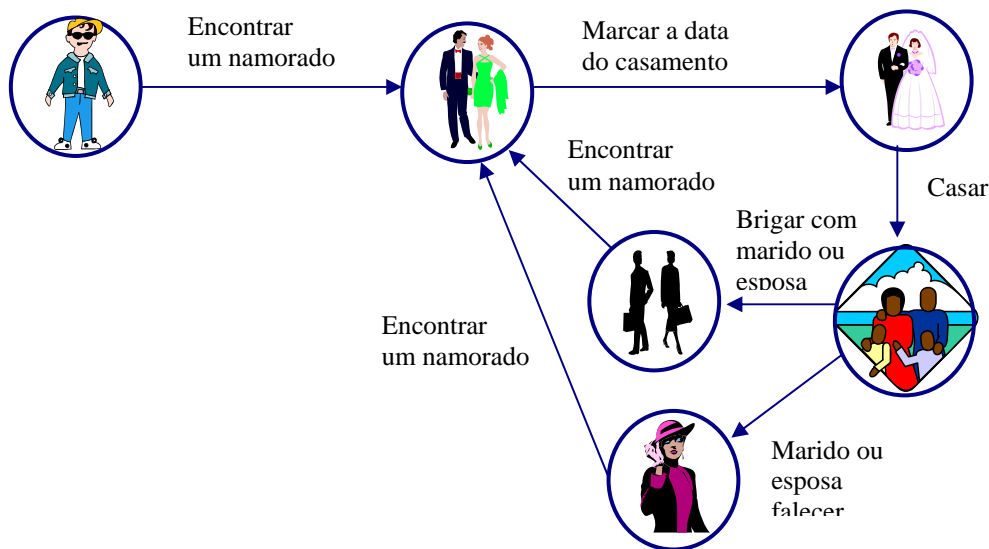


FIGURA 4.29 - Exemplo de exercício e solução de autômatos

O anexo 17 apresenta exercícios para construção de autômatos, que simulam uma máquina de vendas, o exemplo dado, anteriormente, do elevador e sugestões de exercícios para simulação das máquinas de lavar roupas e fax; e um exercício para conjugação de verbo baseado em [KOW 95].

4.5 Computabilidade e Complexidade de Problemas

Computabilidade se refere à busca de que tipo de problemas se conseguem resolver, utilizando-se um computador e, de acordo com [DIV 99], [KAI 72], [PIP 97] e [SAG 86], é um conceito fundamental no estudo de Ciência da Computação, como pode ser observado nas citações a seguir.

O objetivo do estudo da solucionabilidade de problemas é o de investigar a existência ou não de algoritmos que solucionem determinada classe de problemas. Ou seja, investigar os limites da computabilidade e, conseqüentemente, os limites do que pode efetivamente ser implementado em um computador. Em particular, o estudo da solucionabilidade objetiva evitar a pesquisa de soluções inexistente. [DIV 99] p. 165.

Modern digital computers are capable of solving a great many complex problems, often in a surprisingly short time. A natural question is whether there are any problems which are beyond the capabilities of mechanical problem-solving methods, even if given arbitrarily large amounts of

computing time. If any such intrinsically "unsolvable" problems exist, it is natural to wonder what their characteristics are, so we com avoid attempts to solve them. [JON 73] p. 1.

Podemos afirmar que, en general, todo problema computacional se reduce a la búsqueda de un procedimiento, que puede presentar-se como una fórmula, como un programa complejo o como un cierto modelo computacional. Todo lo que haga esa búsqueda más sistemática contribuye a facilitarla. [SAG 86] p. 13.

Segundo [MOR 98], a teoria da computabilidade pode ser estudado, mediante modelos universais de computação. Contudo, para esse autor, a melhor maneira seria por meio do estudo das funções recursivas.

Por este trabalho objetivar uma linguagem mais simples, clara, acessível às crianças, o conceito computabilidade é abordado, mediante uma série de exercícios, por meio dos quais, em suas resoluções, são utilizadas tentativas e erros. Essas tentativas podem conduzir às seguintes conclusões dos exercícios:

- não possuem solução;
- possuem solução.

Se as crianças conseguirem escrever, pelo menos, um algoritmo, que leve à solução do problema, então o problema será solucionável; caso contrário, não possuirá solução conhecida.

Desse modo, está-se introduzindo a noção do que é efetivamente solucionável, por meio de um computador, como na fig. 4.30, retirada de [DIV 99].

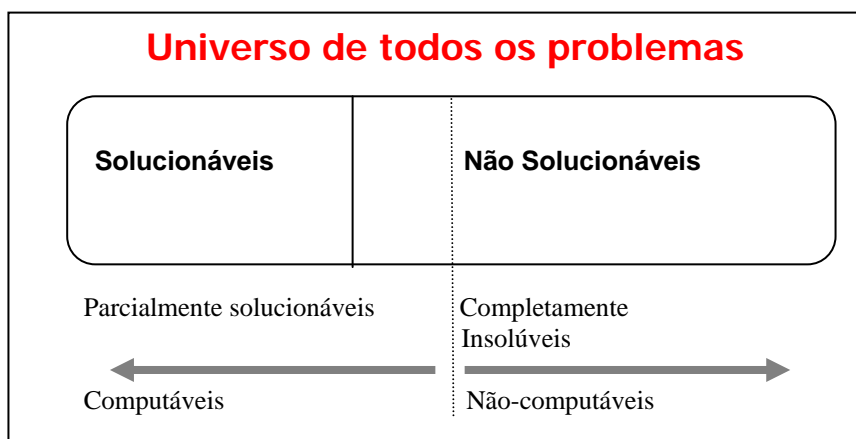


FIGURA 4.30- Relação entre classes de problemas [DIV 99], p. 169

As fig. 4.31, 4.32 e 4.33 contêm exercícios a serem propostos às crianças, alguns dos quais não possuem soluções, e são explicados, mediante a Teoria dos Grafos, [TOS 2000]. Porém, como dito anteriormente, neste trabalho, as soluções ocorrerão somente por meio de tentativas e erros. No anexo 18, encontra-se toda uma seqüência destes tipos de exercícios, inclusive o das "Pontes de Königsberg".

1. Problema do Assassinato do Bilionário Count Van Diamond.

O cenário abaixo é a residência do bilionário Count Van Diamond, que acaba de ser assassinado. Sherlock Gomes (um conhecido detetive, que, nas horas vagas,

é um estudioso da teoria dos grafos) foi chamado, para investigar o caso. O mordomo alega ter visto o jardineiro entrar na sala da piscina (lugar onde ocorreu o assassinato) e, logo em seguida, deixar aquela sala pela mesma porta por que havia entrado. O jardineiro, contudo, afirma que ele não poderia ser a pessoa vista pelo mordomo, pois havia entrado na casa, passado por todas as portas uma única vez e, em seguida, deixado a casa. Sherlock Gomes avaliou a planta da residência e, em poucos minutos, declarou solucionado o caso. Quem poderia ser o suspeito indicado por Sherlock Gomes? Qual o raciocínio utilizado pelo detetive, para apontar o suspeito?

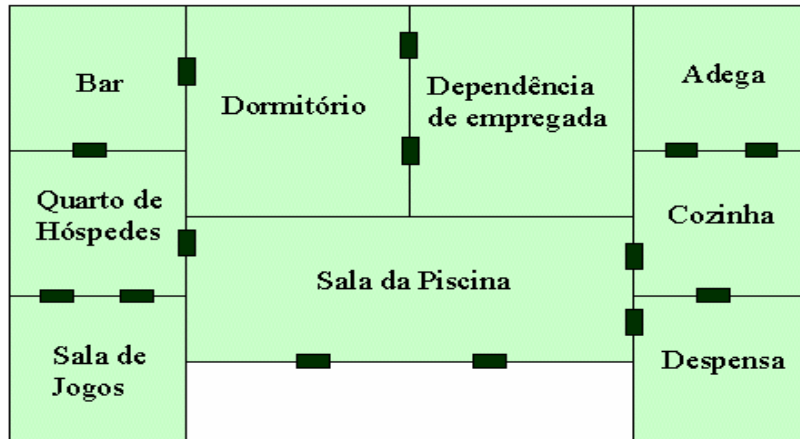


FIGURA 4.31 - Exercício sobre computabilidade "Problema do assassinato do bilionário Count Van Diamond"

2. Problema da ligação de serviços em 3 casas

Em uma rua 3 funcionários devem fazer as ligações das redes de água, luz e gás em uma casa. Decidem fazer o desenho das ligações no papel. Como nenhuma ligação pode cruzar com a outra, decidem primeiro fazer o desenho de como ficarão as ligações. Ajude os funcionários a fazer estas ligações.

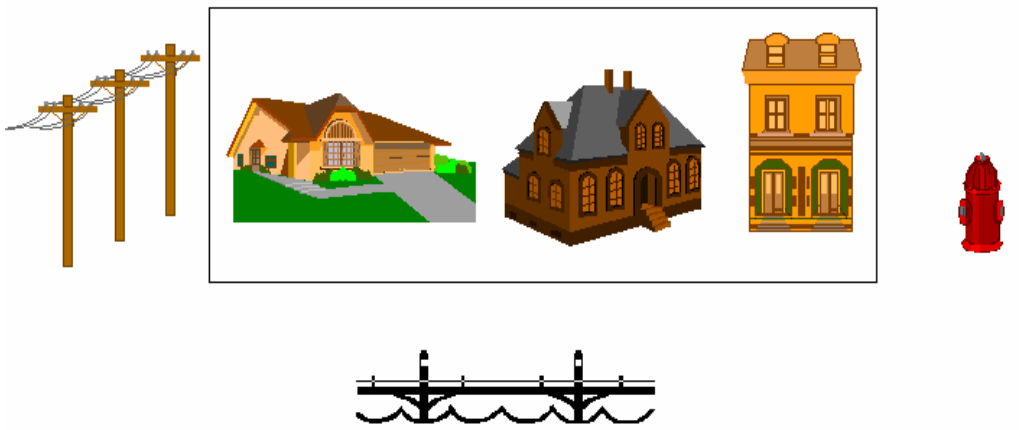


FIGURA 4.32 - Exercício sobre computabilidade "Problema da ligação de 3 serviços em 3 casas"

3. Problema do desenho da casa

No desenho ao lado, uma criança diz ter posto a ponta do lápis numa das bolinhas e, com movimentos contínuos (sem levantar e

sem retroceder o lápis) traçou as linhas que formam o desenho da casa, *traçando cada linha uma única vez*. A mãe da criança acha que ela trapaceou, pois não foi capaz de achar nenhuma seqüência que pudesse produzir tal resultado. Você concorda com essa mãe?

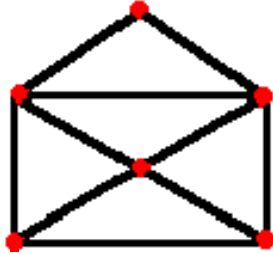


FIGURA 4.33 - Exercício sobre computabilidade "Problema do desenho da casa"

Os exercícios das fig. 4.31 e 4.32 não possuem solução, porém o exercício da fig. 4.33 a possui. Mas existem problemas para os quais existem soluções, todavia leva-se muito tempo, para encontrar a solução ideal, como na fig. 4.34.

Dada uma lista de números e um valor, deve-se encontrar com quais números da lista, se somados, se conseguirá o valor.

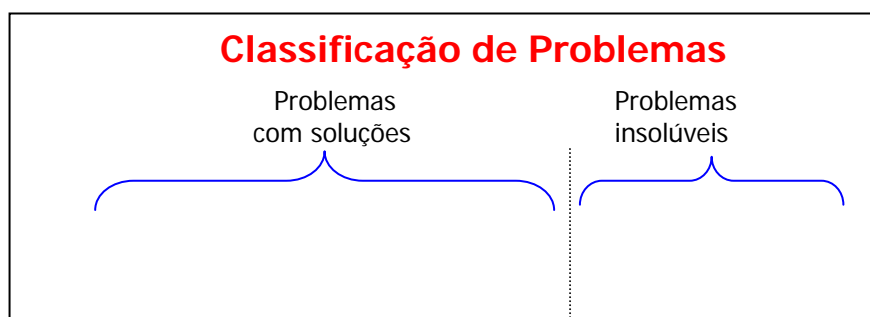
Exemplo:

Lista de números							
1029	625	890	23	694	214	1000	400

Valor
2584

FIGURA 4.34 - Exercício sobre computabilidade

A solução tradicional para este problema seria ir-se somando o primeiro número da lista com o segundo ou o segundo com o terceiro, ou ainda, o primeiro com o segundo e o terceiro e assim por diante, até se chegar à solução ideal. Entretanto, nessa solução, se houver aumento da quantidade de elementos na lista, as combinações possíveis entre os números aumenta consideravelmente. Portanto, o tempo que se leva para encontrar a solução aumenta muito, tendendo ao infinito. Assim, são identificadas outras classes de problemas ilustradas na fig. 4.35.



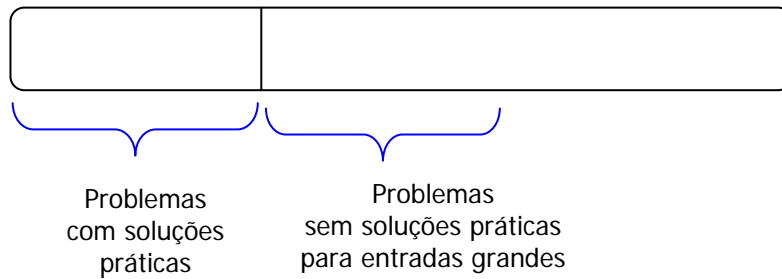


FIGURA 4.35 - Classificação de problemas

Na fig. 4.36, encontra-se um exercício descrito em [BRO 2000], mas que vários autores utilizam, para ilustrar problemas sem soluções práticas, que é o do “Caixeiro Viajante” em [BRO 2000], [TOS 2001], [NUN 2001] e [LEW 98].

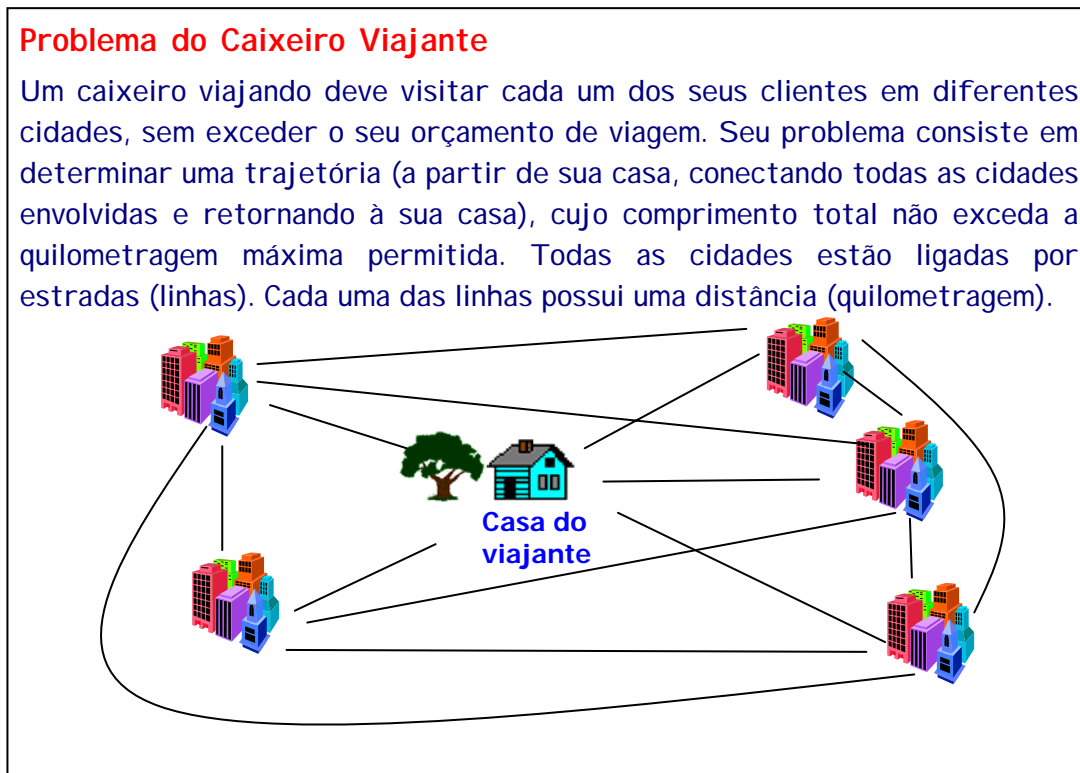


FIGURA 4.36 - Problema do Caixeiro Viajante em [BRO 2000], p. 439

Segundo [BRO 2000], a solução tradicional para este problema é considerar, de maneira sistemática, cada possível trajetória e comparar os seus comprimentos com o limite de quilometragem, até que seja encontrado algum caminho aceitável ou até que todas as possibilidades tenham sido consideradas.

Porém o autor destaca que, à medida que o número de cidades aumenta, o número de trajetórias que devem ser testadas cresce rapidamente. Desse modo, não é prático resolver-se o problema do caixeiro viajante dessa maneira, nos casos em que o número de cidades for grande.

Para resolver esse tipo de problema, [TOS 2001] e [BRO 2000] apresentam a noção de algoritmo não determinístico.

Um algoritmo não determinístico é um algoritmo que faz uso da função escolhe e das instruções falha e sucesso. [TOS 2001] p. 178.

Segundo [TOS 2001], o algoritmo não determinístico para o problema do Caixeiro Viajante ficaria da seguinte forma:

escolhe uma das trajetórias
 Se a quilometragem desta trajetória = quilometragem permitida
 Então sucesso
 Senão falha

Segundo [BRO 2000], o algoritmo não determinístico para o problema do Caixeiro Viajante ficaria da seguinte forma:

Escolher uma das possíveis trajetórias, e calcular sua quilometragem total
 Se esta quilometragem não excede a quilometragem permitida
 Então considerar encontrada uma solução
 Senão nada poderá ser concluído

Assim sendo, dentro da classe de problemas com soluções, porém sem soluções práticas, encontra-se uma subclasse de problemas, onde existe um algoritmo não determinístico, que a satisfaz, dependendo da escolha que se faça. Para esses tipos de algoritmos serem mais eficientes, a escolha deverá ser, cuidadosamente, selecionada, como no problema da Mochila, especificado na fig. 4.37.

Problema da Mochila

Considere uma mochila com capacidade limitada e diversos itens com pesos e valores conhecidos. O problema da mochila consiste em determinar-se um subconjunto, cujo peso total não exceda a capacidade da mochila e cujo valor total seja o maior possível.

Mochila
 Capacidade: 750,00



Itens, valores e pesos

Valores:	2,00	20,00	50.000,00	10,00	65.000,00	5,00	100,00
Pesos:	0,10	5,00	300,00	1,00	500,00	0,30	0,50



FIGURA 4.37 - Exercício sobre computabilidade "Problema da Mochila", [RAN 2001]

Com este tipo de problema, chega-se à classificação mais completa dos problemas, ilustrada na fig. 4.38.

Classificação de Problemas

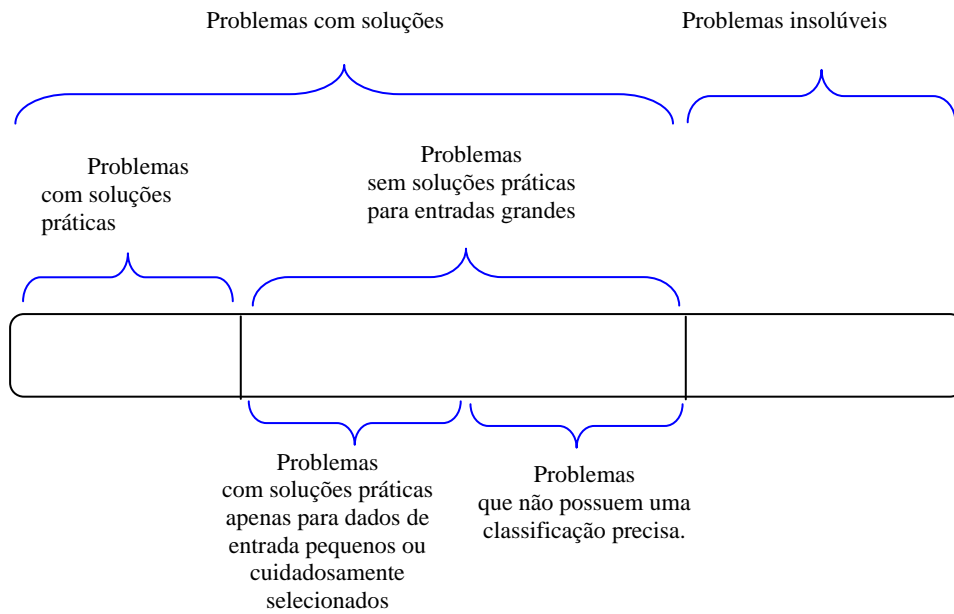


FIGURA 4.38 - Classificação de problemas

Deve ser ressaltado que os problemas insolúveis continuam inviáveis com ou sem o uso do computador. Pode-se também introduzir o termo “tratabilidade” para abordar problemas com soluções práticas ou sem solução prática conhecida.

5 Aplicação da Metodologia

A iniciativa de proceder-se ao ensino de Ciência da Computação para Crianças veio em resposta às necessidades de educação continuada e de atualização profissional,

superando questões de abandono e substituição de ferramentas, bem como técnicas dentro da Computação; ou seja, em Engenharia de Software, pode-se estudar algumas técnicas para o desenvolvimento de sistemas, porém não se pode garantir por quanto tempo a estará utilizando.

Como nas outras áreas de conhecimento, em Ciência da Computação, são necessárias constantes atualizações, re-qualificações e re-capacitações constantes.

Para que isso aconteça, acredita-se que não se pode limitar a inclusão da informática no processo educativo, apenas no ensino das tecnologias correntes, posto que essa inclusão deve visar os conceitos básicos envolvidos na Ciência da Computação.

Nos capítulos anteriores, procurou-se identificar alguns dos conceitos fundamentais da Ciência da Computação e defini-los, de conformidade com algumas disciplinas constantes nos atuais Parâmetros Curriculares [SBC 2002], tendo, como resultado, uma Metodologia do Ensino de Ciência da Computação para Crianças. Entretanto não se poderia afirmar que essa metodologia estaria de acordo com os interesses e com o grau de abstração das crianças.

Dessa feita, procedeu-se à realização de experiências em ambientes reais e com utilidade imediata. A primeira aplicação da metodologia ocorreu paralelamente a sua elaboração e a segunda aplicação, utilizando-se da metodologia já alterada, de acordo com os resultados da primeira aplicação.

5.1 Pré-teste

A primeira aplicação objetivou a adequação dos conceitos fundamentais encontrados em alguns livros tradicionais de Ciência da Computação, [DIV 99], [FOR 93], [GUI 85], [LUC 79], [BRO 2000] etc. e adequação dos exercícios, simultaneamente, a sua construção.

Algumas escolas particulares foram convidadas a fazer parte da pesquisa, porém foram colocadas diversas restrições a esse tipo de trabalho, pois essas escolas já têm direcionado o caminho da informática educacional que está detalhado no capítulo 2.

Algumas escolas públicas se prontificaram a participar dessa pesquisa, mas suas crianças, em geral carentes, vislumbram a informática como uma forma de capacitação profissional, de arrumar um emprego melhor para uma ascensão social em [FER 98]. Como não é esse o objetivo dessa pesquisa, a aplicação da metodologia com as crianças de escolas públicas foi abandonada.

Então, desenvolveu-se um trabalho com algumas crianças muito próximas, sobrinhos da mestrandia.

Nessa fase, participaram 3 crianças com 9, 10 e 11 anos de idade. Essas crianças foram escolhidas, por terem idades ideais sugeridas pela psicóloga que as auxiliou na pesquisa e que coincide com os seguintes estágios de desenvolvimento cognitivo apresentado por Piaget, [FAR 2002]. O estágio operatório-concreto compreende, aproximadamente, de 7 a 11 anos, quando a criança consegue dominar classes, relações e números, conseguindo abstrair dados da realidade; porém ainda depende do mundo concreto, para chegar à realidade. A partir dos 11 anos, estão no estágio operatório-formal, onde a representação permite a abstração total e, segundo [NIT 2001], são capazes de pensar em todas as relações possíveis logicamente, buscando soluções, partindo de hipóteses e não apenas da observação da realidade.

Outro fator que teve relação com a escolha do número de crianças foi a quantidade de microcomputadores disponíveis: somente 2. Esses microcomputadores

possuíam os recursos necessários para o desenvolvimento da pesquisa, como recursos de multimídia e internet.

A seguir, tem-se o detalhamento de cada um dos encontros realizados. Cada um deles não possuía nenhum tempo pré-determinado. Foram agendados de acordo com as atividades dos participantes. Onde se tem as siglas EMS, LMS e GT, são as abreviações dos nomes das crianças, siglas essas, e em algumas situações, utilizadas, com o intuito de se detalhar o comportamento de cada uma.

5.1.1. Primeiro e segundo encontros

O primeiro encontro visou ao esclarecimento das atividades a serem realizadas, seus objetivos, assim como a forma como seriam realizadas. Também houve a identificação de que uma das crianças possuía pouco conhecimento sobre a manipulação do computador. Por isso, o segundo encontro foi somente com essa criança.

O segundo encontro concentrou-se no reforço das atividades de manipulação do computador (mouse, teclado) e seus aplicativos, garantindo que todas as crianças possuíam as habilidades necessárias antes da realização do experimento.

Utilizaram-se as seguintes ferramentas: editor de texto *Microsoft Word*, navegador para internet *Internet Explorer*, Jogos *Simfarm* e *Show do Milhão*, anexo 19, pesquisas na Internet, por meio do *Internet Explorer* e o *Software de Treinamento Básico Sobre o Windows, Utilizando o Jogo de Representação-RPG*.

Com o editor de texto e o navegador, GT elaborou um trabalho para a escola sobre higiene.

No software *Simfarm*, as crianças devem administrar uma fazenda, construindo plantações e cuidando delas, fazem compras e vendas de colheitas etc.

O software *Show do Milhão* é baseado em um programa de televisão, onde são feitas perguntas sobre vários assuntos e o jogador deve respondê-las. Se as respostas forem corretas, irão ganhando prêmios.

O *Software de Treinamento Básico Sobre o Windows Utilizando o Jogo de Representação-RPG* contém a história de um cachorrinho que fora seqüestrado e, para conseguir salvá-lo, o aluno deve cumprir uma série de etapas, que se baseiam em treinamentos sobre o Windows, como o clique do mouse, abrir ou salvar um arquivo, etc. No início, o aluno escolhe seu personagem (menino ou menina) e que tipo de personalidade possuirá (número de pontos de inteligência, beleza, força e demais características). Quando erra uma tarefa, perde pontos e toma um caminho que o leva ao reforço da mesma. O cumprimento de todas as etapas leva-o ao objetivo do jogo, que é o salvamento do cachorrinho. A tela de abertura desse software se encontra na fig. 5.1 e as telas do jogo estão no anexo 20.



FIGURA 5.1 - Tela Inicial do Software de Treinamento Básico sobre o Windows utilizando o Jogo de Representações (RPG)

Esse encontro foi muito importante, pois como a criança participante já possuía conhecimento da manipulação do computador, adquiriu confiança, uma vez que sua insegurança decorria do fato de acreditar que as demais crianças conheciam mais ferramentas que ela.

5.1.2. Terceiro encontro

A partir desse encontro, foi utilizado o anexo 21, onde são anotados o número do encontro, data, horário, os exercícios utilizados e seus objetivos. Igualmente se observa cada uma das crianças individualmente, quais as atividades que ela executou e o que produziu.

Nesse encontro, o objetivo era trabalhar com o raciocínio lógico e aritmético, razão por que se utilizou o anexo 4, que contém vários exercícios abordando esse tema.

Todavia, antes do início do trabalho com os exercícios, foi feita uma pesquisa, na internet sobre o que significa lógica. Com esse conceito discutido, procedeu-se a uma nova pesquisa sobre, em especial Aristóteles, a forma discutida no item 4.1.1. Também se utilizou o livro "Mundo de Sofia" [JOS 95], que EMS havia lido em parte, procedendo a uma narração do que ela havia entendido. As crianças gostaram de conhecer as crenças de alguns dos filósofos e comentavam o que mudou atualmente. De igual forma se interessaram por saber como as crianças eram educadas antigamente.

Foram executados somente os dois primeiros exercícios do anexo 4. Esses exercícios manipulavam dinheiro, motivo pelo qual se procedeu da seguinte forma:

- trabalhos realizados individualmente por cada criança;
- ferramentas utilizadas: edição de figuras, Paint e navegador, Internet Explorer;

Iniciou-se, com o ensino da ferramenta Paint, com a finalidade de se poder desenhar e imprimir o dinheiro a ser manipulado. A tela inicial desse software se encontra no anexo 22.

EMS possuía um pouco de dificuldade na manipulação do mouse e por ser detalhista, foi a que mais demorou, para conseguir fazer o dinheiro.

Quando percebeu que estava atrasada em relação aos outros e estavam querendo imprimir o dinheiro, deixou como estava, mas não ficou satisfeita. Quando foi sugerido que se poderia pegar figuras de dinheiro da Internet, ela quis que o dela fosse dessa forma. Então houve uma pausa, para se trabalhar com pesquisa na *internet*.

Após o trabalho com pesquisas na página www.altavista.com.br, em imagens (palavras-chave: dinheiro, moeda e reais), acharam-se notas de R\$ 50,00, que utilizou e pediu, com muita insistência, para GT mudar o valor de seu dinheiro. Esta página se encontra ilustrada no anexo 22.

Como LMS manipula, muito bem, o mouse e quer começar, logo, a imprimir e a recortar o dinheiro, faz somente um retângulo com o valor do dinheiro, tenta desenhar um rosto, mas não o termina.

Como GT se sentou ao computador de LMS, copiou parte de seu desenho e continuou o desenho do rosto. Mas, quando EMS utilizou a figura do dinheiro, quis colocar parte da figura do rosto em seu desenho. A fig. 5.2 contém a produção desse encontro.

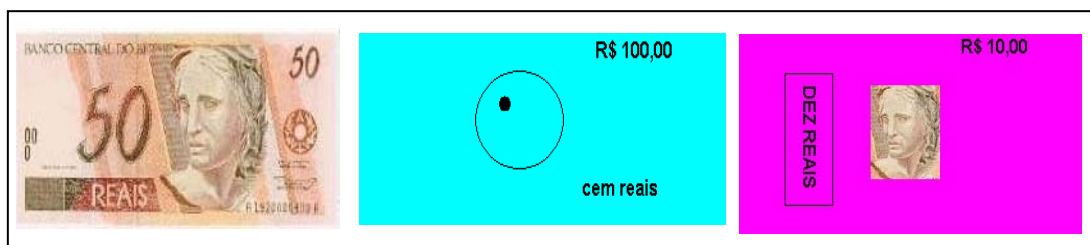


FIGURA 5.2 - Resultado do 3º. encontro realizado com as crianças

5.1.3. Quarto encontro

Nesse encontro, continuou-se com os exercícios de lógica e aritmética, constantes no anexo 4, porém até o exercício 9. O trabalho foi realizado em conjunto, pelas 3 crianças. Percebeu-se grande satisfação, por parte das crianças, pelo trabalho em grupo e também foram executados mais rapidamente. Apesar de EMS sempre liderar o grupo, não havia imposição por parte dela, e todos davam sugestões para a execução das tarefas.

Durante todos os exercícios, houve preocupação com não se conduzirem as atividades. Somente houve intervenção da mestrandia, quando solicitada pelas crianças.

O objetivo era que a execução das tarefas fosse feita de forma bem prática, visando à motivação das crianças. Assim, foram utilizados os seguintes materiais anteriormente preparados:

- 10 caixas de papelão de tamanhos diversos;
- baldes com as capacidades de 3, 4, 5, 7 e 9 litros;
- 4 relógios;
- 6 barbantes, 3 simulando as medidas de 100 metros e 3 com 50 metros;
- atlas;
- folhas quadriculadas;
- lápis de cor e papel;
- desenhos de pessoas.

Os exercícios em que foram utilizados os baldes, que deveriam ser enchidos de água, até conseguir-se a quantidade ideal, foram os de que as crianças mais gostaram.

No exercício em que foi utilizado o barbante, trabalhou-se com os conceitos de escalas e proporções. Para isso, utilizou-se um atlas e folhas quadriculadas, onde foram feitos desenhos. Nesse mesmo, exercício aproveitou-se para introduzir o conceito de primitivas, utilizando, como primitivas, o andar para a frente, virar à direita ou à esquerda.

Foram feitos até o exercício 9 do anexo 4, pois nesse momento, foi sugerido se as crianças queriam parar, o que foi aceito e foram realizar o *Jogo do Milhão*, anexo 19.

Como produção desse encontro foram as respostas corretas dos exercícios. Houve a possibilidade de que alguns exercícios fossem escritos em forma de algoritmo como na fig. 5.3. Nesse momento procurou-se orientar a forma como as crianças deveriam escrever esses algoritmos. Também foi pedido que resolvessem os exercícios das fig. 4.2 e 4.3 em suas casas.

5.1.4. Quinto encontro

Foram terminados e/ou corrigidos todos os exercícios que estavam faltando do anexo 4 e também os exercícios de lógica, ilustrados nas fig. 4.2 e 4.3. O trabalho continuou sendo feito em conjunto pelas 3 crianças e foram utilizados os seguintes materiais:

- papel, régua, lápis e fósforos;
- dinheiro já montado anteriormente, faltando imprimir;
- software *Microsoft Word* para montagem de tabela;
- desenhos de pessoas representando os pais e avós.

Para o exercício 7 (*A soma de 2 números pares consecutivos é 22. Eles são?*), foi trabalhado o conceito de número par e números consecutivos. Todavia as crianças já sabiam o que significava. Para chegarem à solução desse exercício, as crianças começaram somando, 2 e 4. Após, passaram para 6 e 8, quando foi sugerido que poderiam somar 4 e 6 também. E assim seguiram, até que chegaram aos valores finais 10 e 12.

O exercício em que mais se demoraram foi o de número 6 da fig. 4.3, em que era necessário que se descobrissem os proprietários de determinadas casas, qual o animal de estimação de cada uma das casas etc. Mas gostaram muito desse tipo de exercício, de cujo tipo foram escolhidos mais 3, que as crianças levaram para fazer separadamente, em suas casas.

5.1.5. Sexto e sétimo encontros

Nesses encontros, começou-se a trabalhar com os exercícios sobre lógica símbolo imagem constantes no anexo 5. Foram utilizados os seguintes materiais:

- folhas impressas, contendo os exercícios;
- letras impressas: A, C, S e M;
- desenhos das figuras sol, árvore, maçã e casa.

O trabalho foi desenvolvido individualmente, entretanto, quando alguma criança possuía alguma dúvida, procurava esclarecê-la com as outras crianças ou com a mestrandia. Apesar de esses encontros terem transcorrido normalmente, as crianças aparentaram certo cansaço, havendo necessidade de se alternar a resolução dos exercícios com outras atividades, envolvendo jogos no computador. Produziram-se as folhas impressas dos exercícios contendo as soluções. Como resultados desses encontros, um exemplo se encontra na fig. 5.4.

A cada final de encontro e a cada novo encontro, é realizado um diálogo entre todos os participantes, com o objetivo de:

- conhecer-se a opinião das crianças sobre o que elas estão achando dos encontros, o que estão gostando e o que não gostam;
- discutir-se o tópico abordado e sua importância dentro de todo o trabalho;
- detectarem-se as falhas de aprendizado dos conceitos já abordados.

Foi detectado que LMS teve mais dificuldade que as outras crianças nos exercícios finais do anexo abordado nesse encontro. Dessa feita, ao final desse encontro, procedeu-se a uma revisão dos exercícios dos quais participaram todas as crianças, para que os conceitos de negação, conjunção e disjunção ficassem claros.

Exercícios resolvidos - Anexo 5

$\left. \begin{array}{l} A \wedge \neg M ? \\ A \vee \neg M ? \end{array} \right\} \emptyset$	\nrightarrow \rightarrow
$\left. \begin{array}{l} \neg M \vee \neg C ? \\ \neg M \wedge \neg C ? \end{array} \right\} \begin{array}{c} \text{árvore} \quad \text{sol} \end{array}$	\rightarrow \rightarrow
$\neg M \vee \neg C \left\{ \begin{array}{l} ? \quad \text{árvore} \\ ? \quad \text{maçã} \\ ? \quad \text{casa} \end{array} \right.$	\rightarrow \nrightarrow \nrightarrow
<p>Exemplo: $\neg M \wedge \neg C \nrightarrow$ </p>	
$\neg M \wedge \neg C \left\{ \begin{array}{l} ? \quad \text{árvore} \\ ? \quad \text{maçã} \\ \text{casa} \end{array} \right.$	\rightarrow \nrightarrow

FIGURA 5.4 - Transcrição dos exercícios finais do anexo 5 resolvidos pelas crianças

5.1.6. Oitavo encontro

Esse encontro foi muito importante, posto que foi feita uma introdução ao significado de algoritmo e novamente tratado o conceito de primitiva. Para tanto, utilizou-se de receitas de sanduíches, de bombons e do anexo 6, que possui algoritmos na forma de desenhos para construção de um pássaro e um balão de dobradura.

Primeiro, foram apresentados os conceitos de algoritmo e feitos os sanduíches e bombons com demonstração do que seria a execução de um algoritmo e de quanto seria necessário o seu detalhamento, como, por exemplo, até onde cada uma estava entendendo cada linha da receita.

O pássaro de dobradura nenhuma das crianças conseguiu construir. Já o balão todas conseguiram com alguma ajuda, para interpretar as primitivas. Os materiais utilizados foram os seguintes:

- folhas de papel para dobraduras;
- papel e lápis.

Utilizando o papel e o lápis, as crianças tentaram construir seu próprio algoritmo, partindo das primitivas definidas anteriormente, para construção de um copo, navio e chapéu de dobradura. Para 2 crianças, houve necessidade de que elas criassem algumas primitivas, que acharam que estavam faltando.

Os novos algoritmos produzidos foram trocados entre as crianças, a fim de que elas entendessem a diferença entre construir e executar um algoritmo. Houve um pequeno tumulto, quando uma das crianças não conseguiu executar o algoritmo da outra, sendo sugerido pelas crianças que os algoritmos deveriam ser filmados ou fotografados em cada uma das etapas.

As crianças elaboraram os algoritmos utilizando papel e lápis, mas a fig. 5.5 contém um dos algoritmos transcritos no computador pela mestranda.

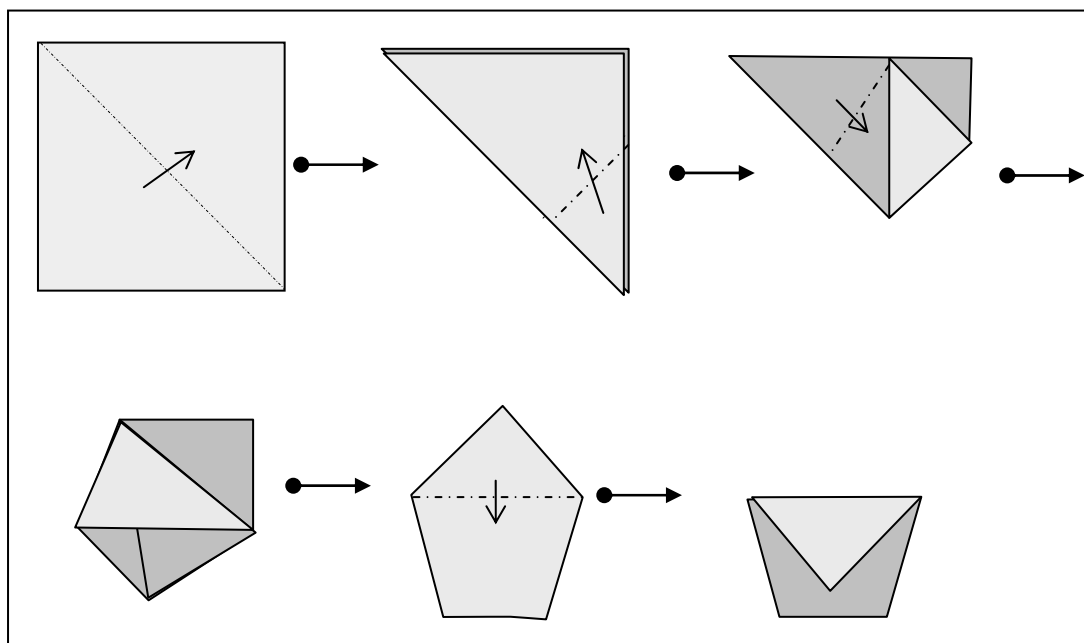


FIGURA 5.5 – Algoritmo produzido por EMS e GT para construção de um copo em dobradura.

5.1.7. Nono encontro

Nesse encontro, uma das crianças trouxe mais um equipamento, totalizando 3. Elas utilizaram somente os microcomputadores e as ferramentas *Microsoft Word* e *Microsoft Paint*, bem como o anexo 9, que consta de um tutor, para a construção de papel de carta personalizado.

O trabalho foi iniciado separadamente, mas, depois, LMS sentou-se junto com EMS e, após algumas tentativas “fracassadas”, GT sentou-se junto com outras 2. O exercício foi realizado com muita conversa e cooperação entre ambos, porém conseguiram-se 3 produções diferentes, uma para cada criança.

O tutor foi utilizado para exemplificar outra forma de se escrever algoritmo e foi proposto que as crianças identificassem quais seriam as primitivas para esse algoritmo. Nesse instante, houve muita confusão das primitivas genéricas e específicas e procurou-se esclarecer as diferenças. O resultado dessa divisão se encontra na tabela 5.1.

TABELA 5.1 - Primitivas identificadas pelas crianças para o anexo 9

Primitivas genéricas	Primitivas específicas
clique do mouse <qual botão>	clique do mouse <com o botão esquerdo>
abrir a ferramenta <qual>	abrir a ferramenta <word>
copiar e colar <qual desenho>	
escrever <qual texto>	
utilizar marcadores e numeração	

5.1.8. Décimo e décimo primeiro encontros

Nesses encontros, continuou-se trabalhando o conceito de algoritmos e foi utilizado o anexo 7, que contém exercícios para sua construção. O trabalho foi realizado por 2 crianças em um equipamento e 1 em outro utilizando a ferramenta *Microsoft Word*, para escreverem seus algoritmos, que, após a impressão de cada algoritmo construído pelas crianças, era executado por outra, visando a testar se os algoritmos estavam corretos. Esse trabalho transcorreu calmamente e EMS achou que executar um algoritmo era igual seguir um mapa. Nesse encontro, pode ser percebido que as crianças sabiam, claramente, a diferença entre algoritmo e primitivas, assim como o que significava construir e executar um algoritmo.

Para o exercício 2, utilizaram-se, na execução do algoritmo, figuras representando rio, índios e canibais. No exercício 3, o trabalho foi feito em conjunto pelas 3 crianças, sendo que 2 jogavam e a outra anotava. Pela primeira vez, foram utilizadas estruturas de seleção, repetição e as crianças foram orientadas a usar indentações tendo em vista organizar melhor os algoritmos.

Também, como novidade, no exercício 3, foi sugerido às crianças que deixassem o algoritmo genérico, ou seja, cada vez que o algoritmo fosse executado seria escolhido um número do quadrado. Mesmo assim, as crianças puderam perceber que não houve tática de defesa e que havia a necessidade de não executar determinada primitiva em algum momento do algoritmo. Com essa constatação, pôde ser introduzida a estrutura de seleção que continha o senão e que seria trabalhada no próximo encontro. As produções desses encontros estão nas fig. 5.6, 5.7 e 5.8.

Algoritmo para exercício 1 do anexo 7 - Robô que troca a lâmpada
<ul style="list-style-type: none"> • desloque-se para <onde tem que trocar a lâmpada> • mova <mão> para <lâmpada velha> • pressione <lâmpada velha> • gire garras 180° no sentido anti-horário • mova <lâmpada velha> para <lixo> • desloque-se para <onde está a lâmpada nova> • mova <mão> para <lâmpada nova> • pegue <lâmpada nova> • desloque-se para <onde tem que trocar a lâmpada> • mova <lâmpada nova> para <lugar da troca> • gire garras 180° no sentido horário

FIGURA 5.6 - Um dos algoritmos produzidos pelas crianças para o exercício 1 do anexo 7

Algoritmo para exercício 2 do anexo 7 – Jesuítas e canibais atravessando o rio

- Canibal entra no barco
- Canibal entra no barco
- Barco atravessa o rio da margem direita para a esquerda
- Canibal sai do barco
- Barco atravessa o rio da margem esquerda para a direita
- Canibal entra no barco
- Barco atravessa o rio da margem direita para a esquerda
- Canibal sai do barco
- Barco atravessa o rio da margem esquerda para a direita
- Canibal sai do barco
- Jesuíta entra no barco
- Jesuíta entra no barco
- Barco atravessa rio da margem direita para a esquerda
- Jesuíta sai do barco
- Canibal entra no barco
- Barco atravessa rio da margem esquerda para a direita
- Canibal sai do barco
- Jesuíta entra no barco
- Barco atravessa o rio da margem direita para a esquerda
- Jesuíta sai do barco
- Jesuíta sai do barco
- Canibal entra no barco
- Barco atravessa o rio da margem esquerda para a direita
- Canibal entra no barco
- Barco atravessa o rio da margem direita para a esquerda
- Canibal sai do barco
- Barco atravessa o rio da margem esquerda para a direita
- Canibal entra no barco
- Barco atravessa o rio da margem direita para a esquerda
- Canibal sai do barco
- Canibal sai do barco

FIGURA 5.7 - Um dos algoritmos produzidos pelas crianças para o exercício 2 do anexo 7

Algoritmo para exercício 3 do anexo 7 - Jogo da Velha

- Enquanto todos os quadrados não estiverem ocupados e X ou O não tiverem marcado três quadrados em linha
 - Escolha o <número do quadrado>
 - Se o quadrado não estiver ocupado, marque o quadrado <número do quadrado> com um <X>
 - Se O tem dois quadrados quaisquer em linha, marque, no terceiro, desocupado
 - Escolha o <número do quadrado>
 - Se o quadrado não estiver ocupado, marque o quadrado <número do quadrado> com um <O>
 - Se X tem dois quadrados quaisquer em linha marque, no terceiro, desocupado

FIGURA 5.8 - Um dos algoritmos produzidos pelas crianças para o exercício 3 do anexo 7

5.1.9. Décimo segundo e décimo terceiro encontros

Infelizmente, esses foram os últimos encontros com as crianças, devido a dois motivos: o semestre acabou e não houve horários compatíveis entre os participantes; e a metodologia aqui elaborada será aplicada em um ambiente mais real, que será detalhado adiante, no item 5.2.

Esses encontros concentraram-se na identificação de primitivas e na construção e execução de algoritmos, utilizando o anexo 8, até o 3º. exercício. Para a resolução desses exercícios, as crianças se basearam, sempre, no que tinham visto anteriormente. Percebe-se isso na produção desse encontro, fig. 5.9 e 5.10.

Resolução do exercício 1 do anexo 8 – Torre de Hanói	
Primitiva	Algoritmo
<ul style="list-style-type: none"> • mova disco <número do disco> de pino <letra do pino> para pino <letra do pino> 	<ul style="list-style-type: none"> • mova disco <1> de pino <A> para pino <C> • mova disco <2> de pino <A> para pino • mova disco <1> de pino <C> para pino • mova disco <3> de pino <A> para pino <C> • mova disco <1> de pino para pino <A> • mova disco <2> de pino para pino <C> • mova disco <1> de pino <A> para pino <C>

FIGURA 5.9 - Resolução pelas crianças do exercício 1 do anexo 8

Resolução do exercício 2 do anexo 8 - Barco que atravessa o rio com 3 cargas	
Primitiva	Algoritmo
<ul style="list-style-type: none"> • <nome da carga> entra no barco • <nome da carga> sai no barco • barco atravessa a margem da direita para esquerda • barco atravessa a margem da esquerda para direita 	<ul style="list-style-type: none"> • <ovelha> entra no barco • barco atravessa a margem da direita para a esquerda • <ovelha> sai no barco • barco atravessa a margem da esquerda para a direita • <maço de alfafa> entra no barco • barco atravessa a margem da direita para a direita • <maço de alfafa> sai do barco • <ovelha> entra no barco • barco atravessa a margem da esquerda para a direita • <ovelha> sai do barco • <lobo> entra no barco • barco atravessa a margem da direita para a esquerda • <lobo> sai do barco • barco atravessa a margem da esquerda para a direita • <ovelha> entra no barco • barco atravessa a margem da direita para a esquerda • <ovelha> sai do barco

FIGURA 5.10 - Resolução pelas crianças do exercício 2 do anexo 8

No exercício 3, fig. 11, 12 e 13, surgiram muitas dúvidas e puderam ser trabalhados muitos conceitos:

- crianças não sabiam se deveriam deixar genérico, e pôde ser discutido, novamente, que um algoritmo deve abranger todos os casos possíveis;
- na fig. 5.13, a criança não sabia se deveria colocar a primitiva "se operação for divisão então passar para multiplicação". Em consequência disto, foi discutido que a constante a poderia ser igual a $\frac{1}{2}$, então chegando à conclusão de que deveria ser deixada essa primitiva;
- deveriam escrever *sinal* ou *operação* e se deveria constar o cálculo da operação.

1º. tipo de primitivas para o exercício 3 do anexo 8 – Equação do 1º grau

- passar <letra da constante> da direita para a esquerda
- se sinal da constante for <sinal> então passar para <sinal>

FIGURA 5.11 - Primitiva construída pela 1ª. criança para o exercício 3 do anexo 8

2º. tipo de primitivas para o exercício 3 do anexo 8 – Equação do 1º grau

- passar a constante b para o lado esquerdo
- passar a constante a para o lado esquerdo
- trocar operação

FIGURA 5.12 - Primitiva construída pela 2ª. criança para o exercício 3 do anexo 8

3º. tipo de primitivas para o exercício 3 do anexo 8 – Equação do 1º grau

- passar o valor de b do lado direito para o lado esquerdo
- passar o valor de a do lado direito para o lado esquerdo
- se operação da constante for *subtração* então passar para *adição*
- se operação da constante for *adição* então passar para *subtração*
- se operação da constante for *multiplicação* então passar para *divisão*
- se operação da constante for *divisão* então passar para *multiplicação*

FIGURA 5.13 - Primitiva construída pela 3ª. criança para o exercício 3 do anexo 8

Foi esclarecido às crianças que todas essas dúvidas seriam elucidadas quando entrassem no próximo item: programas. Essa explicação foi dada, por quanto assim se consegue chegar a um algoritmo genérico (com a leitura das constantes) e os cálculos irão constar nas operações aritméticas. Como ilustração para as crianças, foi mostrada a fig. 5.14.

Algoritmo para exercício 3 do anexo 8 – Equação do 1º grau

Início

Ler o valor da constante a

Ler o valor da constante b

Calcular $X \leftarrow (-1 * b) / a$

Exibir o valor de X

Fim

FIGURA 5.14 - Algoritmo para o exercício 3 do anexo 8

5.2 Aplicação da metodologia ao Colégio de Aplicação da UFRGS

No item anterior, procurou-se mostrar como se procedeu à aplicação da metodologia para o ensino de Ciência da Computação para Crianças. Essa aplicação ocorreu simultaneamente a sua elaboração e, por isso, cada seleção ou criação dos próximos exercícios teve, como base, o andamento do encontro anterior.

Como pôde ser observado através da descrição do comportamento e, mais ainda, através de suas produções, a metodologia revelou-se adequada para aquelas crianças. Entretanto não se podia garantir que, em um ambiente real de aprendizado, o comportamento das crianças seria o mesmo. Dessa forma, será essencial avaliar-se posteriormente, a metodologia. Na seqüência desse trabalho, serão explicados como se procedeu a uma outra aplicação dessa metodologia já pré-avaliada.

A partir disso, alguns questionamentos deram uma direção a esta pesquisa: Qual a diferença de comportamento/aprendizado das crianças participantes da primeira avaliação com a de crianças pertencentes a uma sala de aula tradicional? A faixa de idade sugerida para a aplicação da metodologia é a indicada?

Esses questionamentos são, agora, objeto de análise, por meio de uma nova aplicação da metodologia no Colégio de Aplicação da Universidade Federal do Rio Grande do Sul. No entanto, nessa aplicação, até esse momento, houve somente 3 aulas, devido à greve dos professores das universidades federais.

5.2.1 O Colégio de Aplicação da UFRGS e o Projeto Amora

A seguir, são apresentados o histórico e o funcionamento do Colégio de Aplicação, especialmente sobre o projeto Amora. Esses dados foram retirados da página do Colégio de Aplicação da UFRGS em [CAP 2001].

5.2.1.1 Histórico do Colégio de Aplicação da UFRGS

Criado por decreto de 1946, o Colégio de Aplicação (CAp) da UFRGS começou a funcionar no dia 14 de abril de 1954. Os primeiros professores eram os mesmos que davam aula na antiga Faculdade de Filosofia, na época também responsável pela formação de todos os alunos que também queriam ser professores primários ou secundaristas. Aquela Faculdade de Filosofia englobava cursos (como os de Pedagogia, História, Letras etc.) que, hoje, estão desvinculados da atual Filosofia, cada qual constituindo faculdades ou institutos autônomos.

A primeira seleção de alunos não foi realizada, porque o prazo já não era possível, segundo os critérios da rede de ensino da época. Frente à necessidade de se iniciarem as atividades, a solução para formar-se o primeiro corpo discente do CAp foi recorrer aos suplentes dos exames de seleções de alunos, realizadas pelo Instituto Estadual de Educação Flores da Cunha (o Instituto) e pela Escola Estadual Júlio de Castilhos (o Julinho). A aproximação só foi possível, visto que a Universidade era estadual. (A federalização ocorreu no início dos anos setenta). Durante os primeiros anos do CAp só havia curso ginásial (mais ou menos o equivalente às 6^a, 7^a e 8^a séries).

Em 1958, o colégio obteve licença para oferecer o curso científico. Em geral, era uma turma de alunos por série, com, mais ou menos, 25 alunos por turma. Em 1977, o colégio passou a ter alunos de 5^a série do 1^o grau. Somente em 1979, começou a receber alunos para a 1^o série do 1^a grau, como parte de um projeto de pesquisa que também criou o Gabinete Pedagógico para acompanhá-la. O projeto denominou as séries iniciais de Alfas e, assim, são conhecidas, até hoje, as turmas de

alunos até o 4º. Na mesma época, as séries de 2ª grau passaram a ser, regularmente, em número de duas turmas de alunos. Outras séries do 1ª grau, a partir da 5ª, também eram constituídas de duas turmas de alunos, sem regularidade entretanto. No início dos anos oitenta, o sistema de seleção (com provas escritas e entrevistas) foi substituído pelo sorteio público. Até hoje, essa é a modalidade de ingresso de novos alunos para o CAp.

5.2.1.2 Apresentação do Projeto Amora

No atual processo de formação de uma nova cultura planetária, constata-se transformações vertiginosas em relação a todos os campos. Essas transformações ocorreram no contexto de mudanças de paradigmas, que exigem repensar em se as relações natureza/ sociedade/ cultura, as quais afetam o mundo do trabalho e do conhecimento. A educação constitui-se num espaço propício para a efetivação desse repensar. Nessa perspectiva, o Colégio de Aplicação da UFRGS (CAp), como escola de Ensino Fundamental e Médio integrada a uma Universidade, vem trabalhando, desde sua fundação, há quarenta e cinco anos, a fim de afirmar-se como centro de referência educacional, gerador de experiências inovadoras, prioritariamente voltadas para a melhoria da ação pedagógica desenvolvida nas escolas da rede pública. Em função disso, desenvolve ações de ensino, pesquisa e extensão que propõem a construção de propostas pedagógicas diversas das implementadas nas redes regulares de ensino, contribuindo para a formação de professores (via estágios regulares e avançados), em nível de estágios e para a efetivação da educação continuada de professores em serviço, em consonância com as atuais exigências e necessidades de desenvolvimento social do país.

O Colégio de Aplicação tem procurado fazer de sua práxis pedagógica cotidiana um campo de investigação e reflexão permanentes, de onde emergem elementos úteis às trocas e à fertilização de novas práticas. Segundo Fazenda (1994, p.49), é em instituições (como o CAp) que incentivam os professores e propiciam infraestrutura necessária para o seu desenvolvimento que se encontra o "germe de projetos interdisciplinares de ensino, cuja a tônica é o diálogo e a marca, o encontro, a reciprocidade".

O Projeto Amora reúne tais características, aceitando os desafios que se originam dos processos de mudança, sobretudo aqueles que se colocam em duas instâncias:

a) nas relações que se verificam no ambiente escolar, entre professor/aluno, aluno/aluno, professor/ professor e professor/aluno/conhecimento, com vistas ao desenvolvimento de relações sociais, comprometidas não só com as essências singulares mas também com o coletivo;

b) nos conceitos de espaço e tempo, posto que há uma produção de conhecimento em progressão geométrica, num cenário em que as novas tecnologias são vistas não só como recurso poderoso na geração e manejo de informações mas principalmente como recurso original para a geração de novos conhecimentos e trocas, viabilizador de transformações substanciais na comunicação entre os homens.

O Projeto em desenvolvimento, desde o final de 1995, iniciou-se com uma turma de 5ª. Série. Sua expansão tanto para outros níveis da escola quanto para outras instituições dependerá da realidade das mesmas, no que se refere à faixa etária, disponibilidade e interesse de professores em construir uma nova proposta pedagógica.

À medida que forem sendo desenvolvidas ações compartilhadas entre diferentes parceiros, presencialmente e a distância, pressupõe-se o crescimento de propostas alternativas ao ensino tradicional. Em 1996, o Projeto Amora, teve como

parceiros iniciais, alunos e professores de escolas públicas estaduais e municipais dos bairros Restinga e Agronomia, limítrofes do Campus do Vale, na cidade de Porto Alegre/RS, onde o CAP está situado. Tais escolas, à época, faziam parte do subprojeto "Sociedade do Conhecimento", que, por sua vez, integrava o Projeto Porto Alegre-Tecnópole (UFRGS/SE-RS/SMED-POA). Nesse mesmo ano, o Projeto Amora ampliou sua parceria com o Laboratório de Estudos Cognitivos do Instituto de Psicologia - LEC/UFRGS - e passou a integrar o Projeto Educação a Distância, em Ciência e Tecnologia - EducaDi/ CNPq², coordenado, nacionalmente, pela Prof.a. Dra. Léa da Cruz Fagundes (LEC/UFRGS). O EducaDi envolveu quatro núcleos (Rio Grande do Sul, São Paulo, Distrito Federal e Ceará), nos quais escolas das redes públicas estadual e municipal, interconectadas por Internet, desenvolvem projetos cooperativos à distância. Nesse projeto, o Amora atuou como campo prioritário de investigação e tessagem reflexiva de metodologias que incorporam as Novas Tecnologias de Informação e Comunicação (NTICs)³.

As oficinas fazem parte das atividades do Projeto Amora e têm como objetivo desenvolver habilidades de pensamento. Essas oficinas são formadas por grupos de 15 a 20 alunos, com diferentes estilos cognitivos e tempos de aprendizagem geralmente indicados pelo grupo de professores. Nela, esses alunos desenvolvem atividades específicas, tais como produção textual, histórias em quadrinhos, culinária, jogos pedagógicos etc.

5.2.2 Detalhamento das aulas da Oficina de Ciência da Computação ao Colégio de Aplicação da UFRGS

De acordo com as características inovadoras do Colégio de Aplicação e a faixa de idade do Projeto Amora coincidir com essa pesquisa, foi criada uma oficina para 15 crianças e as aulas foram coordenadas por bolsistas da Iniciação Científica (PIBIC-Cnpq) - UFRGS. Por isso foram montados os detalhamentos sugeridos para cada aula, especificados no anexo 23.

Para a criação das oficinas, o Colégio de Aplicação procede da seguinte forma: são apresentados aos alunos os temas de cada oficina e os horários em que funcionarão escolhendo os alunos as que lhes interessam.

Na apresentação da oficina do ensino de Ciência da Computação, procurou-se deixar claro que o objetivo da oficina não era a manipulação do computador ou o ensino de algum software específico, uma vez que o ensino de Ciência da Computação não é baseado em uma máquina real.

Onze crianças se inscreveram com idade entre 11 e 12 anos. Todas elas eram do sexo masculino; e, assim, no detalhamento das aulas, referenciar-se-ão como meninos. Para as aulas, foram utilizadas duas salas, uma com computadores e outra sem.

5.2.2.1 Primeira aula

Na primeira aula, foi utilizada a sala contendo computadores e realizada uma apresentação da oficina. Também foram passadas as fichas, para os alunos preencherem, "*o que gostariam de aprender sobre o computador e Ciência da Computação*", anexo 3.

² O Projeto Educadi foi concluído em dezembro de 1998.

³ Alguns resultados são apresentados no site do Projeto Amora <http://www.cap.ufrgs.br/~amora>.

As respostas dos meninos foram colocadas no quadro, a fim de serem feitas comparações sobre o que queriam aprender e relacioná-las com Computador e Ciência da Computação. A maioria das respostas sobre o que gostariam de aprender eram alguns aplicativos e esses se encontravam tanto no item Computador como em Ciência da Computação. Conforme essas respostas, classificou-se e explicou-se a diferença entre esses conceitos, ficando claro que havia muita confusão entre eles.

Juntamente com a explicação dos conceitos, também foi feita uma apresentação dos conteúdos (lógica, máquinas, algoritmos etc.) a serem trabalhados e uma introdução sobre a história da computação. Após essas explicações, as fichas foram novamente entregues aos meninos, para que pudessem preenchê-las novamente, de acordo com a explicação dada. Conforme a tabela 5.2, pode-se perceber que a grande maioria conseguiu distinguir os conceitos.

A seguir, foi pedido aos meninos que fizessem uma pesquisa na internet, sobre a história da computação e a evolução dos computadores, devendo entregá-la na próxima aula.

TABELA 5.2 - Respostas sobre o que gostariam de aprender do computador e de Ciência da Computação

Menino	O que você gostaria de aprender sobre o computador?	O que você gostaria de aprender sobre a Ciência da Computação?
1	Mandar e-mail	Saber fazer programas
2	Sobre internet	Como o computador surgiu
3	Mexer mais no MS-DOS e no telnet	Montar e desmontar CPU Quem inventou o computador
4	Fazer animações no computador	Em que o computador serve no dia a dia
5	Tentar usar todos os programas	Eu gostaria de aprender tudo Saber montar e desmontar CPU e saber como é a linguagem do computador
6	Tudo	Tudo Como fazer um programa
7	Telnet Flash Player 5 Office 2000 Criar programa	Saber a origem da CPU
8	Montagens Configurações Utilizações de software Programações de sistemas	Montagem (placas, cabos) A evolução da tecnologia eletrônica
9	Programar clipper e html Criar endereço de telnet	O que seria placa mãe on-board Qual a diferença entre processador e memória Se tiver um modem ruim e internet a cabo, a conexão será boa?
10	Montar, desmontar Programação de sistemas	A evolução dos computadores
11	Tudo o necessário para o dia a dia	Tudo Como usar, de forma segura, um computador

5.2.2.1 Segunda e terceira aulas

Nessas aulas, utilizou-se a sala em que não havia computadores e deu-se início aos exercícios de lógica e aritmética, anexo 4. Os exercícios 1 e 2 foram solucionados, rapidamente, pelos meninos. Já o terceiro foi considerado difícil, porém devido à falta de compreensão e interpretação do enunciado.

A seguir, foi feita a correção individual dos 3 exercícios e alguns dos alunos se dirigiram à lousa, para explicar seus raciocínios em suas resoluções, sendo que as explicações foram complementadas pelos coordenadores das aulas.

Foi observado que os meninos estavam chateados e desmotivados, por não estarem trabalhando no computador e muitos deles não gostavam de matemática ou nela tinham dificuldade. E o mais importante era que não conseguiam relacionar a matemática com a computação.

Mesmo assim, prosseguiu-se com os exercícios de lógica e aritmética e, para os próximos encontros, planejou-se utilizar alguns produtos de software disponíveis na página www.plastelina.net ilustrados nas fig. 5.15 e 5.16. Todo o conjunto se encontra no anexo 24.

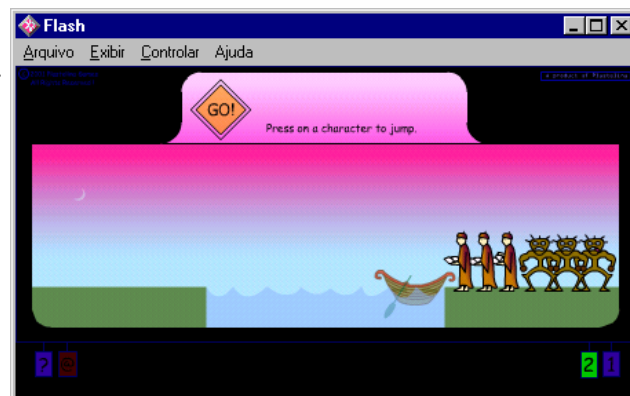


FIGURA 5.15 – Software de lógica para o exercício 2 do anexo 7



FIGURA 5.16 - Software de lógica para o exercício 2 do anexo 8

Outro software construído especificamente para motivar os alunos foi feito para o exercício 1 do anexo 7 e está ilustrado na fig. 5.17 e especificado no anexo 25.



FIGURA 5.17 - Software para construção de algoritmo - exercício 1 do anexo 7

5.3 Considerações

Tomando-se como base a primeira pesquisa realizada com crianças, procedeu-se a um diálogo com perguntas sobre a vida de cada criança e suas preferências a respeito das disciplinas e o que gostariam de fazer, quando não estivessem na escola. Em seguida, foi pedido que respondessem o que gostariam de aprender do Computador e de Ciência da Computação.

Antes de começar a ensinar saiba se seu aluno quer aprender. Se não quiser, convença-o. Aí está o segredo do sucesso no ensino. [ROD 2000]

Durante a conversa, pôde ser percebido que as crianças ficaram impressionadas em saber que poderiam fazer seu próprio jogo e construir um computador. Após, seguiu-se apresentando os conceitos fundamentais, na mesma ordem apresentada nesse trabalho.

Quando estiver escrito meninos, estarão sendo referenciadas as crianças da segunda etapa da aplicação da metodologia, pois era composta somente de crianças do sexo masculino. Quando estiver sendo tratado de crianças, estarão sendo referenciados os participantes da primeira aplicação.

Na primeira aplicação, utilizou-se, além do computador, de materiais simples, como caixas de papelão; garrafas de água etc. Procurou-se, em todos os exercícios, trabalhar de forma lúdica, quando foram estimuladas a cooperação, a autonomia e o trabalho prático, vivenciando situações comuns ao cotidiano dessas crianças, como forma de motivar a criança a aprender.

A primeira turma trabalhou os itens exatamente como na seqüência dessa pesquisa. Mas alguns dos conceitos não precisam ser trabalhados nessa ordem. Por exemplo, os conceitos máquinas, computabilidade e autômatos, podem ser abordados em qualquer ordem, sem comprometimento de seus entendimentos.

Também, houve ênfases diferentes em relação aos conceitos, enquanto que na primeira turma os exercícios de lógica foram todos resolvidos e revistos, na segunda turma os exercícios foram conduzidos com maior facilidade e rapidez.

Apesar de nenhuma das aplicações terem sido conduzidas até o final, os resultados parciais demonstram que as crianças conseguem acompanhar os conceitos vistos e os aplicam no encontro seguinte. Como conseqüência, pode-se afirmar que a faixa de idade de 9 a 12 anos é ideal para esse tipo de trabalho.

Um dos fatores fundamentais para a motivação das turmas é que as atividades sejam desenvolvidas no computador. Percebe-se isso tanto para as crianças quanto para os meninos; porém muito mais evidente para estes últimos. Isso pode ter ocorrido, devido ao ambiente em que foram desenvolvidas as aplicações. Já para a segunda turma, as atividades se desenvolveram na escola limitada a uma sala. Para a primeira turma, foi desenvolvida na residência da mestranda, onde as crianças possuíam total liberdade para fazer um intervalo, fazer um lanche, jogar, conversar etc.

Houve diferenças entre as turmas, enquanto as atividades foram resolvidas, com maior facilidade, pela segunda turma, a primeira não apresentou quase nenhum traço de cansaço durante as atividades. Acredita-se que todas essas crianças gostam de desafios.

6 Conclusão e Trabalhos Futuros

Este capítulo final da dissertação está dividido em 3 seções, que apresentam as conclusões do trabalho realizado, suas principais contribuições com apresentação dos artigos produzidos e publicados no decorrer da realização da dissertação e, finalmente, a definição de trabalhos futuros, que darão continuidade aos resultados obtidos.

6.1 Conclusões

Os dados coletados com esses professores de diversas instituições sinalizaram o conteúdo fundamental para a Ciência da Computação, o qual prestigia conceitos relativos à noção de algoritmo.

Além das pesquisas com as escolas, foram feitas diversas pesquisas na Internet, com o intuito de se descobrirem trabalhos correlatos. Todavia o mais próximo a que se chegou foi a descoberta de trabalhos envolvendo o ensino de linguagens de programação para crianças, [VIC 88].

As pesquisas com as escolas refletiram o ambiente que envolve o ensino de Informática, na região de Presidente Prudente – SP e Londrina – PR, onde, em grande parte delas, está limitado ao ensino de tecnologias correntes.

A pesquisa realizada pelo Jornal “*Folha de São Paulo*” [INF 2000] identifica como estão as estruturas das escolas do Brasil, onde, apesar de o número de escolas que possuem laboratórios de informática estar aumentando, ainda hoje, boa parte dos alunos não têm acesso ao computador.

A última pesquisa realizada visou a esclarecer e a verificar se as crianças da faixa de idade entre 10 e 12 anos gostariam de aprender Ciência da Computação. Essa pesquisa demonstrou expectativas favoráveis ao aprendizado da Ciência da Computação e a percepção positiva da influência do computador em seu cotidiano.

Na elaboração dos exercícios para demonstração e ensino dos conceitos fundamentais da Ciência da Computação e em sua aplicação, há diversos pontos que foram considerados:

- houve preocupação com trabalhar de forma interdisciplinar. Utilizou-se de mapas geográficos para explicação de proporções, de conjugação de verbos para construção de algoritmo, e, principalmente com a matemática, sendo algoritmos utilizados para desenvolvimento de cálculos, de noções de lógica, de teoria dos conjuntos etc.;
- outra preocupação foi transformar esses conceitos em aplicações que utilizassem elementos mais simples possíveis e presentes no cotidiano, sendo a transformação desses conceitos em aplicações úteis de fundamental importância, por levar motivação para o estudo de Ciência da Computação;
- houve preocupação com interagir com as crianças ouvindo-as sempre, sendo que alguns exercícios, na medida do possível, foram acrescentados ou retirados, de acordo com suas sugestões;
- procurou-se, sempre, argumentar com as crianças a utilização de cada conceito no ensino de Ciência da Computação;
- para a primeira pesquisa, as crianças possuíam liberdade, para parar ou prosseguir com os exercícios, quando achavam necessário, respeitando, assim, o tempo de aprendizado de cada uma;
- também foi dada ênfase ao trabalho em grupo, como forma de cooperação e união entre os participantes.

A primeira aplicação ocorreu simultaneamente à elaboração da metodologia, visando à adequação da forma de apresentação dos conceitos e exercícios às crianças participantes. A segunda aplicação objetivou a avaliação da metodologia em ambientes reais de aprendizagem, utilizando-se da metodologia já alterada em consequência da primeira aplicação. Infelizmente, devido a diversos fatores intransponíveis, nenhuma das aplicações foi completada. Entretanto, até esse momento os resultados obtidos revelaram que o ensino de Ciência da Computação para Crianças é viável e muito gratificante.

6.2 Principais contribuições e produção científica

São contribuições dessa pesquisa:

- desenvolvimento de uma proposta de currículo que aborde alguns dos conceitos fundamentais da Ciência da Computação. Essa é uma proposta livre, ou seja, a inversão de alguns conteúdos, ou mesmo, ênfases diferentes nos conceitos podem ser importantes, devido a grande diversidade de escolas e turmas de alunos. De fato, alguma flexibilização neste sentido foi desenvolvida com sucesso na segunda aplicação;
- identificação de uma das faixas etárias adequadas para o trabalho com Ciência da Computação;
- produção da teoria e os exercícios específicos para cada um dos conceitos fundamentais selecionados;
- produção da teoria e os exercícios de conformidade com alguns conceitos de algumas das atuais disciplinas;
- constatação de que existem escolas que desempenham um papel inovador no ensino e que estão dispostas a aceitar novos desafios como o ensino de Ciência da Computação para Crianças;
- aplicação da teoria e os exercícios, a fim de propiciar um trabalho interdisciplinar em um ambiente cooperativo e reflexivo;
- e, finalmente, o mais importante, resultados satisfatórios em que as crianças da faixa etária dessa pesquisa conseguem expressar-se formalmente, compreender o comportamento de problemas modelando, corretamente, as soluções para esses problemas.

Esta dissertação de mestrado resultou na produção e publicação de 2 artigos em congressos internacionais [FER 2000] e [FER 2000a] e 1 em congresso nacional [FER 2001]. Esses artigos estão em formato completo nos anexos 26, 27 e 28:

FERNANDES, C. S.; MENEZES, P. B.; ACCORSI, F. A Propose of Teaching Computer Science for Children. In: INTERNATIONAL CONFERENCE ON ENGINEERING AND COMPUTER EDUCATION, 2000. **Proceedings...** São Paulo: Senac, 2000.

FERNANDES, C. S.; MENEZES, P. B. Computer Science for Children. In: INTERNATIONAL CONFERENCE SOCIETY FOR INFORMATION TECHNOLOGY & TEACHER EDUCATION AND ASSOCIATION FOR THE ADVANCEMENT OF COMPUTING IN EDUCATION, 2001. **Proceedings...** Flórida: SITE, 2000.

FERNANDES, C. S.; MENEZES, P. B.; ACCORSI, F. Metodologia do Ensino de Ciência da Computação: uma proposta para crianças. In: WORKSHOP DE

INFORMÁTICA NA ESCOLA - CONGRESSO DA SOCIEDADE BRASILEIRA DE COMPUTAÇÃO, 2001. **Proceedings...** Fortaleza: SBC, 2001.

6.3 Trabalhos futuros

Esse estudo apresentou algumas limitações normais, considerando o tempo hábil para o mestrado e, outras, de força maior, como, greve dos professores e horários disponíveis pelos participantes, que poderão ser contornadas em futuras aplicações:

- a aplicação de toda a metodologia em uma turma de crianças da faixa etária identificada nessa pesquisa;
- a necessidade de criação de vários produtos de software, como os identificados, no anexo 24, para os exercícios de lógica, pois o uso do computador nas atividades é de fundamental importância para crianças, havendo desmotivação entre elas, caso isto não ocorra;
- a necessidade de criação de um software como o do anexo 25, generalizando a construção e execução de algoritmos, em cujo novo software deve haver a possibilidade de inserção de novas primitivas e figuras correspondentes a essas primitivas, possibilitando, assim, que seu uso seja para qualquer um dos algoritmos identificados nos anexos 7 e 8;
- um estudo das teorias de aprendizagem e da melhor forma de se proceder ao ensino de Ciência da Computação para Crianças. Também, um estudo da avaliação da aprendizagem para análise do desempenho das crianças, pois a avaliação de conhecimentos é parte integrante do processo ensino-aprendizagem;
- a necessidade de um estudo, nos moldes da primeira aplicação, das aulas, resultados e produções da segunda, visto que continua após a greve que a cessou;
- um maior detalhamento na comparação entre as duas aplicações, confrontando os resultados obtidos na segunda com a primeira;
- um estudo de quais dos conceitos identificados podem ser ensinados às crianças. Quais estruturas de dados se poderia incluir: fila, pilha? Pode-se, também, pensar em trabalhar outros tipos de lógica, como a lógica da incerteza (fuzzy) ou, ainda, outras arquiteturas, como redes neurais. Vale ressaltar, que estruturas de dados foi um dos itens principais, selecionados na pesquisa com os professores. Acredita-se que o momento ideal seria após serem abordados todos os conceitos detalhados nesta pesquisa, pois seria fácil introduzir as estruturas utilizando-se do conceito de máquinas, como, máquina de Post, em que é trabalhada a estrutura do tipo fila e máquina com pilhas.
- outra sugestão para trabalho futuro, porém mais distante, seria que a Ciência da Computação fosse uma disciplina do currículo do ensino médio, devendo essa disciplina, sempre, interagir com as demais. Não é tarefa simples a concretização de uma proposta com essa dimensão. Exigiria toda uma reestruturação do ensino e vontade de todos os envolvidos: vontade dos políticos que atuam no ensino, interação de docentes das disciplinas tradicionais e de Ciência da Computação, assim como apoio dos estudantes, para melhoria de sua formação educacional.

Anexo 1 Ficha de pesquisa com escolas

O que é feito?

Ficha de Pesquisa com Escolas

Sou aluna do Mestrado em Ciência da Computação da UFRGS e para fundamentação em minha pesquisa solicito que me responda o seguinte questionário.

Cláudia Santos Fernandes,

Aluna do Curso de Mestrado em Ciência da Computação da UFRGS; Professora da Faculdade de Informática de Presidente Prudente da UNOESTE

Paulo Blauth Menezes,

Doutor em Matemática pelo IST/Universidade Técnica de Lisboa, Portugal; Professor Adjunto da UFRGS; Chefe do Departamento de Informática Teórica do Instituto de Informática da UFRGS

Data: __ / __ / ____

Pesquisador: _____

Escola: _____

Endereço: _____

Quantidade de Alunos: _____

Séries abrangidas: _____

Pessoa que respondeu: _____

Função: _____

1) Há informática Educacional na escola?

 Sim

Não

2) Quais equipamentos a escola possui?

Equipamentos	Configuração	Quantidade
Computadores		
Impressora		
Scanner		
Recursos para robótica		
Lousa eletrônica		
Outros		

3) Como é o local onde estão os recursos de informática, e formas de uso?

Localização da sala em relação às outras aulas	Integradas (como auxílio durante as aulas): _____ _____
	Separadas (como apoio às disciplinas): _____ _____
	Outra forma: _____ _____ _____
Disposição dos Equipamentos	Em linha: _____ Em "U": _____ Outra forma: _____ _____
Horários das aulas:	
Observações:	

4) **Como as aulas com recursos de informática são ministradas?**

- **Professores de informática:**

Quantos: _____

Formação: _____

- **Professores das disciplinas básicas:**

- **Forma de integração com profissionais de informática:**

- **Treinamento oferecido:**

- **equipamentos:**

Quantidade por salas: _____

Detalhamento dos tipos por salas:

- **alunos**

- **Quantidade por aulas:** _____

- **Quantidade por equipamentos:** _____

- **Séries:** _____

Observações: _____

5) Quais produtos de software são utilizados para fins educacionais?

Produto	Quantidade de Cópias	Há quanto tempo possui	Que frequência é utilizado

Observações adicionais:

Anexo 2 Ficha de pesquisa com os professores

FICHA DE PESQUISA COM OS PROFESSORES

Data: ___ / ___ / ___

Professor: _____

Instituição: _____

Formação: _____

Disciplinas em que atua: _____

Prezado professor,

Sou aluna do Mestrado em Ciência da Computação – UFRGS.

Solicito, independentemente da aplicação, que me responda de forma intuitiva e objetiva:

PARA VOCÊ QUAIS SÃO OS CONCEITOS FUNDAMENTAIS DA CIÊNCIA DA COMPUTAÇÃO?

- _____
- _____
- _____
- _____
- _____
- _____

Muito obrigado!

Cláudia Santos Fernandes

Aluna do Curso de Mestrado Ciência da Computação da UFRGS; Professora da Faculdade de Informática de Presidente Prudente – UNOESTE.

Paulo Blauth Menezes

Doutor em Matemática pelo IST/Universidade Técnica de Lisboa, Portugal; Professor Adjunto da UFRGS; Chefe do Departamento de Informática Teórica do Instituto de Informática da UFRGS.

Anexo 3 Ficha de pesquisa com as crianças

FICHA DE PESQUISA COM AS CRIANÇAS

Data: ___/___/___

Aluno: _____

Idade: _____ Série: _____

Escola: _____

Disciplinas que mais gosta	Disciplinas que menos gosta

O que você gostaria de aprender sobre o computador?

O que você gostaria de aprender sobre Ciência da Computação?

Exemplos: Uma calculadora em que você pode aprender a manusear ou verificar seu funcionamento, ou construí-la.

Um carro em que se pode

Você se consegue lembrar de mais alguns exemplos?






Anexo 4 Exercícios de lógica e aritmética (Autores desconhecidos)

1. Se o salário de um homem é de 2.000,00 por semana, se ele gasta 1.400,00 por semana, quanto demorará, para juntar 3.000 em economia?
2. Se 2 lápis custam 5,00, quantos lápis poderão ser comprados com 60,00?
3. Tenho 1 caixa: dentro dela 2 outras caixas; dentro de cada uma delas, outras 2. Quantas caixas tenho?
4. Tendo-se 2 vasilhas de 5 e 3 litros, como se conseguem exatamente 7 litros de água?
5. Tendo-se 2 vasilhas de 7 e 5 litros, como se conseguem exatamente 8 litros de água?
6. Tendo-se 2 vasilhas de 4 e 9 litros, como se conseguem exatamente 7 litros de água?
Obs. Pode-se encher e esvaziar as vasilhas várias vezes
7. O relógio A marca mais 12 minutos que o B, que marca menos 7 minutos que o de C, que tem atraso de 4 minutos sobre o D, que está atrasado 9 minutos. Um relógio certo marca 14:57. Quantos marcam os de A, B, C e D?
8. A soma de 2 números pares consecutivos é 22. Eles são?
9. João caminhou 150 metros para a frente. Dobrou para a direita e caminhou 50 metros. Dobrou para a direita e caminhou 100 metros. Dobrou para a direita e caminhou 50 metros. A que distância está do ponto de partida?
10. A está à direita de B e à esquerda de C, que está à esquerda de D. Se E ficar entre A e B, e F entre C e D, qual a posição de C e D, qual a posição de B e D, considerando os que estão de um lado e outro de cada uma?
11. A é mais alto 5 centímetros do que B, que é mais baixo 12 centímetros do que C e este mais alto 3 centímetros do que D. Qual a diferença de altura entre A e D?
12. A teria mais 50,00 do que B, se obtivesse mais 30,00, porém, tem menos 40,00 do que C, que tem mais 80,00 do que D, que com mais 20,00, ficará com 150,00. Quanto tem cada um?
13. 4 galinhas põem 8 ovos em 4 dias, 12 galinhas quantos ovos porão em 12 dias?
14. De 15 em 15 minutos, um trem pára numa estação intermediária. Viajou 1 hora. Em quantas estações intermediárias parou?
15. Qual o total de bisavós de seu avô paterno?
16. Quem é o filho de sua mãe que não é seu irmão nem sua irmã?




Anexo 5 Exercícios de lógica símbolo-imagem (Piaget em [FUR72])







O exercício inicia com a visualização de um exemplo e depois prossegue de acordo com a demonstração.

1ª Parte: Elementos Básicos














Exemplo: C → 	
? → 	S
? → 	A
? → 	M
? → 	C

As respostas seriam S, A, M, C, respectivamente, mas é ressaltado que são válidas outras resposta alternativas.









Exemplo: S → 	
C →	
A →	

Exemplo: C →	 	
		 
		 




Substitua ? por alguma coisa


C →			?
? →	}		
? →			
? →	}		
? →			
? →			
? →			
? →			



Exemplo: A \nrightarrow  (\nrightarrow novo símbolo)

A	{	?		
		?	 	
		?	 	
		?	  	
M	\rightarrow	?		
M	\nrightarrow	{	?	
			?	
			?	

Lembrar que as respostas exibidas ao lado são apenas possíveis respostas. Nunca há somente uma resposta correta, exceto no caso das setas, (\rightarrow , \nrightarrow) quando uma das alternativas é correta e a outra é falsa.

?	\rightarrow	{		
?	\nrightarrow			
?	\nrightarrow		 	

Corrigir: H \rightarrow 


?	\rightarrow	}		
?	\rightarrow			
?	\nrightarrow			
?	\nrightarrow			

2ª Parte: Negação

$M \rightarrow ?$	
-------------------	--

Exemplo: (\neg novo símbolo)

$\neg M \rightarrow$   	
--	--

$? \rightarrow$ $? \rightarrow$ $? \rightarrow$ $? \rightarrow$		
--	--	--

$C \rightarrow ?$	
-------------------	--

$C \left\{ \begin{array}{l} \nrightarrow ? \\ \nrightarrow ? \\ \nrightarrow ? \end{array} \right.$	
---	--

$\neg C \rightarrow ? ?$	
--------------------------	--





$\neg C ?$   	
--	--

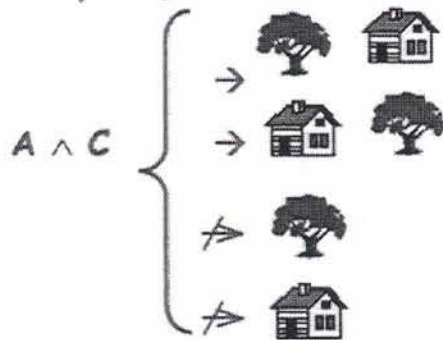
$\left. \begin{array}{l} ? \rightarrow \\ ? \nrightarrow \\ ? \nrightarrow \\ ? \nrightarrow \end{array} \right\} \text{☀}$	
---	--

$-M \left\{ \begin{array}{l} \rightarrow ? \\ \nrightarrow ? \end{array} \right.$	
---	--



$\begin{array}{l} -S \nrightarrow ? \quad ? \\ -C \nrightarrow \\ ? \nrightarrow \\ ? \nrightarrow \end{array}$	
---	--



3ª Parte: Conjunção

$C \rightarrow$?	
$A \rightarrow$?	
$C \ ?$	 	
$A \ ?$	 	

Exemplo: (\wedge novo símbolo)

$S \wedge M$	\rightarrow	
$M \wedge S$	\rightarrow	
$M \wedge S$	\rightarrow	


$? \wedge ?$	\rightarrow	}  
?	\rightarrow	
?	\rightarrow	
?	\rightarrow	
?	\rightarrow	
?	\nrightarrow	
?	\nrightarrow	



$A \wedge C$? $\neg S$? $\neg M$?	}  	
--	---	--

$C \wedge M$	\rightarrow ? \nrightarrow ? ? \nrightarrow ? ? \nrightarrow ? ?	
--------------	---	--

$A \wedge C$	$\left\{ \begin{array}{l} ? \text{ \\ ? \text{ \\ ? \text{ \\ ? \text{ \end{array} \right.$	
--------------	--	--

Corrigir esta expressão sem rasurar nenhuma parte:

$A \wedge C \rightarrow$ 	
--	--

$M \wedge C$? M ? C ?	}  	
----------------------------------	---	--

$\left. \begin{array}{l} -M \wedge S \\ \\ \\ ? \quad \emptyset \\ ? \quad \text{apple} \\ ? \quad \text{apple} \quad \text{house} \end{array} \right\} \begin{array}{l} \rightarrow ? \\ \rightarrow ? \\ \rightarrow ? \quad ? \\ \\ \end{array}$	
---	--

$\left. \begin{array}{l} -C \quad ? \\ C \quad ? \\ -C \wedge A \quad ? \\ -C \wedge \neg A \quad ? \end{array} \right\} \emptyset$	
---	--

$\left. \begin{array}{l} M \wedge C \quad ? \\ ? \rightarrow ? \\ ? \rightarrow ? \end{array} \right\} \text{apple} \quad \text{house}$	
---	--

$\left. \begin{array}{l} ? \quad ? \quad \rightarrow \\ ? \quad ? \quad \rightarrow \\ ? \quad ? \quad \rightarrow \\ \quad ? \quad \rightarrow \\ \quad ? \quad \rightarrow \\ \quad ? \quad \rightarrow \\ A \wedge C \quad ? \\ A \wedge M \quad ? \\ A \wedge \neg M \quad ? \\ \neg A \wedge M \quad ? \\ \neg A \wedge \neg M \quad ? \end{array} \right\} \text{tree} \quad \text{sun}$	
--	--








$\neg A \wedge M$	$\left\{ \begin{array}{l} ? \quad \emptyset \\ \rightarrow ? \\ \nrightarrow ? \end{array} \right.$	
-------------------	---	--

$\neg A \wedge M$	$\left\{ \begin{array}{l} ? \quad \emptyset \\ \rightarrow ? \\ ? \quad \text{tree} \\ ? \quad \text{house} \\ ? \quad \text{apple} \end{array} \right.$	
-------------------	--	--

5ª Parte: Disjunção








Exemplo:



$A \wedge C$


$\left. \begin{array}{l} \rightarrow \\ \nrightarrow \\ \nrightarrow \\ \nrightarrow \end{array} \right\}$		
		
		
		


Exemplo: (\vee símbolo novo)



$A \vee C$

$\left. \begin{array}{l} \rightarrow \\ \rightarrow \\ \rightarrow \\ \nrightarrow \end{array} \right\}$		
		
		
		

$S \vee C?$ $S \wedge C?$	$\left. \right\}$  	
------------------------------	---	--

$S \wedge M?$ $S \vee M?$	$\left. \right\}$ 	
------------------------------	---	--

$\neg C \wedge S?$ $\neg C \vee S?$	$\left. \right\}$ 	
--	---	--

$\neg C \wedge S?$ $\neg C \vee S?$	$\left. \right\}$  	
--	---	--

6ª Parte: Conjunção e Disjunção com negação

$\left. \begin{array}{l} A \wedge \neg M ? \\ A \vee \neg M ? \end{array} \right\} \emptyset$	
---	--

$\left. \begin{array}{l} \neg M \vee \neg C ? \\ \neg M \wedge \neg C ? \end{array} \right\} \text{árvore} \quad \text{sol}$	
--	--

$\left. \begin{array}{l} \neg M \vee \neg C ? \\ \neg M \wedge \neg C ? \end{array} \right\} \text{maçã} \quad \text{sol}$	
--	--

$\neg M \vee \neg C \left\{ \begin{array}{l} ? \text{ árvore} \\ ? \text{ maçã} \\ ? \text{ casa} \end{array} \right.$	
--	--

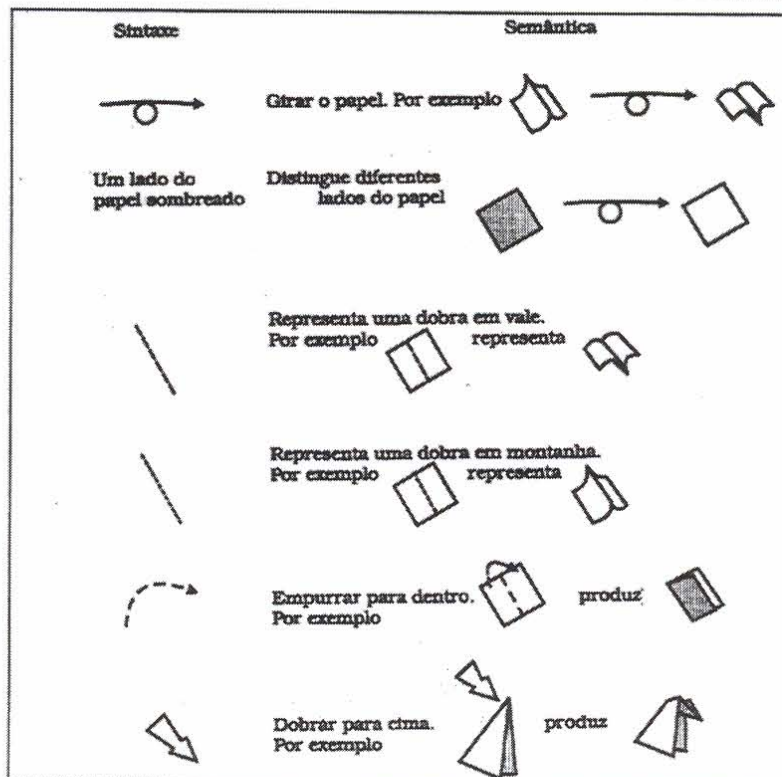
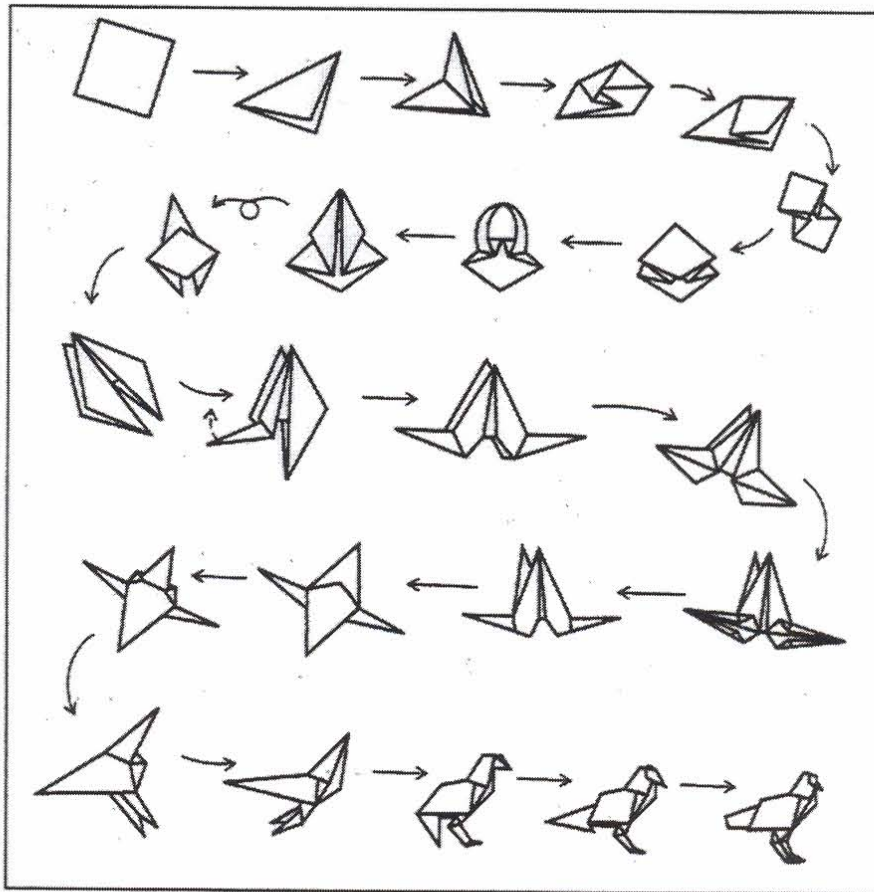
Exemplo: $\neg M \wedge \neg C \not\Rightarrow \text{maçã} \quad \text{casa}$

$\neg M \wedge \neg C \left\{ \begin{array}{l} ? \text{ árvore} \\ ? \text{ maçã} \\ ? \text{ casa} \end{array} \right.$	
--	--

Exemplo: $\neg M \vee \neg C \not\Rightarrow \text{maçã} \quad \text{casa}$

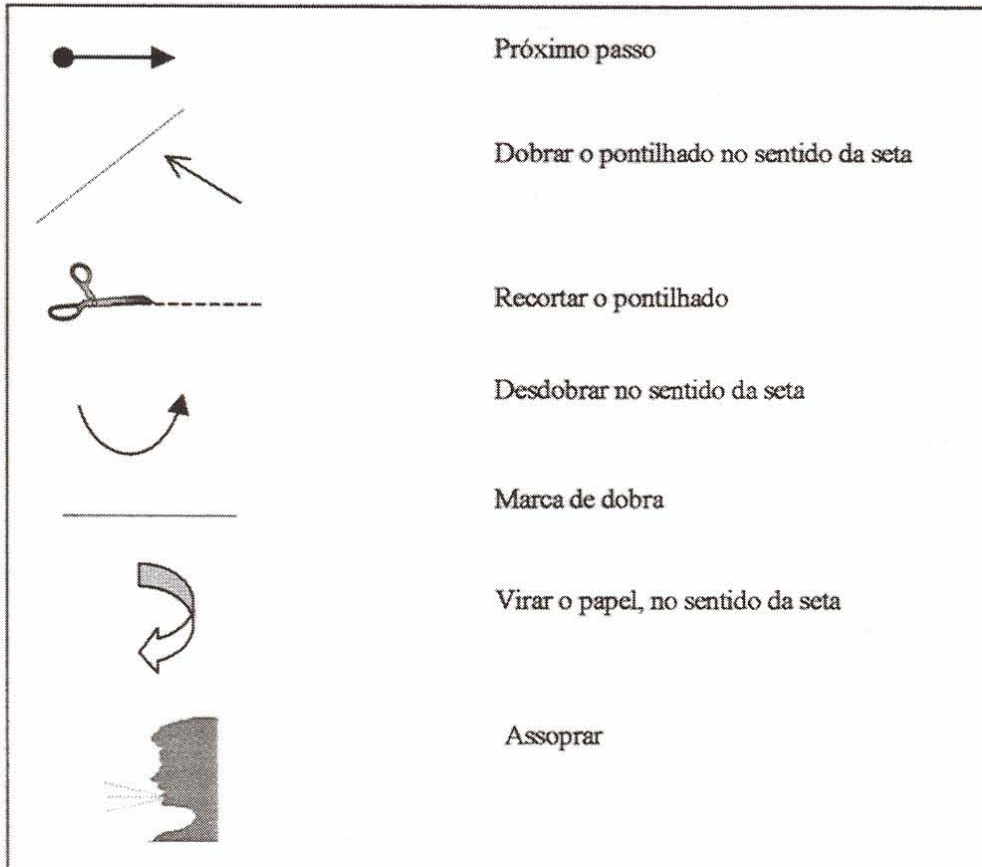
$\neg ? \vee \neg ? \rightarrow \text{casa} \quad \text{sol} \quad \text{maçã}$	
$? \vee \neg ? \not\Rightarrow \text{casa} \quad \text{sol}$	
$? \vee ? \not\Rightarrow \text{casa} \quad \text{sol}$	

Anexo 6 Exemplos de algoritmos e primitivas para construção de um pássaro [BRO00] e de um balão em dobradura

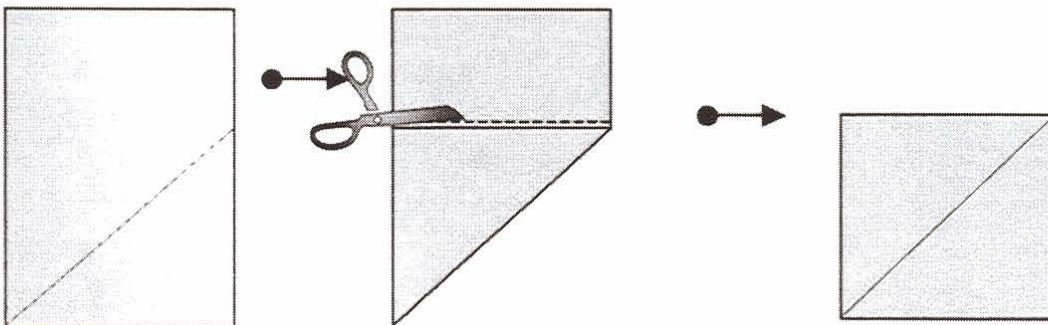


Exemplo de primitiva e algoritmo, respectivamente, para construção de um balão de dobradura

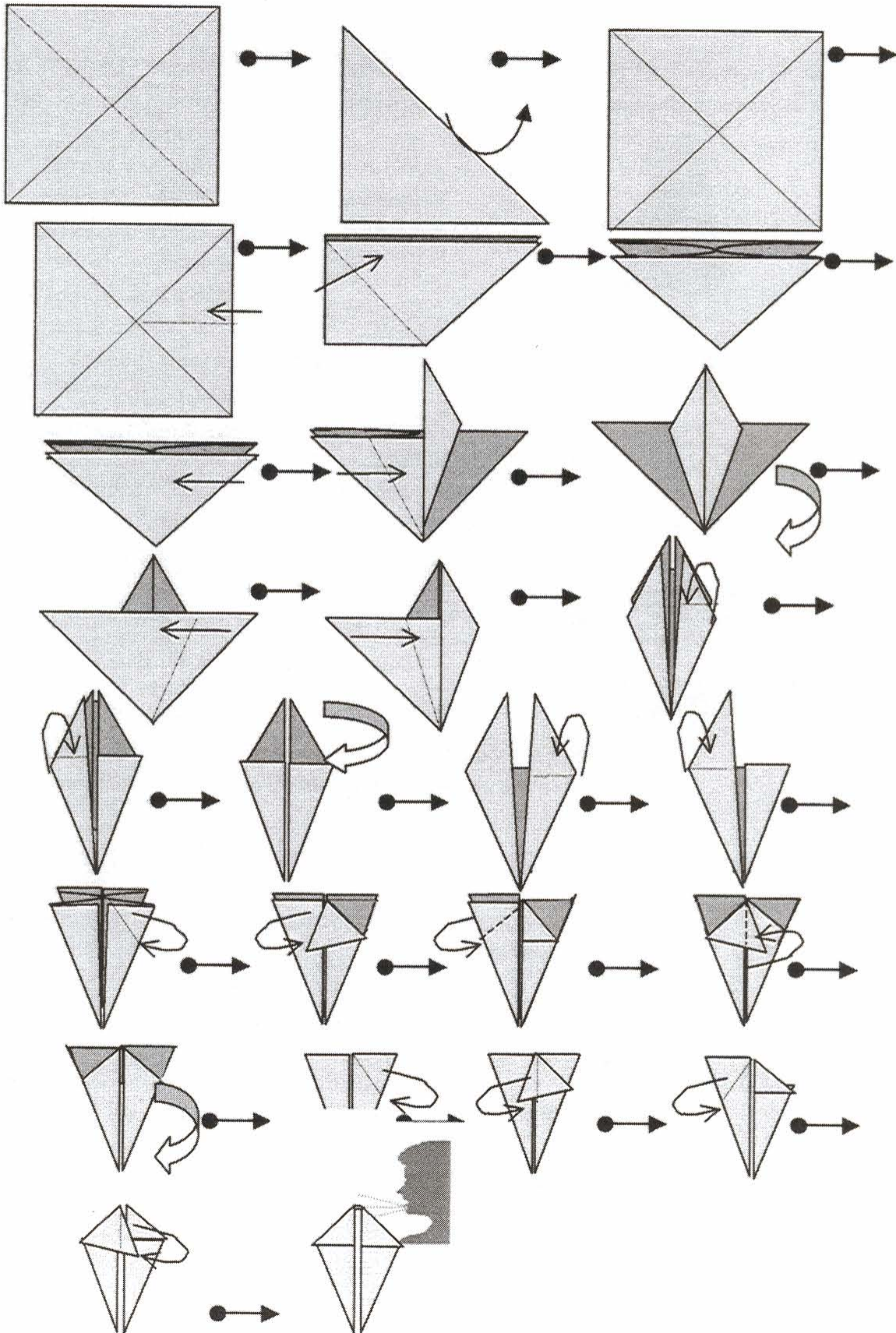
Primitivas



Algoritmo para transformar um papel no formato de um retângulo em um papel de formato quadrado



Algoritmo para construir um balão de dobradura partir de um papel no formato quadrado



Anexo 7 Exercícios para construção de algoritmos

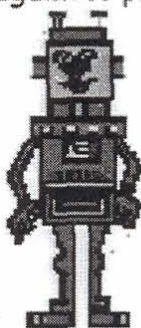
Utilizando as seguintes primitivas construa os algoritmos para os seguintes problemas:

1. [FOR93] Suponha que você possua um robô e queira fazê-lo trocar uma lâmpada, sendo que o mesmo foi programado para obedecer as seguintes primitivas:

- pegue <objeto>
- pressione <objeto>
- gire pernas 180° no sentido horário
- gire pernas 180° no sentido anti-horário
- mova <objeto> para <lugar>
- desloque-se para <lugar>

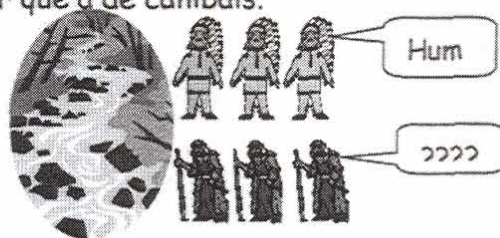
e ainda é capaz de:

- perceber quando algum comando não pode mais ser
- sentir alguma fonte de calor



2. [FOR93] Três jesuítas e três canibais precisam atravessar um rio para tal dispõem de um barco com capacidade para duas pessoas. Faça o algoritmo utilizando as seguintes primitivas para atravessar todos, mas tome cuidado para que em nenhuma das margens a quantidade de jesuítas seja menor que a de canibais.

- Jesuíta entra no barco
- Canibal entra no barco
- Jesuíta sai do barco
- Canibal sai do barco
- Barco atravessa rio da margem direita para esquerda
- Barco atravessa rio da margem esquerda para direita



3. Utilizando as primitivas abaixo, elabore um algoritmo que execute um jogo da velha.

- Enquanto todos os quadrados não estiverem ocupados e X ou O não tenham marcado três quadrados em linha
- Escolha o <número do quadrado>
- Se o quadrado não estiver ocupado, marque o quadrado <número do quadrado> com um <X ou O>
- Escolha o <número do quadrado>
- Se X ou O tem dois quadrados quaisquer em linha marque no terceiro desocupado

1	2	3
4	5	6
7	8	9

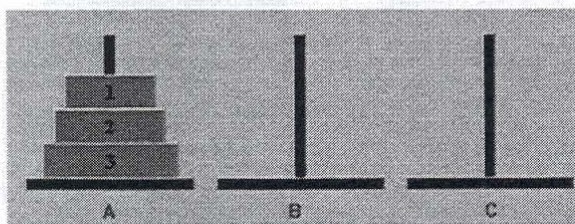
Obs.: Este exercício utiliza-se de estruturas de seleção e repetição

Anexo 8 Exercícios para construção de primitivas e de algoritmos

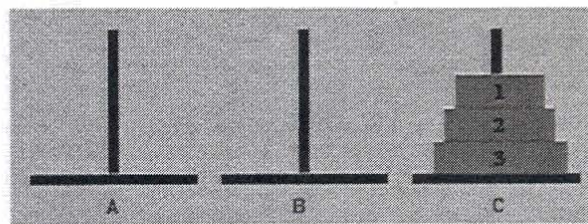
Construir as primitivas e o algoritmo para os seguintes problemas:

1. [GRA94] Problema da Torre de Hanói

São dados três pinos com 3 discos, inicialmente empilhados por tamanho decrescente em um dos três pinos. O objetivo é transferir a torre inteira de discos para um dos outros pinos, movendo apenas um disco de cada vez e nunca colocando um disco maior em cima de um menor. Apenas um disco pode ser movido por vez e cada disco deve sempre ser colocado em um dos pinos.



Configuração inicial



Configuração final

2. [FOR93] Problema para atravessar um rio.

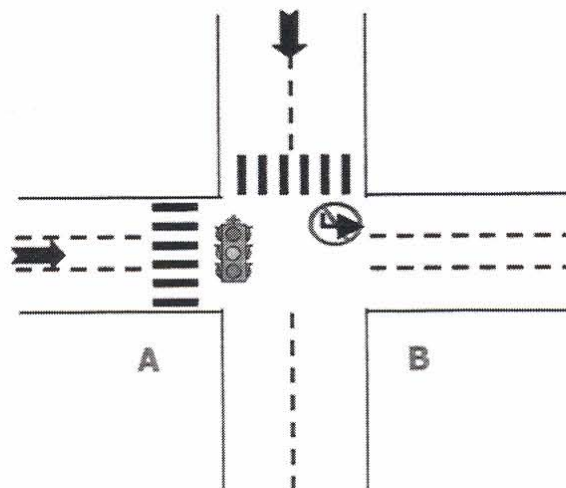
Um homem precisa atravessar um rio e suas três cargas: um lobo, uma ovelha e um maço de alfafa. Porém ele possui somente um barco em que cabem ele e somente uma de suas cargas por vez. Tomar cuidado para não perder nenhuma carga.

3. [AMS00] Problema para resolver uma equação do primeiro grau.

Chama-se equação do 1º grau, na incógnita x , toda equação que pode ser expressa na forma $ax + b = 0$, onde a e b são constantes quaisquer com $a \neq 0$.

4. [FOR93] Problema para atravessar uma rua.

Resolver o problema para uma pessoa que está na posição A chegar em segurança para o local B. Para tal, observe o sentido do tráfego na encruzilhada, a faixa, o sinal de pedestres, o semáforo e a placa de trânsito.



Anexo 9 Exemplo de algoritmo para criação de um papel de carta na ferramenta *Microsoft Word*

**Algoritmo para construir um papel de carta personalizado
(com borda e figura)**

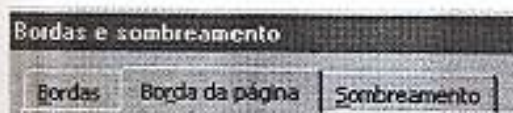
Quais seriam as primitivas?

Siga os seguintes passos:

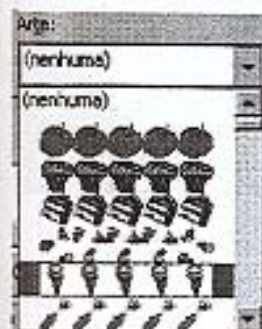
1ª Parte: Colocar uma Borda decorada:



1. Clique no Menu Formatar e escolha a opção Bordas e Sombreamento



2. Escolha a opção Borda de Página

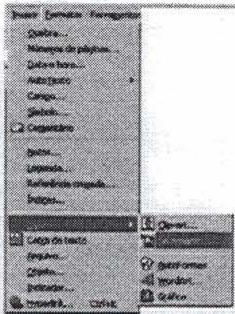


3. Na caixa Arte escolha o desenho que deseja para sua borda

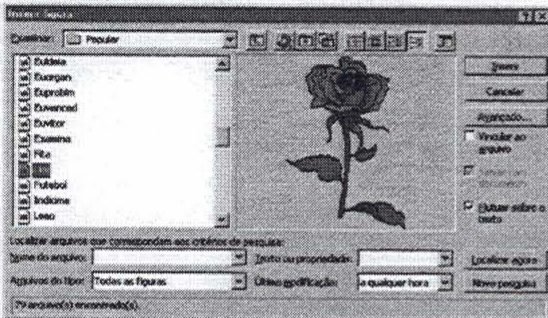


4. Clique em OK e terá uma folha com bordas decoradas

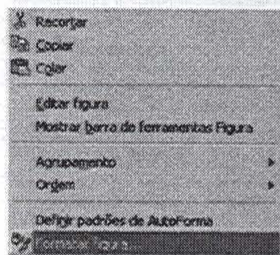
2ª Parte: Colocar uma Figura de Fundo:



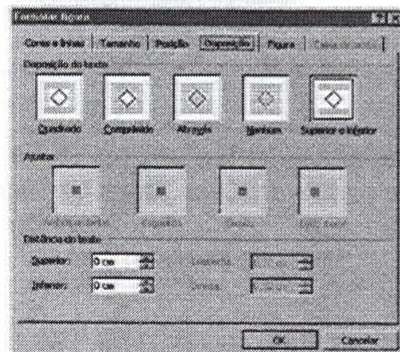
5. Clique no Menu Inserir e escolha as opções Figura e Do Arquivo



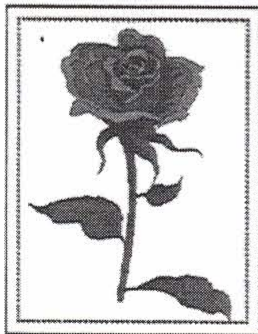
6. Selecione a figura desejada e clique em Inserir



7. Aumente a figura do tamanho desejado, e clique com o botão direito do mouse na opção Formatar Figura



8. Escolha a opção Disposição e Através

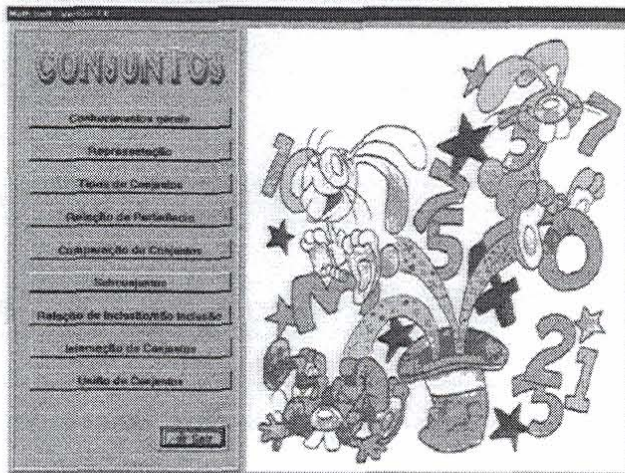


9. Clique em OK e terá uma folha com uma figura que pode ser escrita sobre ela

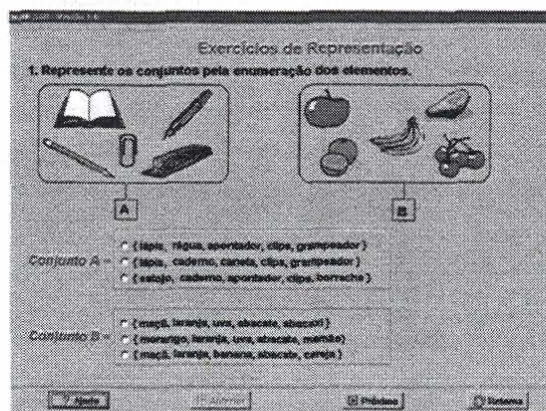
Anexo 10 Software Matemático para Ensino Fundamental: Teoria dos Conjuntos

Software Matemático para Ensino Fundamental: Teoria dos Conjuntos

Um software que pode ser utilizado como reforço para o ensino da Teoria dos Conjuntos está incluído no sistema CAI, dentro da modalidade de exercício-e-prática. É baseado numa seqüência de exercícios sobre a teoria dos conjuntos, reunindo exercícios de cada tópico do estudo: Representação, Tipos, Comparação de Conjuntos etc. Há também um tópico que contém exercícios sobre Conhecimentos Gerais, podendo ser utilizado como uma forma de avaliação geral do aprendizado. A figura abaixo é a tela inicial do “Software Para Ensino Fundamental: Teoria dos Conjuntos.”



A manipulação do software exige habilidade de uso do mouse, utilizando-se poucas vezes o teclado. Foram elaborados estilos de exercícios bem diferentes. Após a resolução de cada exercício, o aluno tem a visualização de seu desempenho. Cada tela de exercícios possui opções para avançar ou voltar de tela, com opção de ajuda que contém explicações sobre cada tópico abordado. A seguir são mostradas algumas telas que servem como exemplos do funcionamento do software, o restante das telas seguem o padrão destas.



Tela do “Software Para Ensino Fundamental: Teoria dos Conjuntos”, onde existem exercícios de escolha da resposta correta.

Anexo 11 Exercícios para criação de Identificadores

Defina o nome, o tipo e dê exemplos de conteúdos para definir os identificadores para as seguintes situações.

O exercício inicia com a visualização de um exemplo e depois prossegue de acordo com a demonstração.

Exemplo: Definir o identificador para o conjunto de nome das frutas

Nome	Tipo	Exemplos de conteúdos
Frutas	Caracter	"maça", "pêra", "goiaba"

Definir o identificador para o conjunto de valores de sua mesada

Nome	Tipo	Exemplos de conteúdos

Definir o identificador para o conjunto dos valores das idades das pessoas

Nome	Tipo	Exemplos de conteúdos

Definir o identificador que guarde a resposta para a pergunta: Seu pai é advogado?

Nome	Tipo	Exemplos de conteúdos

Definir o identificador para o conjunto dos nomes das pessoas

Nome	Tipo	Exemplos de conteúdos

Anexo 12 Exercícios utilizando o comando de atribuição

Completar o conteúdo dos identificadores de acordo com os comandos de atribuição

O exercício inicia com a visualização de um exemplo e depois prossegue de acordo com a demonstração.

Exemplo: Definir o conteúdo do identificador *Número* do tipo *Inteiro* após o comando de atribuição: *Número* \leftarrow 2

Número
2

Definir os conteúdos do identificador *Número* e *X* que são do tipo *Inteiro* após os comandos de atribuição.

Comandos	Número	X
Número \leftarrow 10		
X \leftarrow 5		
Número \leftarrow 2		
X \leftarrow 4		
Número \leftarrow 10 + 2		
Número \leftarrow 8 - 3		
Número \leftarrow Número + 1		
Número \leftarrow Número + 3 - 1		
Número \leftarrow X - 2		
Número \leftarrow Número - X		
Número \leftarrow 14		
X \leftarrow 7		
Número \leftarrow Número \div X		

Definir os conteúdos do identificador *Nome1* e *Nome2* que são do tipo *Caracter* após os comandos de atribuição.

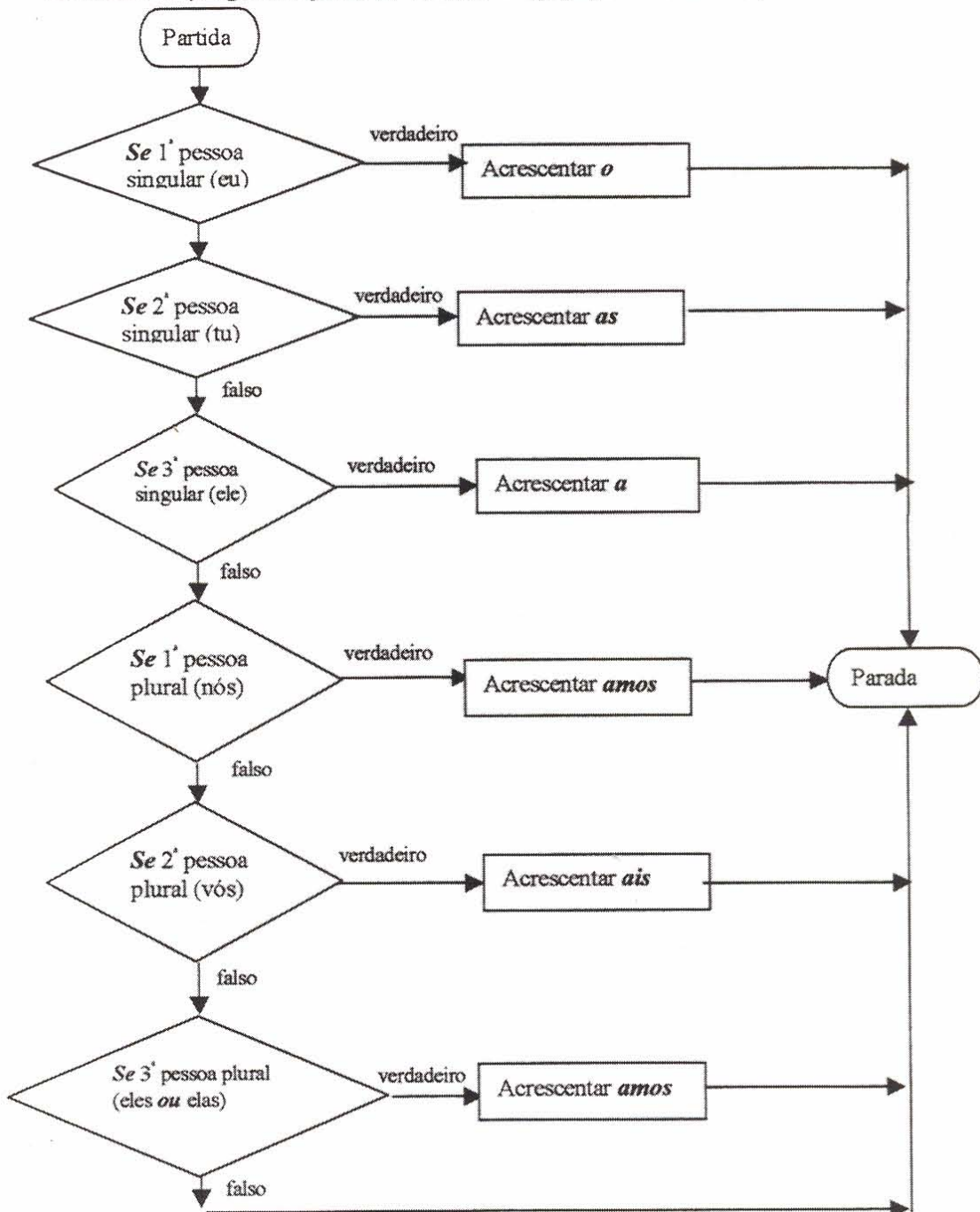
Comandos	Nome1	Nome2
Nome1 \leftarrow 'Maria'		
Nome2 \leftarrow 'Ana'		
Nome1 \leftarrow Nome1 + Nome2		

Anexo 13 Exemplos e exercícios para desenvolvimento de programas

I. Trabalhando com Estrutura de Seleção

Construir um programa de conjugação de verbos, no presente do modo indicativo, que pertençam a 1ª conjugação (vogal temática a): am-a-r, cant-a-r.

obs.: - o algoritmo mostrará qual a terminação que deverá acrescentada ao radical
- estender o programa para as demais conjugações e/ou tempo e/ou modo



II. Trabalhando com identificadores, estrutura de Seleção, Repetição, leitura e exibição de valores

1. Construir um programa para eleger um representante da classe.

Obs. :- Irão ser fornecidos os nomes de 3 candidatos

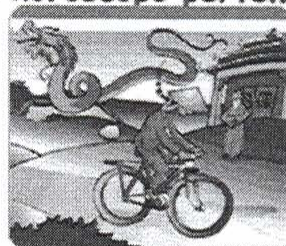
- Fazer uma repetição para ir totalizando o votos de cada candidato
- exibir qual candidato ganhou (fez o maior número de votos)



2. Construir um programa que descubra a qual signo do horóscopo pertence uma pessoa.

Obs. :- Irá ser fornecida a data de nascimento da pessoa

- Estender para:
 - descobrir o signo no horóscopo Chinês
 - exibir as principais características de cada signo



3. Construir um programa para somar a quantidade de pontos de um jogo de boliche. Para cada partida ir pedindo os pinos derrubados para cada bola jogada (que podem variar de 0 pontos (canaleta) a 10 pontos (strike))

Obs. :- uma partida é constituída de 10 jogadas

- em cada jogada pode-se arremessar uma ou duas bolas
- cada pino equivale a 1 ponto
- exibir qual jogador (1° ou 2°) ganhou (fez o maior número de pontos,
- estender o programa para as demais regras do jogo
 - Caso, em uma jogada um jogador fez um SPAR (soma das duas bolas de uma jogada sejam derrubados os 10 pinos), os pontos obtidos com a próxima bola arremessada também devem contar para esta referida jogada;
 - Caso, em uma jogada um jogador fez um STRIKE, os pontos obtidos com as duas próximas bolas arremessadas também devem contar para esta referida jogada;



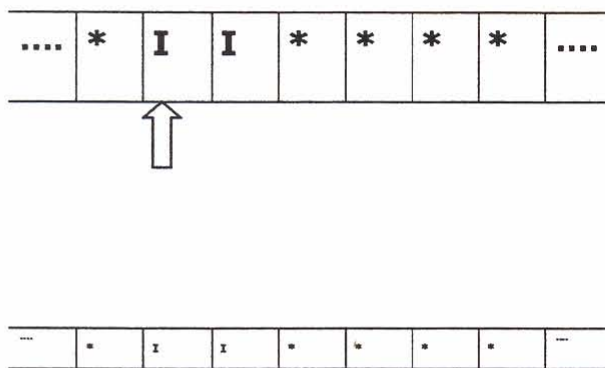
- Caso, na décima jogada o jogador fizer um STRIKE, então ele terá direito a arremessar mais duas bolas, que também contam pontos para a décima jogada. Caso, na 10ª jogada ocorra um SPAR, o jogador terá o direito a arremessar mais uma bola.

Anexo 14 Simulação da Máquina de Turing para duplicação de símbolos

Simulação da Máquina de Turing: valor inicial I e valor final II

Deve-se dar a configuração inicial para simulação

1. Configuração inicial da máquina de turing



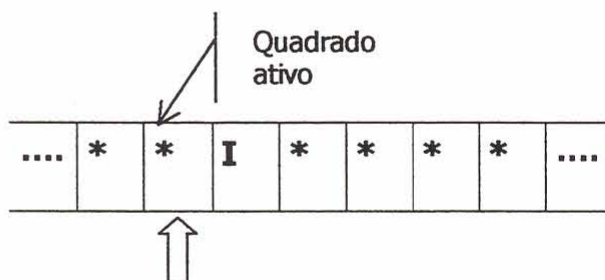
Estado da máquina

1

Lista de regras ou instruções

1	I	2	*
2	*	3	>
3	I	3	>
3	*	4	>
4	I	4	>
4	*	5	I
5	I	5	>
5	*	6	I
6	I	6	<
6	*	7	<
7	I	8	<
8	I	8	<
8	*	1	>

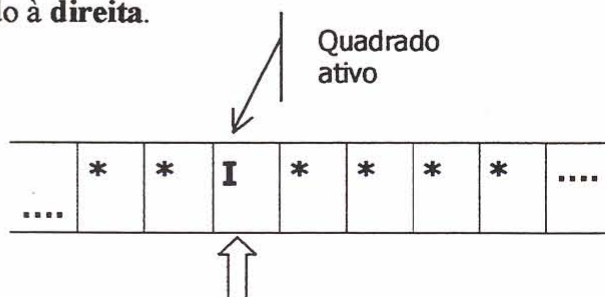
2. A máquina está no estado 1 e de acordo com a lista de regras se tiver um I no quadrado ativo deve ser substituído por * e ir ao estado 2.



Estado da máquina

2

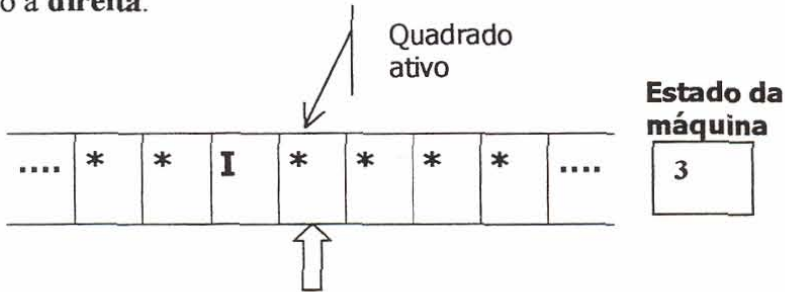
3. A máquina está no estado 2 e de acordo com a lista de regras se tiver um * no quadrado ativo deve-se ir ao estado 3 e a cabeça de leitura/gravação deve andar um quadrado à direita.



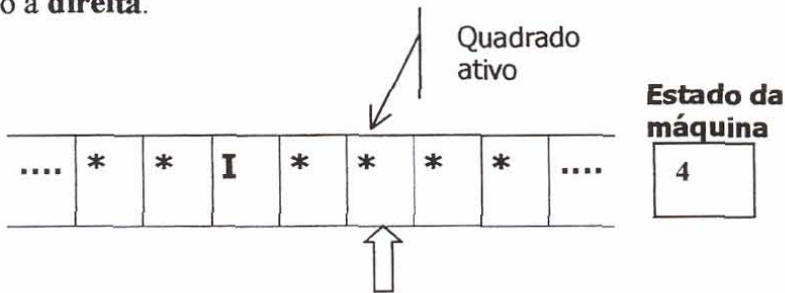
Estado da máquina

3

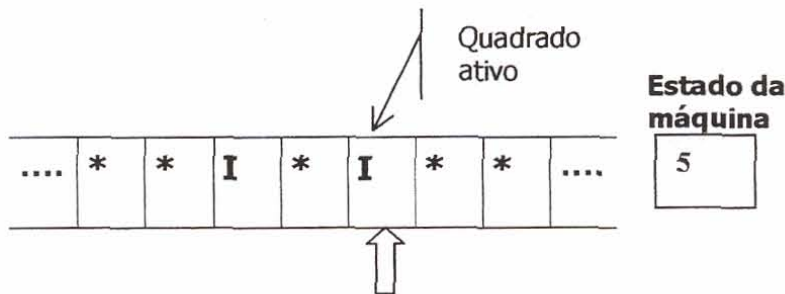
4. A máquina está no estado 3 e de acordo com a lista de regras se tiver um **I** no quadrado ativo deve-se ir ao estado 3 e a cabeça de leitura/gravação deve andar um quadrado à **direita**.



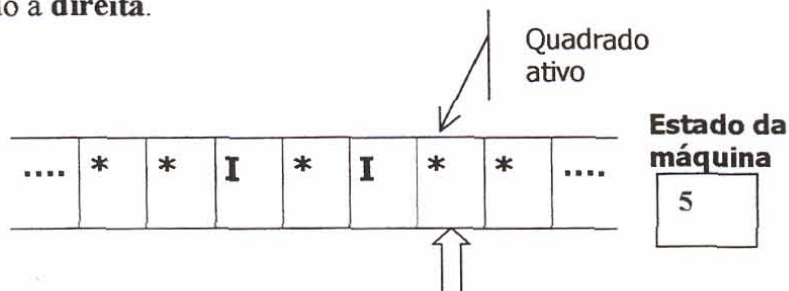
5. A máquina está no estado 3 e de acordo com a lista de regras se tiver um * no quadrado ativo deve-se ir ao estado 4 e a cabeça de leitura/gravação deve andar um quadrado à **direita**.



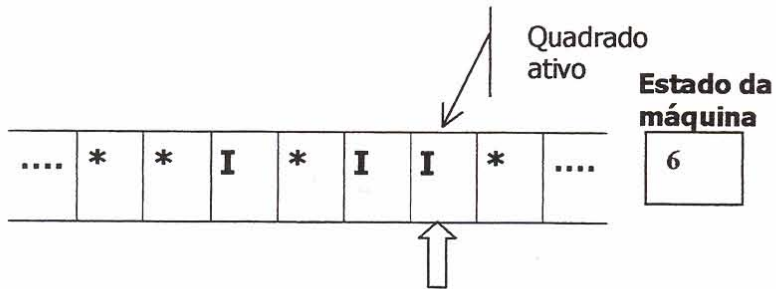
6. A máquina está no estado 4 e de acordo com a lista de regras se tiver um * no quadrado ativo deve-se substituí-lo por **I** e ir ao estado 5.



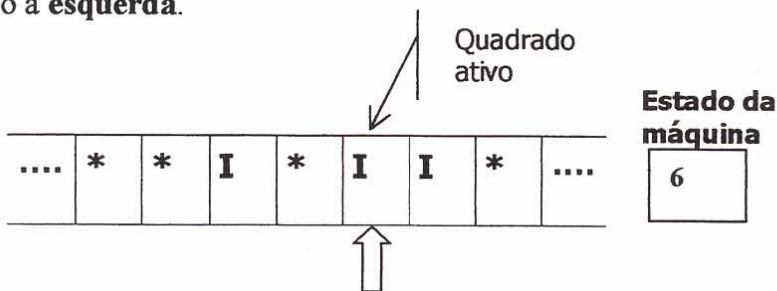
7. A máquina está no estado 5 e de acordo com a lista de regras se tiver um **I** no quadrado ativo deve-se ir ao estado 5 e a cabeça de leitura/gravação deve andar um quadrado à **direita**.



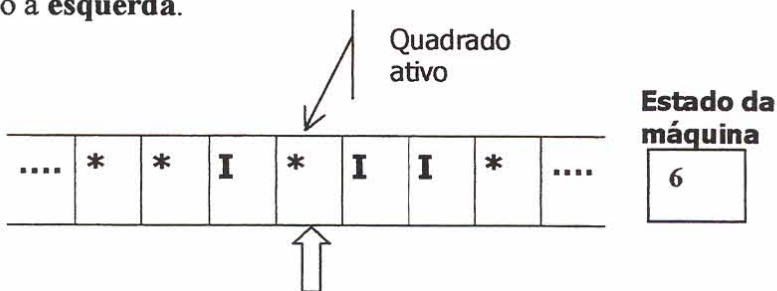
8. A máquina está no estado 5 e de acordo com a lista de regras se tiver um * no quadrado ativo deve-se substituí-lo por I e ir ao estado 6.



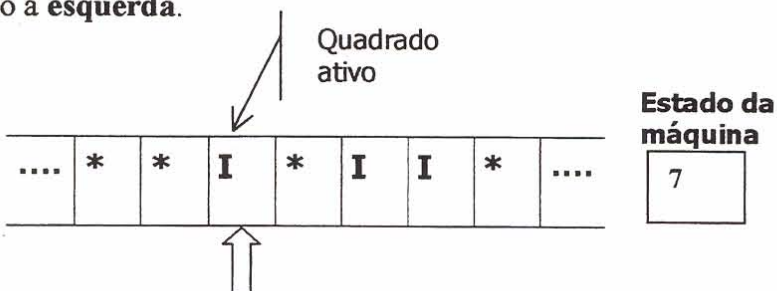
9. A máquina está no estado 6 e de acordo com a lista de regras se tiver um * no quadrado ativo deve-se ir ao estado 6 e a cabeça de leitura/gravação deve andar um quadrado à esquerda.



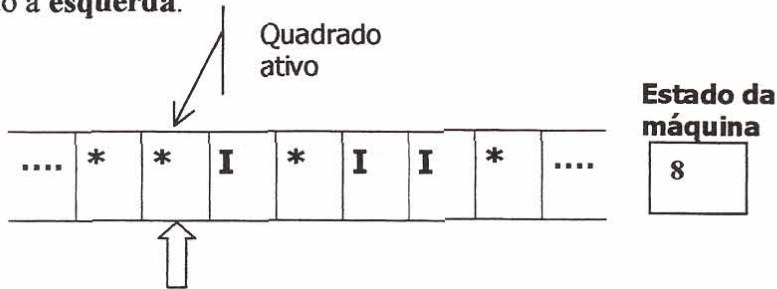
10. A máquina está no estado 6 e de acordo com a lista de regras se tiver um I no quadrado ativo deve-se ir ao estado 6 e a cabeça de leitura/gravação deve andar um quadrado à esquerda.



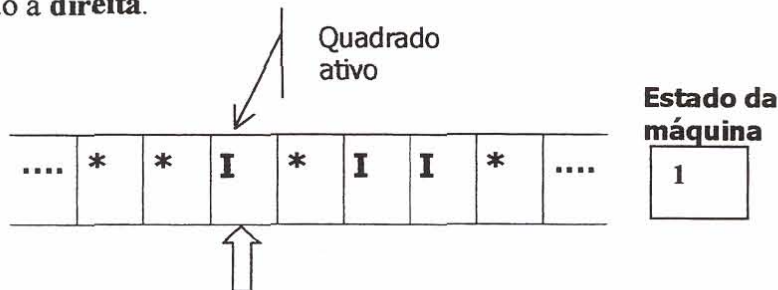
11. A máquina está no estado 6 e de acordo com a lista de regras se tiver um * no quadrado ativo deve-se ir ao estado 7 e a cabeça de leitura/gravação deve andar um quadrado à esquerda.



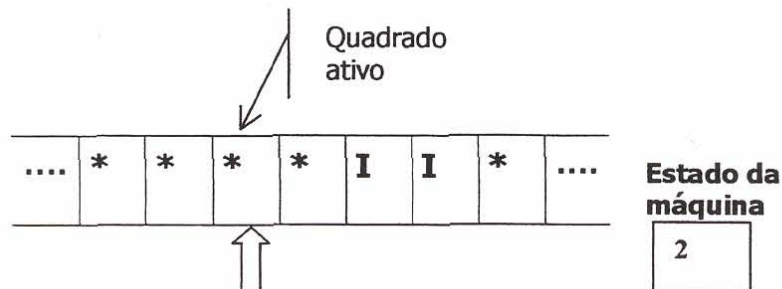
12. A máquina está no estado 7 e de acordo com a lista de regras se tiver um I no quadrado ativo deve-se ir ao estado 8 e a cabeça de leitura/gravação deve andar um quadrado à esquerda.



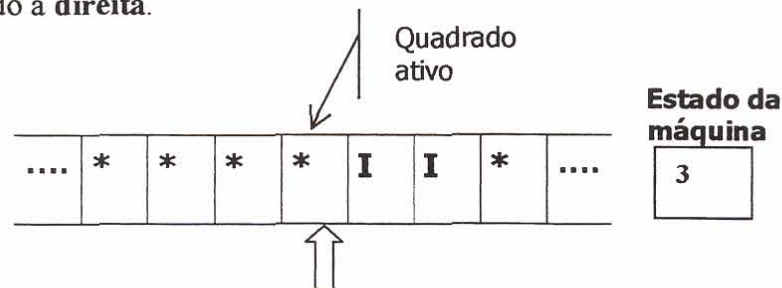
13. A máquina está no estado 8 e de acordo com a lista de regras se tiver um * no quadrado ativo deve-se ir ao estado 1 e a cabeça de leitura/gravação deve andar um quadrado à direita.



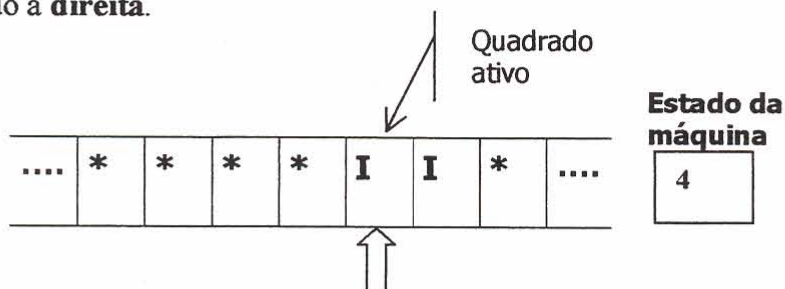
14. A máquina está no estado 1 e de acordo com a lista de regras se tiver um I no quadrado ativo deve-se substituí-lo * e ir ao estado 2.



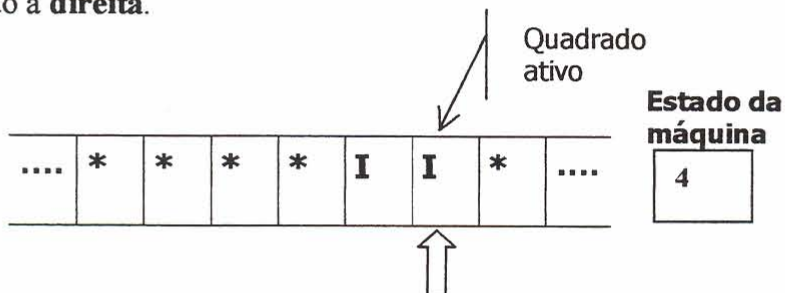
15. A máquina está no estado 2 e de acordo com a lista de regras se tiver um * no quadrado ativo deve-se ir ao estado 3 e a cabeça de leitura/gravação deve andar um quadrado à direita.



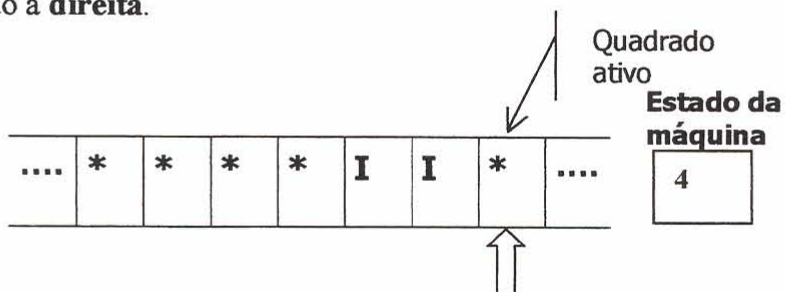
16. A máquina está no estado 3 e de acordo com a lista de regras se tiver um * no quadrado ativo deve-se ir ao estado 4 e a cabeça de leitura/gravação deve andar um quadrado à **direita**.



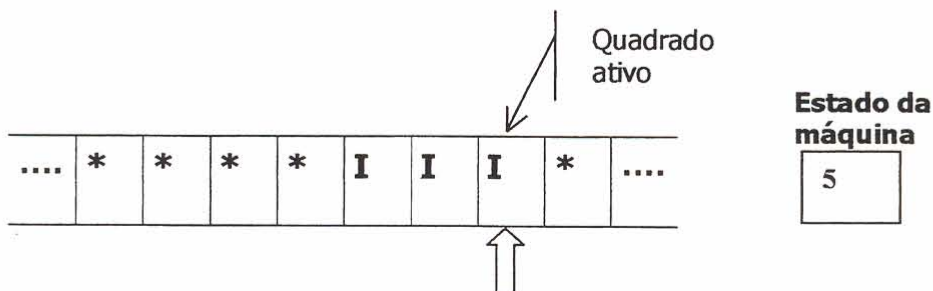
17. A máquina está no estado 4 e de acordo com a lista de regras se tiver um I no quadrado ativo deve-se ir ao estado 4 e a cabeça de leitura/gravação deve andar um quadrado à **direita**.



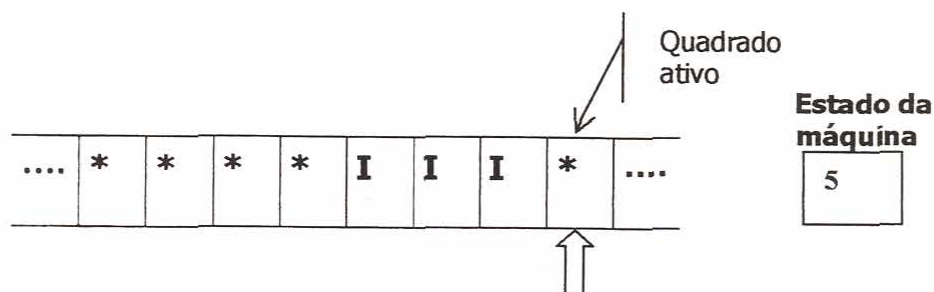
18. A máquina está no estado 4 e de acordo com a lista de regras se tiver um I no quadrado ativo deve-se ir ao estado 4 e a cabeça de leitura/gravação deve andar um quadrado à **direita**.



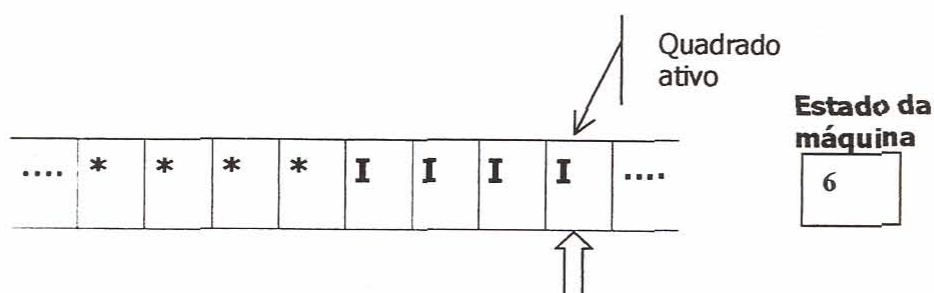
19. A máquina está no estado 4 e de acordo com a lista de regras se tiver um * no quadrado ativo deve-se substituí-lo por I e ir ao estado 5.



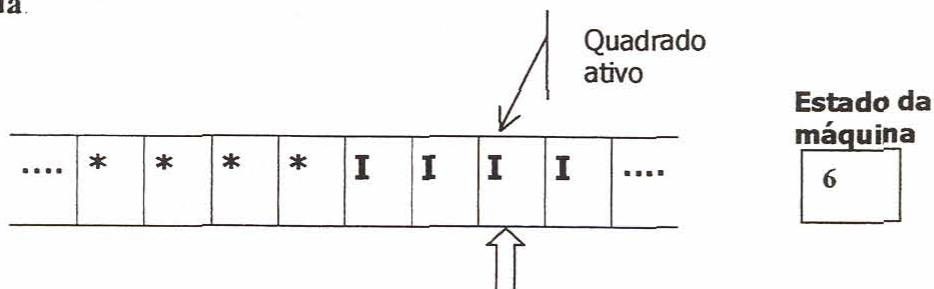
20. A máquina está no estado 5 e de acordo com a lista de regras se tiver um I no quadrado deve-se ir ao estado 5 e a cabeça de leitura/gravação deve andar um quadrado à direita.



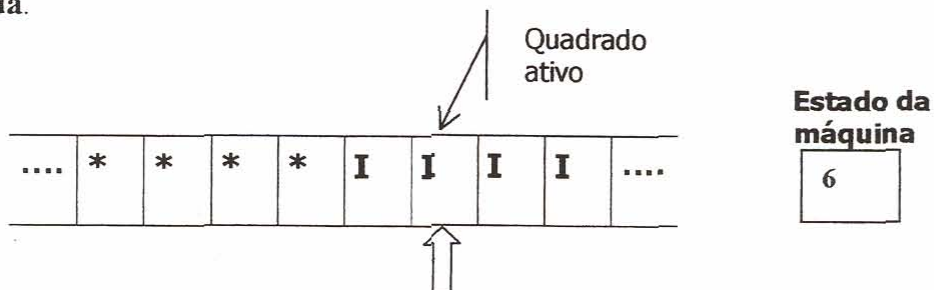
21. A máquina está no estado 5 e de acordo com a lista de regras se tiver um * no quadrado deve-se substituí-lo por I e ir ao estado 6.



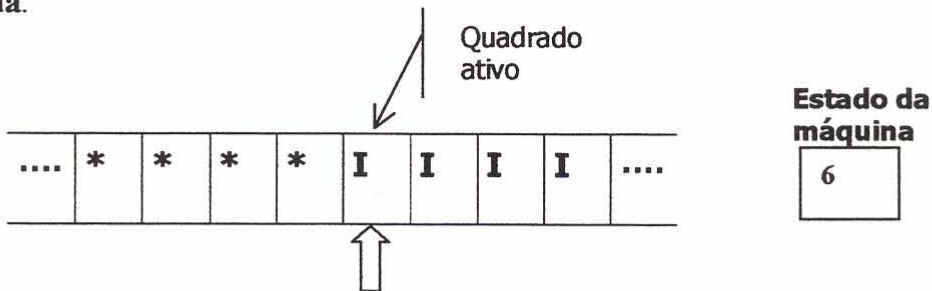
22. A máquina está no estado 6 e de acordo com a lista de regras se tiver um I no quadrado deve-se ir ao estado 6 e a cabeça de leitura/gravação deve andar um quadrado à esquerda.



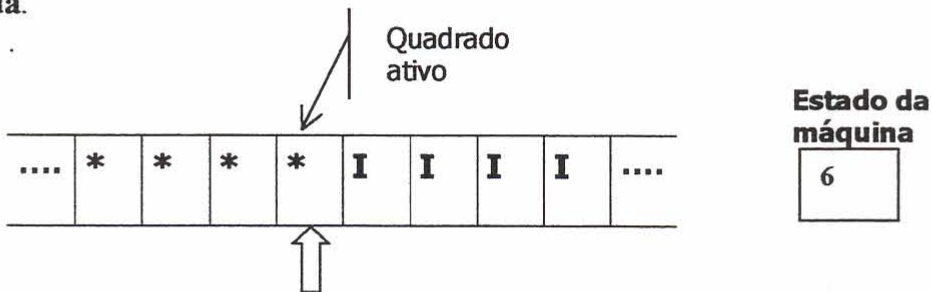
23. A máquina está no estado 6 e de acordo com a lista de regras se tiver um I no quadrado deve-se ir ao estado 6 e a cabeça de leitura/gravação deve andar um quadrado à esquerda.



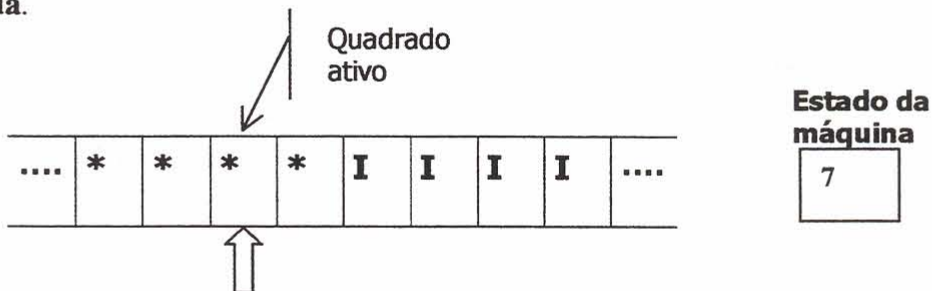
24. A máquina está no estado 6 e de acordo com a lista de regras se tiver um I no quadrado deve-se ir ao estado 6 e a cabeça de leitura/gravação deve andar um quadrado à esquerda.



25. A máquina está no estado 6 e de acordo com a lista de regras se tiver um I no quadrado deve-se ir ao estado 6 e a cabeça de leitura/gravação deve andar um quadrado à esquerda.



26. A máquina está no estado 6 e de acordo com a lista de regras se tiver um * no quadrado deve-se ir ao estado 7 e a cabeça de leitura/gravação deve andar um quadrado à esquerda.



27. A máquina está no estado 7 e de acordo com a lista de regras, não há instrução para o caso para o estado 7 se há um *, portanto a máquina pára pois não sabe o que fazer



A **configuração final** da máquina, ilustrada acima, contém o dobro de símbolos (I) iniciais.

Valor inicial: I

Valor final: II

Anexo 15 Exercícios sobre Máquina de Turing

1. Descreva uma máquina de Turing que substitua uma cadeia de III por uma de ***.

....	*	I	I	I	*	*	*
------	---	---	---	---	---	---	---	------

Estado da máquina

Lista de regras ou instruções

2. Para uma máquina em que a fita possa conter 3 tipos diferentes de símbolos: a , b e $*$.
 Descreva uma máquina de Turing que substitua uma cadeia de $a(s)$ e $b(s)$ por uma de um único a .

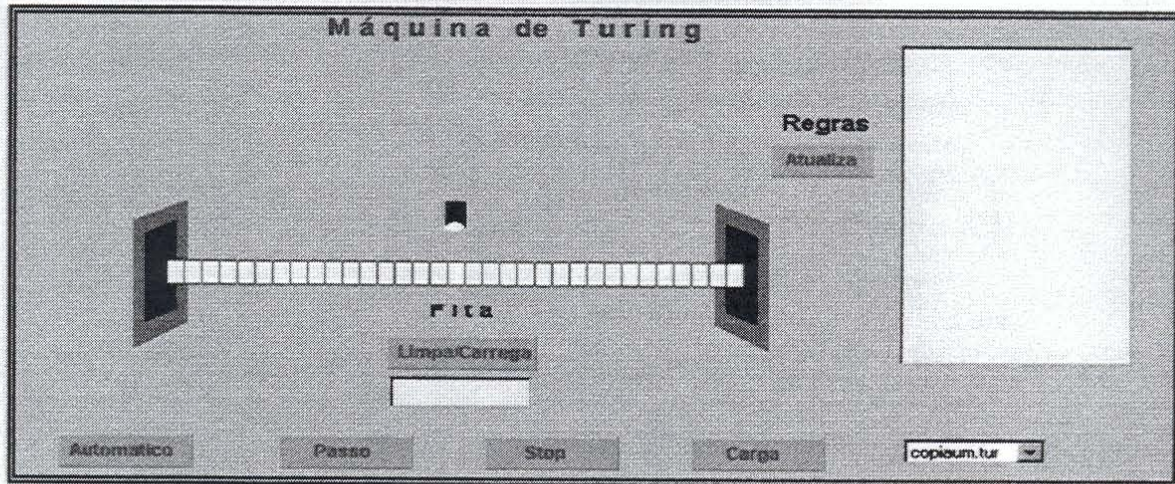
....	*	a	b	a	*	*	*
------	---	---	---	---	---	---	---	------

Estado da máquina

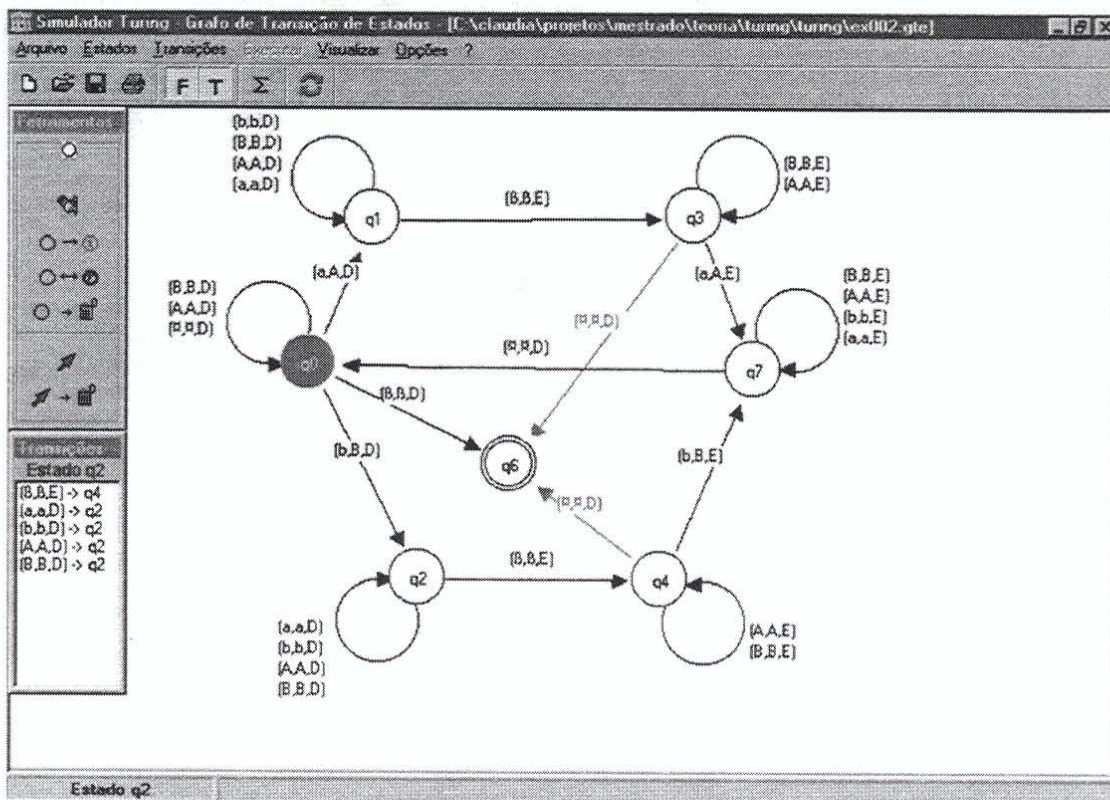
Lista de regras ou instruções

Anexo 16 Produtos de software para simulação da Máquina de Turing

A tela abaixo pertence a um software encontrado em <http://www.cic.unb.br/tutores/turing/MaqTur.html> que simula o funcionamento de uma Máquina de Turing. Podemos ver a fita, a cabeça de leitura e gravação e as regras que determina o movimento da máquina em um determinado momento.



A tela abaixo pertence a um software desenvolvido por Thiago Fernandes Moesch tendo como orientador o Prof. Dr. Tiarajú Asmuz Diverio para o Grupo de Matemática Computacional da Universidade Federal do Rio Grande do Sul. Este software edita e simula o funcionamento de uma Máquina de Turing.



Anexo 17 Exercícios para construção de autômatos

1. Exemplo de construção de autômatos para definição dos estados civis de uma pessoa

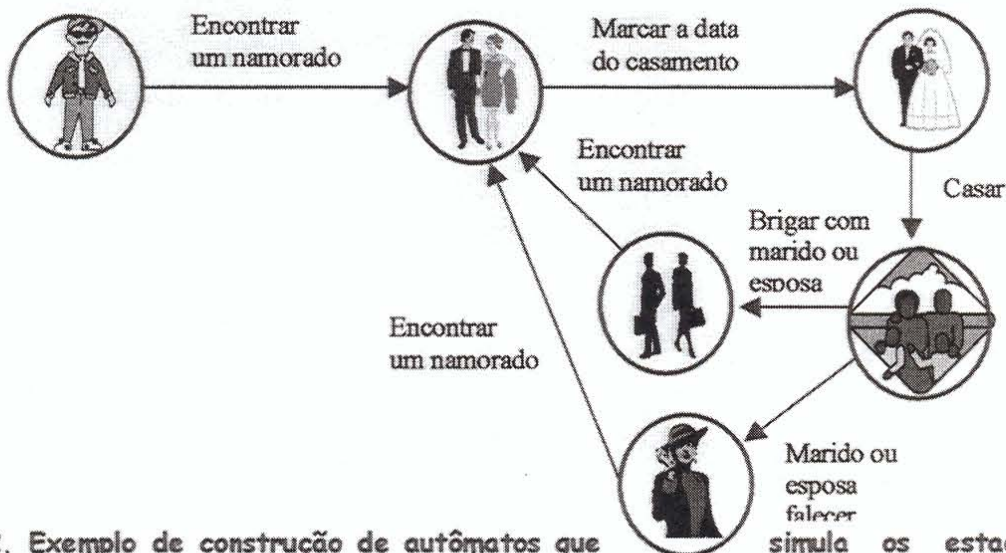
- obs.: - pode-se completar o exemplo
 - descobrir quais são estados finais e iniciais

Estados (civis)

- solteiro(a)
- estar namorando
- noivo(a)
- casado(a)
- divorciado(a)

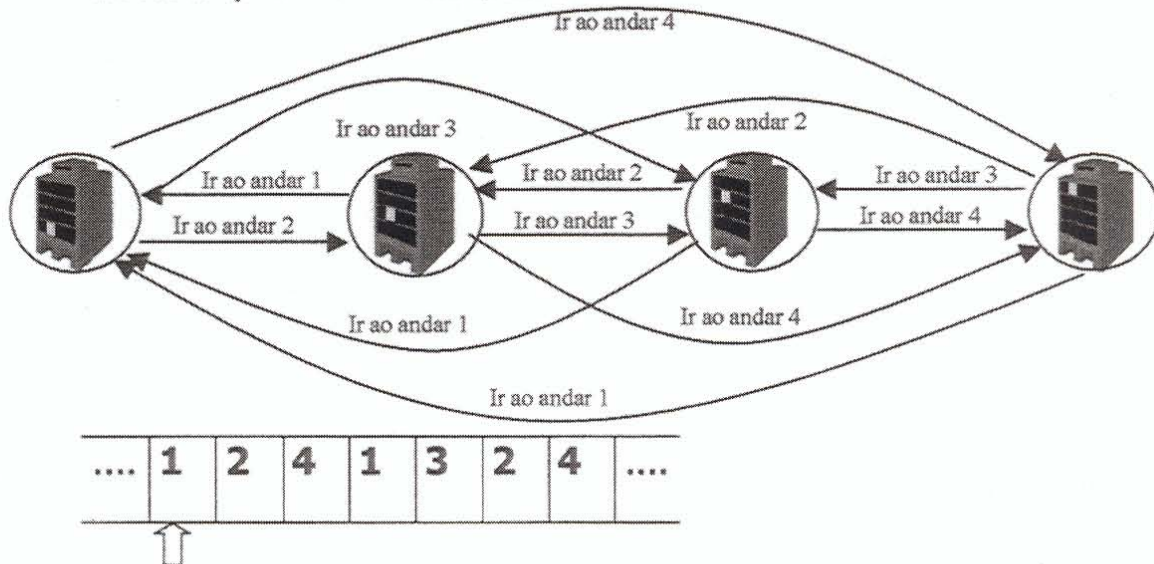
As ações

- Encontrar um namorado(a)
- Marcar a data do casamento
- Casar
- Marido/esposa falecer
- Brigar com marido/esposa



2. Exemplo de construção de autômatos que simula os estados de um elevador de 4 andares, também apresenta a fita que contém um exemplo de andares requisitados.

- obs.: - pode-se completar o exemplo
 - descobrir quais são estados finais e iniciais



3. Construir o autômato para a máquina de vendas

Uma máquina de vendas de coca-cola, descrita em [SUD98] e [DAV94], é usada para ilustrar os componentes de uma máquina de estados finitos. A máquina só aceita moedas dos seguintes valores: R\$ 1,00, R\$ 0,50, R\$ 0,25, R\$ 0,10. Supõe-se que o preço do produto à venda é de R\$ 1,00. Quando R\$ 1,00 é inserido, a tampa da máquina pode ser aberta e um produto retirado. Se o total de moedas excede R\$ 1,00, a máquina aceita o pagamento e não entrega o troco.

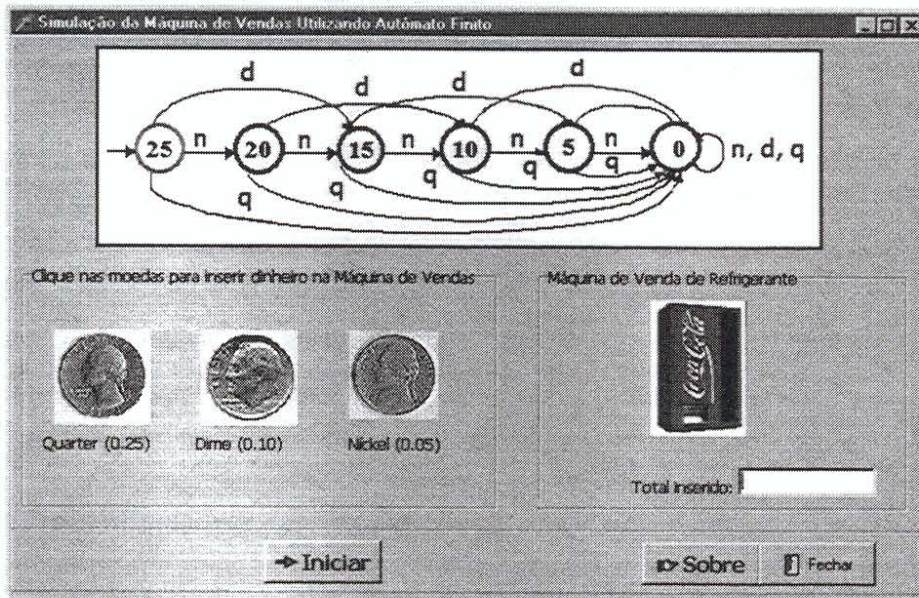
Estados

- Faltam R\$ 1,00
- Faltam R\$ 0,50
- Faltam R\$ 0,25
- Faltam R\$ 0,10
- Faltam R\$ 0,05
- Faltam R\$ 0,01

Ações

- Inserir moeda R\$ 1,00
- Inserir moeda R\$ 0,50
- Inserir moeda R\$ 0,25
- Inserir moeda R\$ 0,10
- Inserir moeda R\$ 0,05
- Inserir moeda R\$ 0,01

A tela abaixo pertence a um software que simula o funcionamento de uma máquina de vendas, porém para moedas em dólar.



4. Seguindo o padrão da máquina de vendas, fazer outros autônomos para simular uma máquina de lavar roupas, máquina de fax e conjugação de verbos.

Anexo 18 Exemplo de problemas solucionáveis/não solucionáveis

1. Problema do Assassinato do Bilionário Count Van Diamond

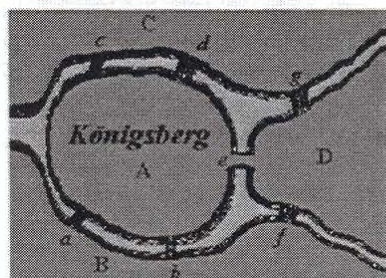
O cenário abaixo é a residência do bilionário Count Van Diamond, que acaba de ser assassinado. Sherlock Gomes (um conhecido detetive que nas horas vagas é um estudioso da teoria dos grafos) foi chamado para investigar o caso. O mordomo alega ter visto o jardineiro entrar na sala da piscina (lugar onde ocorreu o assassinato) e logo em seguida deixar aquela sala pela mesma porta que havia entrado. O jardineiro, contudo, afirma que ele não poderia ser a pessoa vista pelo mordomo, pois ele havia entrado na casa, passado por todas as portas uma única vez e, em seguida, deixado a casa. Sherlock Gomes avaliou a planta da residência e em poucos minutos declarou solucionado o caso. Quem poderia ser o suspeito indicado por Sherlock Gomes? Qual o raciocínio utilizado pelo detetive para apontar o suspeito?



2. Problema das Pontes de Königsberg

No século 18 havia na cidade de Königsberg um conjunto de sete pontes (identificadas pelas letras de a até f na figura ao lado) que cruzavam o rio Pregel. Elas conectavam duas ilhas (A e D) entre si e as ilhas com as margens (B e C).

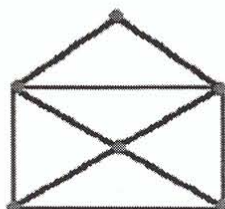
Por muito tempo os habitantes daquela cidade perguntavam-se se era possível cruzar as sete pontes numa caminhada contínua sem que se passasse duas vezes por qualquer uma delas.



3. Problema do desenho da casa

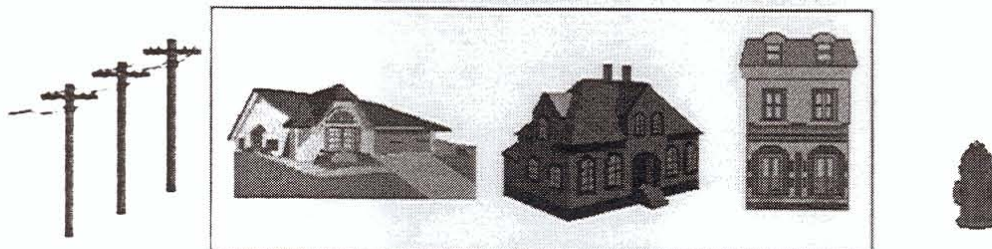
No desenho ao lado, uma criança diz ter posto a ponta do lápis numa das bolinhas e com movimentos contínuos (sem levantar e sem retroceder o lápis) traçou as linhas que formam o desenho da casa, *traçando cada linha uma única vez*.

A mãe da criança acha que ela trapaceou, pois não foi capaz de achar nenhuma sequência que pudesse produzir tal resultado. Você concorda com esta mãe?



4. Problema da ligação de serviços em uma casa

Em uma rua 3 funcionários devem fazer as ligações das redes de água, luz e gás em uma casa. Como nenhuma ligação pode cruzar com a outra, decidem primeiro fazer o desenho de como ficarão as ligações. Ajude os funcionários a fazer estas ligações.



5. Dada uma lista de números e um valor, deve-se encontrar com quais números da lista, se somados, se conseguirá o valor.

Exemplo:

Lista de números							
1029	625	890	23	694	214	1000	400

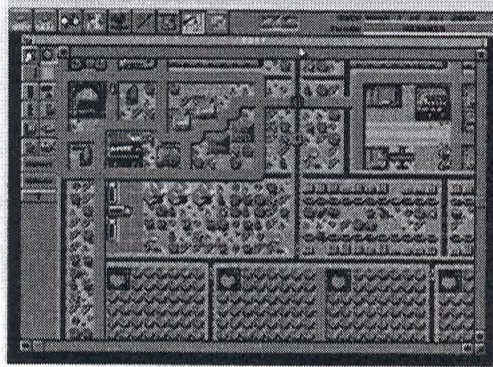
Valor
2584

Anexo 19 Telas dos jogos *Simfarm* e *Show do Milhão*

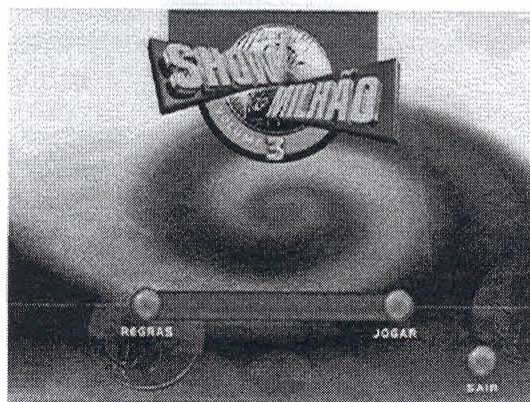
Tela inicial do software *Simfarm*



Tela do software *Simfarm* onde é construída a fazenda com suas plantações e benfeitorias



Tela do software *Show do Milhão*



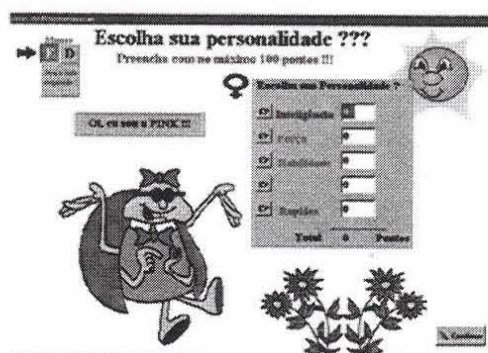
Anexo 20 – Telas do *Software de Treinamento Básico Sobre o Windows Utilizando o Jogo de Representação - RPG [FER98]*



Tela Inicial do Software de Treinamento Básico sobre o Windows utilizando o Jogo de Representações (RPG)



Tela do Software de Treinamento Básico sobre o Windows utilizando o Jogo de Representações (RPG), onde há a explicação das teclas do mouse e o aluno deve escolher se será menino ou menina.



Tela do Software de Treinamento Básico sobre o Windows utilizando o Jogo de Representações (RPG), aqui o aluno escolhe sua personalidade através de pontos quanto a sua característica (inteligência, força etc.) totalizando 100 pontos.

Anexo 21 Detalhamento dos Encontros



Ciência da Computação para Crianças

Colégio de Aplicação da UFRGS
Porto Alegre-RS - <http://www.cap.ufrgs.br>

DETALHAMENTO DOS ENCONTROS

_____ Encontro Data: ___/___/___ Horários: ___:___ a ___:___

Anexos: _____

Ferramentas:

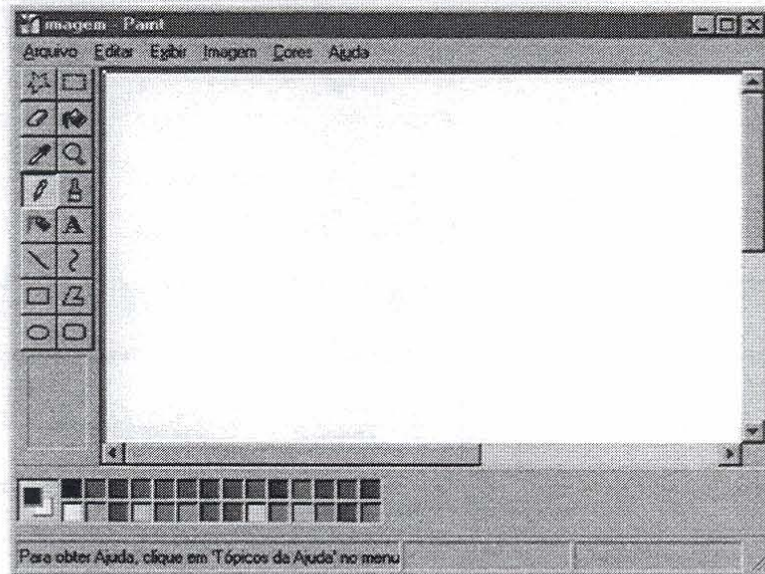
- _____
- _____
- _____

Objetivos:

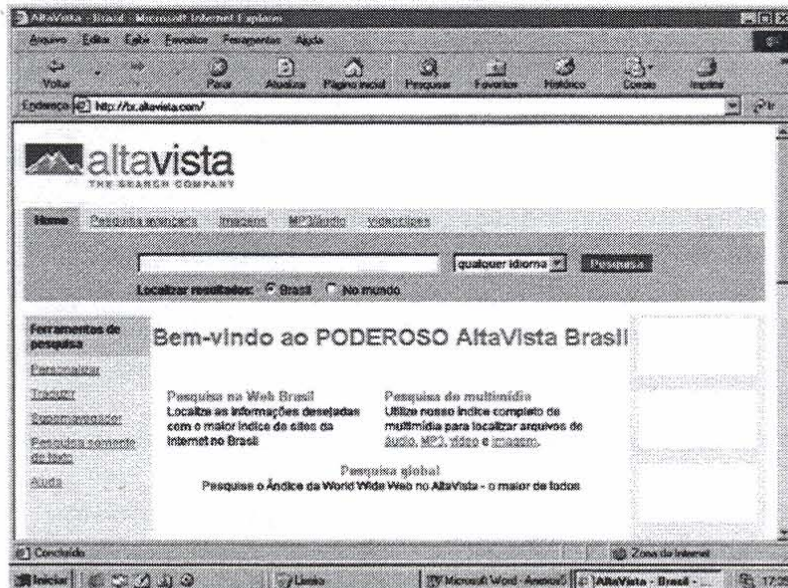
- _____
- _____
- _____
- _____
- _____

Criança	Atividades	Produção

Anexo 22 Telas do software *Microsoft Paint* e do site *www.altavista.com.br*



Tela do software *Microsoft Paint* onde são feitos desenhos



Tela do site para pesquisas na Internet *Altavista*

Anexo 23 Especificação para as aulas na Oficina do Colégio de Aplicação



Ciência da Computação para Crianças

Colégio de Aplicação da UFRGS
Porto Alegre-RS - <http://www.cap.ufrgs.br>

Atividades propostas

1o. Atividade: *Detecção de dificuldades na manipulação do computador*

Detectar se existem crianças que necessitam de aprimoramento na manipulação do computador.

Exemplos: - Manipulação do mouse, teclado, impressora, scanner;
- Manipulação de ferramentas para navegação na internet e edição de textos e figuras (podem ser *Word* e *Paint*).

Marcar uma hora de reforço com as crianças que apresentam dificuldades.

2o. Atividade: *Anexo 4 – Exercícios de Lógica e Aritmética* Exercícios 1, 2 e 12

Materiais: - computador com as ferramentas
- Impressora colorida
- Também pode ser utilizado o scanner para copiar dinheiro
- Papel e tesoura

Decidir se as crianças trabalharam individualmente ou em grupos de 2.

Sugestão:

1a. etapa: imprimir os exercícios para uma das crianças, onde ela deverá anotar seus resultados.

2a. etapa: - propor às crianças a criação dos dinheiros de várias formas:
- desenhar no *Paint*
- scanear o dinheiro
- capturar figuras na internet
- imprimir e recortar

3a. etapa: As crianças resolvem os exercícios

4a. etapa: Após as questões (decidir se a cada questão ou todas) preenchidas, deve-se reunir todo o grupo para discutir quais são os resultados e como cada criança raciocinou para obter os resultados. Procurando que elas organizem o raciocínio em uma seqüência (como se fosse um algoritmo).



Ciência da Computação para Crianças

Colégio de Aplicação da UFRGS
Porto Alegre-RS - <http://www.cap.ufrgs.br>

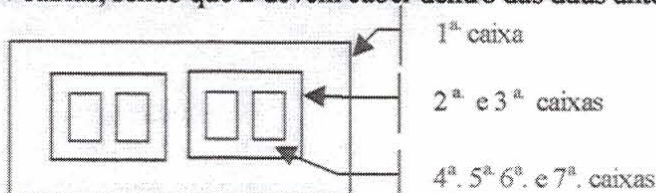
Atividades propostas

30. Atividade: Anexo 4 – Exercícios de Lógica e Aritmética

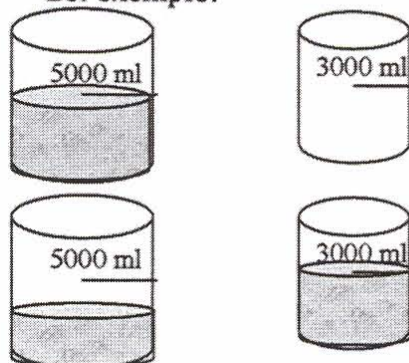
Exercícios 3 a 6 (trabalhando com medida de volume e com hora)

Materiais: para cada criança ou grupo (dependendo do trabalho individual ou em grupo): - 1 caixa grande;

- 2 caixas que devem caber dentro da anterior;
- 4 caixas, sendo que 2 devem caber dentro das duas anteriores;



- vasilhas com medidas de volume (podem ser baldes, canecas etc. Melhor se forem transparentes e não utilizar vidro);
- medidas de volumes: 3, 4, 5, 7 e 9 litros;
- água, papel, lápis e borracha;
- avisar que podem encher e esvaziar as vasilhas quantas vezes forem necessárias;
- é melhor que as crianças esquematizem primeiro no papel e depois manipulem com a água;
- Muito importante: as crianças devem anotar todas as etapas:
 - Exemplos para o exercício 3:
 - 1o. exemplo:
 - o Encher a vasilha de 5 litros com a água da torneira
 - o Encher a vasilha de 3 litros com a vasilha de 5 litros
 - o Sobraram 2 litros na vasilha de 5 litros
 - o Colocar estes 2 litros na vasilha de 3 litros
 - o Encher a vasilha de 5 litros totalizando 7 litros
 - 2o. exemplo:



- e assim em diante

- Para o exercício 7, podem ser utilizados relógios ou serem feitos vários desenhos de relógios. As crianças também devem anotar todas as etapas.

- Reunir com todo o grupo e cada um deve apresentar o raciocínio utilizado.



Ciência da Computação para Crianças

Colégio de Aplicação da UFRGS
Porto Alegre-RS - <http://www.cap.ufrgs.br>

Atividades propostas

4o. Atividade: Anexo 4 – Exercícios de Lógica e Aritmética
Exercícios 8 a 11 e 13 a 16 (trabalhando com teoria dos números;
medida de comprimento, relação)

Materiais: - Papel, lápis, borracha;
- Barbante ou corda;
- Figuras de pessoas.

Sugestão:

Continuar no mesmo esquema de trabalho anterior, com as crianças resolvendo e anotando a forma de raciocínio e depois todo o grupo discutindo os resultados.

5o. Atividade: Anexo 5 – História da Matemática ou da Computação e Exercícios de Lógica símbolo-imagem (Piaget em [FUR72])

Materiais: - Folhas com os exercícios, já impressas anteriormente
- várias letras (A, C, S, M)
- cola e vários desenhos das figuras (sol, árvore, maçã e casa), impressos e recortados. Ou lápis de cor para desenhar.
- Livro “O Mundo de Sofia”
- Internet (pesquisar a história da computação e suas máquinas, exemplos de páginas:

<http://www.cic.unb.br/turing/introduc.html>

<http://www.inf.ufrgs.br/~cabral/museu.html>

<http://www.das.ufsc.br/gia/computer/node5.html>

<http://www.bestway.com.br/~rigoletto/historia.html>

<http://www.manoeltorres.com.br>

<http://www.intel.com.br/pórtugues/home/pentium4/tech-info.htm>

[The Alan Turing Home Page.](#)

[The Virtual Museum of Computing.](#)

[Turing's World.](#)

[History of Mathematics:](#)

[David Hilbert.](#)

[Friedrich Frége.](#)

Bertrand Russel.
Alfred Whitehead.
Kurt Gödel.
Alonzo Church.
Alan Turing.
Stephen Kleene.
Emil Post.
ACM (Association for Computing Machinery):
Turing Award.
The History of Computers.

Sugestão:

Inicialmente foi feita uma pesquisa dos filósofos, especialmente Aristóteles. (livro Mundo de Sofia e [FON 98]). Podendo também ser utilizado o resumo que estou passando anexado.

Deixar o aluno pesquisar na internet para que ele anote o que achar mais interessante e passar para a sala, discutindo com todo o grupo.

Imprimir os exercícios assim os alunos irão ou colar as figuras ou desenhar.

Também discutir os resultados com todo o grupo.

6o. Atividade: Primeiras noções de algoritmos.

Arquivo: - algoritmo dobradura balão.doc
Anexo 6: - algoritmo e primitiva para construção de um pássaro de dobradura
Anexo 8: - algoritmo para construção na ferramenta *Word* de um papel de carta com borda e fundo.

Materiais: - papel, lápis, tesoura
 - ingredientes da receita do bombom (acho que dá para \cong 10 crianças).
 vasilhas grandes para serem misturados os ingredientes.
 Neste momento não sei como é a escola, se a escola pode arrumar os ingredientes, ou pode ser pedido para os alunos trazerem ou até vocês terem que comprar.

Exemplo de algoritmo para fazer bombons

Ingredientes

1 lata de leite ninho
 1 lata de leite condensado
 1 lata de nescau

Algoritmo:

- Se as mãos estiverem sujas então lavar as mãos;
 - Abrir as latas;
 - Colocar em uma vasilha grande os ingredientes;
 - Enquanto a mistura não for homogênea misturar os ingredientes;

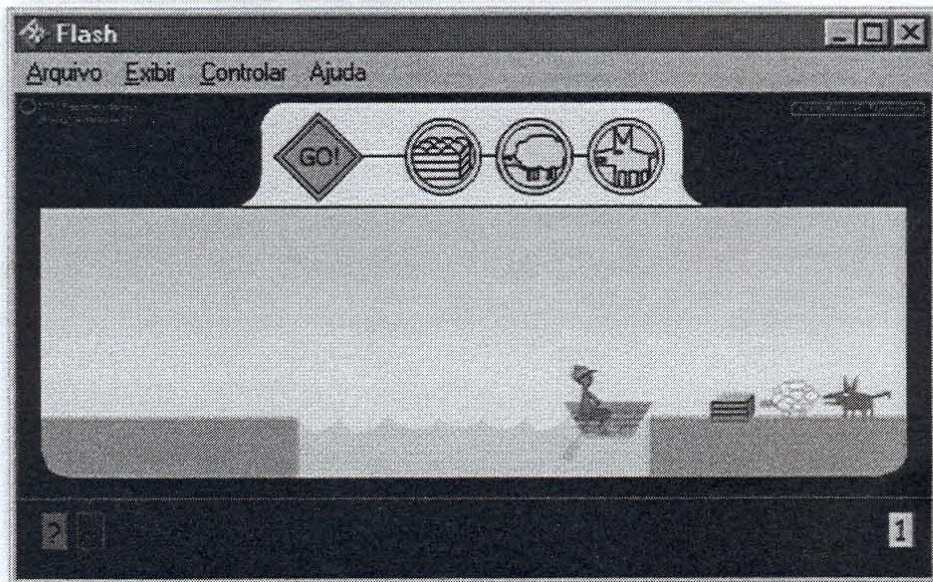
- Fazer bolinhas com as mãos.

Sugestão:

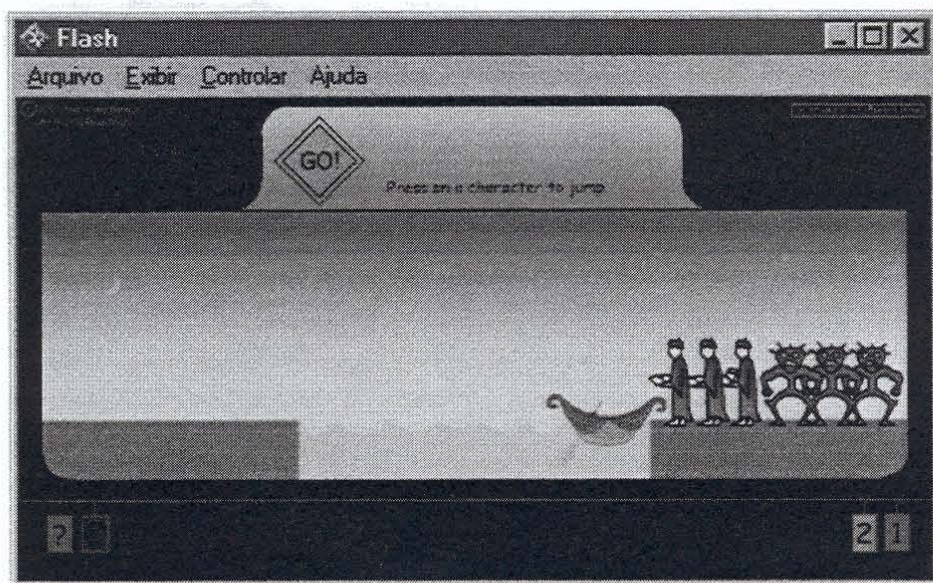
- Dar uma explicação inicial do que seja um algoritmo e distinguir a diferença entre construí-lo e executá-lo. (se quiserem item 4.1 e 4.1. do TI, páginas 11 e 13).
- Falar que um algoritmo pode ser escrito de várias formas e uma das formas pode ser o anexo 8 (ferramenta Word). Em meu encontro as minhas crianças construíram o papel de carta e completaram o algoritmo para inserir cabeçalho e rodapé, porém se vocês acharem que as crianças já sabem, só exemplifiquem que o anexo 8 é uma forma de escrever algoritmos.
- Dividir os alunos que podem ser em grupos e devem através da receita acima tentar fazer seus bombons.
- Depois da atividade devem ser reforçados os conceitos vistos do algoritmo e pedir para os alunos identificarem onde está a construção ou execução do algoritmo. Também pedir para identificarem as dificuldades que tiveram para fazer o bombom: *ambiguidade* (Quais ingredientes? Todos? Qual a quantidade de mistura para fazer as bolinhas?), *finitude* (misturar até quando?), se seguidos conduzem à solução de um problema.
- Também deve ser identificado o que seriam primitivas de um algoritmo.
- Mostrar o arquivo da construção do pássaro e do balão. Pedir para construírem o balão (o pássaro nós não conseguimos, mas pode ser tentado que também serve para identificar a necessidade de primitivas).
- Pedir às crianças para *construírem* seus algoritmos para construção de dobraduras que eles sabem (navio, chapéu, copo, etc.). Elas podem usar as primitivas definidas no balão ou definirem suas primitivas (com papel/lápis ou no computador). Após podem ser trocados os algoritmos e eles podem trocar os algoritmos para que cada um execute o que os outros *construíram*.

Anexo 24 Produtos de software abordando lógica

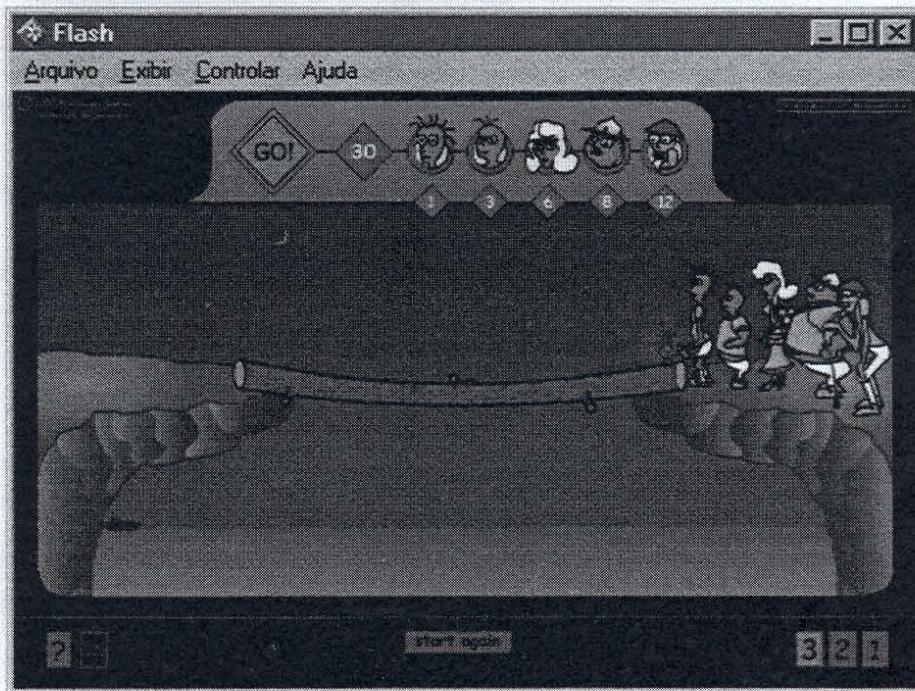
Software para um homem atravessar o rio com suas cargas: maço de alfafa, ovelha e lobo. No barco só pode ir o homem e uma de suas cargas, deve-se tomar cuidado para não perder nenhuma carga.



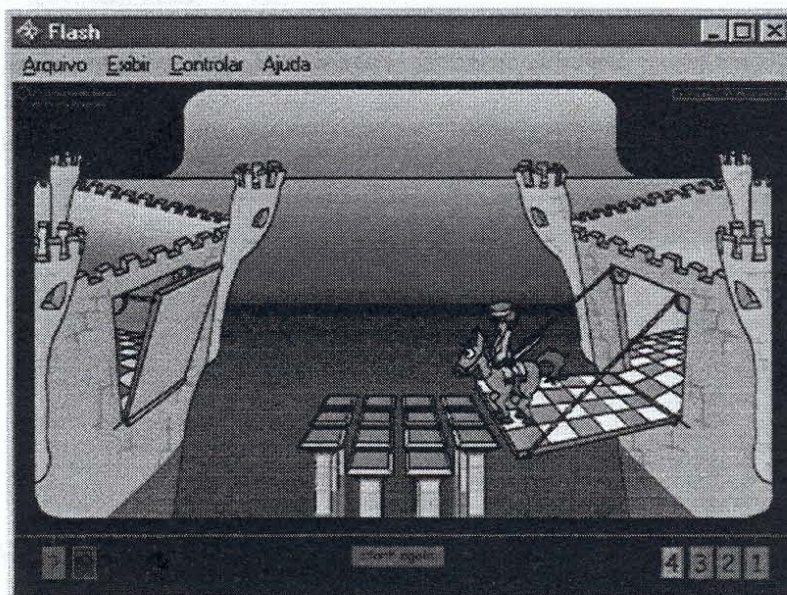
Software para 3 jesuítas e 3 canibais atravessarem um rio. No barco só cabem 2 e não se pode deixar um número maior de canibal em nenhuma margem.



Software para 5 pessoas atravessarem uma ponte. Nesta travessia pode passar 2 pessoas ao mesmo tempo sempre com a lamparina. Cada pessoa possui um peso e um tempo que leva para atravessar o rio. A ponte agüenta um determinado peso e tem-se o tempo máximo que deve durar a travessia.

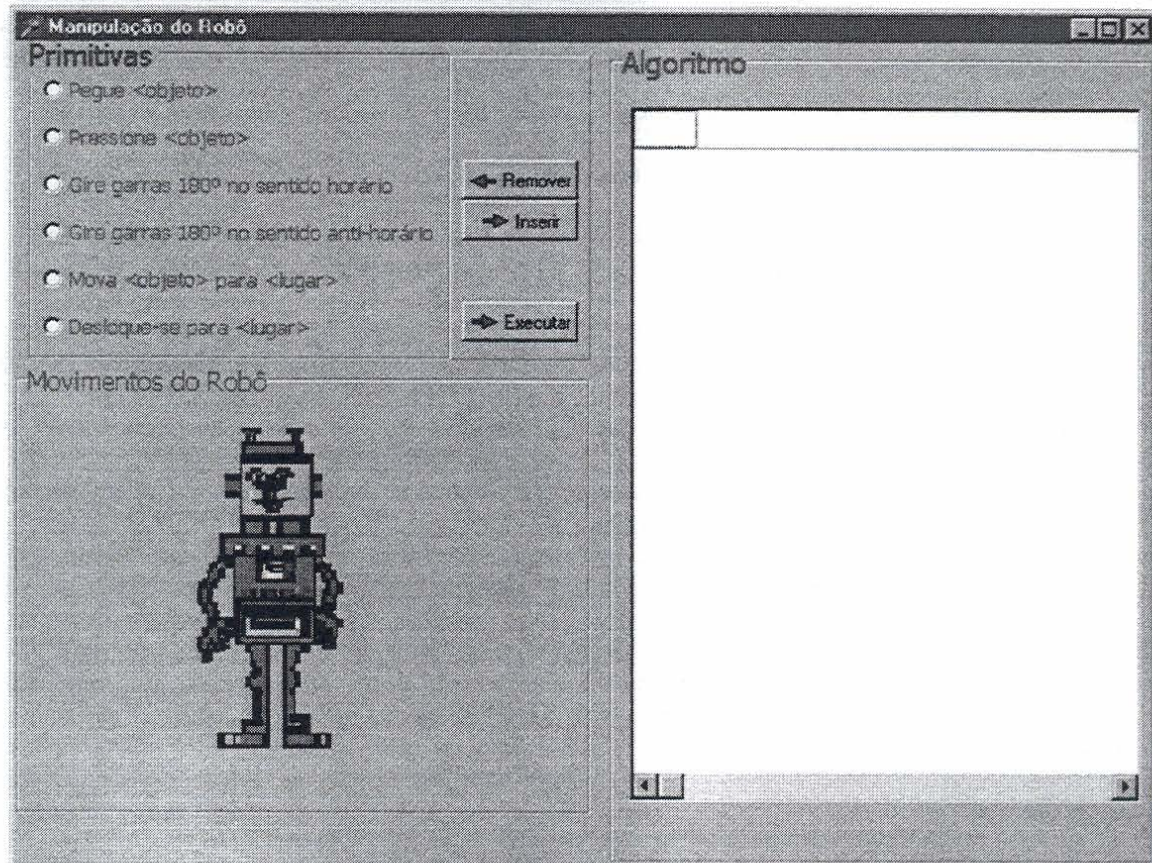


Software para atravessar de um lado para outro. O cavalo anda em L e cada quadrado que o cavalo pisa abaixa a ponte, porém este quadrado abaixa e perde-se o impulso para acessar a ponte.



Anexo 25 Software construído para o exercício 1 do anexo 7

Software de simulação em que se escolhe as primitivas e monta-se o algoritmo e depois pode-se simulá-lo.



Anexo 26 Artigo ICECE'2000

A Propose of Teaching Computer Science for Children*

Cláudia Santos Fernandes, Paulo Blauth Menezes, Fernando Accorsi

Abstract — This paper work shows researches about the use of Educational Computers on schools, the identification, through college teachers, of the fundamental concepts of the Computer Science and a research about what children would like to learn of Computer Science. From the results considerations are made to apply the teaching of Computer Science for Children.

Index Itens— Educational computer on the schools, teaching of the Computer Science for children.

I. INTRODUCTION

Considering the usage of computers and all its technologies involved, the school as a member of the total constitution process of the human being is looking forward to use these new technologies to improve the teaching procedures.

Several efforts are being developed with the meaning of apply this technology to be used on the teaching procedures. These efforts are related on the utilization and classification of the educational software products, on techniques of high distance communication, on the usage of multimedia resources or virtual reality to support the content applied by the teacher and also on the constitution of computer environments that may allow a educational change. However the known works on these efforts do not include the basics concepts involved on Computer Science.

According with [1] the computer theory is very important to the Computer Science, because it gives a right theory support requested to a correct and great learning of the science involved on computers, creates the development of a logical and formal thinking, and also, introduces fundamental concepts that are developed in other areas.

Within another aspects involved on the learning process, the development of the logical thinking is fundamentally important, that way is proposed that the teaching of the fundamental concepts of Computer Science can improve the learning in school and develops skills that are not usually requested.

This article is organized in three sections where considerations were made and detailed about the researches with teachers, in schools and with the children.

II. WHAT MUST BE TEACHED ON COMPUTER SCIENCE FOR CHILDREN

To know which topics of computer science should be teach to children, a questionnaire was made with teachers of Computer Science College, System Information College and Data Processing College from the fowling institutions: UFRGS (Universidade Federal do Rio Grande do Sul), UEM (Universidade Estadual de Maringá), USP-São Carlos (Universidade de São Paulo – Campus São Carlos), UNOESTE (Universidade do Oeste Paulista), UNIPAR (Universidade Paranaense) UFPel (Universidade Federal de Pelotas) e UNOPAR (Universidade Norte do Paraná).

A. Research with the teachers

The teachers that participated act on several subjects, such as: Software engineering, Data Resource, Programming language, Math, General System Theory, etc.

It was requested that the questions were answered on an objective way what are the fundamental concepts of Computer Science and they are shown on table I exactly how they were answered.

TABLE I
RESEARCH WITH TEACHERS

Subject where act	Identified concepts
1. Fundaments of System Information.	Semantic thinking
General System Theory	Knowledge of information technology. Fully knowledge of information technology.
Project Management	
System Information Topics I and II	Communication and team work

* This work is partially supprted by: FAPERGS (Project WaP_For), CNPq (Projects HoVer-CAM, GRAPHIT) and CAPES (Project TEIA) in Brazil.

Development Skills	Production Agent, Dissemination of Technology Research
2. Computer Architecture Digital Circuits Logic Discrete Math	Computer Architecture Software Engineering Data Storage Graphical Computer
3. BD Fundaments BD Project	Logic and Algebra Computer Theory, computability Programming Languages and algorithms Formal Language Data Structure, file structure, BD Operational Systems
4. Algorithms and Programming Techniques I and II Data Structure Computer Tools I Basic Informatics	Logic for development Data Transmission Data Structure (Types and usage) Saving Data Computer Math Search Methods (Information access)
5. General System Theory Administration Introduction Development Skills Organization and Methods	Systems Organizations Algorithms Hardware/software Programming Language Data/Info
6. Algorithms and programming techniques I Fundamental English Basic Informatics	Information Formalism Logic Semantics Batch and on-line Data Transmission
7. Lab. I and II Grafos Theory Supervised Trainee	Programming Language Computational Logic Math
8. Ordination and Research Data Structure Computer Programming I	Algorithms Universal Machines Computability Computed Functions Formal Languages
9. Formal Language e Automates	Program Machine Grammar Language Automates
10. Programming techniques and methods Algorithms and Data Structure Formal Language and Automates	Formal Language (Generators Recognizers) Computability/Algorithms Data Structure (Lists, Diagrams)
11. Info Lab.(History, hardware concepts, software, SO, text editor) Computer Tools (Delphi) Supervised Trainee	Math Knowledge Logic Concepts Algorithms Concepts (principal)
12. Computer Programming II e IV Computer Tools I TTC Project and Supervised Trainee	Scientific Initiation Not Commercial Software Project Programming Technique Software and Hardware Architecture
13. Distributed System Business Model Distant Teaching	Algorithms Conceptual Modeling Abstraction Formal Logic Data Structure System Structure
14. Software Engineering SGDB Graduation Project	Algorithms Data Structure Automates and Languages Centralized and Distributed Systems Man-Machine Interface Hiperdocuments e Multimedia System Information Software Engineering
15. Software Engineering Programming Language	Algorithms and its Studies: complexities, etc.

16. Programming III Data Storage	Programming Logic Math Concepts Operational Systems Nets
17. Numbering Computers Computer Theory High-Performance processing	Logic e discrete Structure Algorithms Theory Computability & Complexity Theories Formal Methods Computer Technologies
18. Software Engineering	Algorithms e Complexities Abstraction Abstracts Types of Data Modeling

The concepts identified were separated in relation to the amount of answers and organized on table II which has only the concepts that were identified for more than one teacher.

Some criteria were applied as: where had language, was rearranged to the concept Programming languages and the concept Algebra was placed with Math just like Computer Logic of Development and Algebra that are placed only on the logic concept and the concept identified as data transmission is placed with nets.

TABLE II
CONCEPTS IDENTIFIED BY THE TEACHERS AND ORGANIZED BY THE AMOUNT OF ANSWERS

	Concepts	Amount of answers
1.	Algorithms	9
2.	Logic	8
3.	Programming Languages	5
4.	Computability	4
5.	Data Structure	4
6.	Math and Algebra	4
7.	Formal Languages	3
8.	Program	3
9.	Nets	3
10.	Abstraction	2
11.	Automates	2
12.	Data Storage	2
13.	Complexity	2
14.	Data/Info	2
15.	Formalisms	2
16.	Machines	2
17.	Conceptual Modeling	2
18.	Distributed Systems	2
19.	Operational System	2

B. Considerations

Algorithms was the most voted concept, followed by Logic. Program Language, Computability, Data Structure Math and Algebra are next.

Based on these votes can be realized that these concepts can be reorganized in three groups:

- Fundamental Math: Logic; Abstraction; Math and Algebra.
- Info Theory: Algorithms (concept); Program (concept); Formal Language (and automates); Computability; Complexity; Formalisms and Machines (concepts).
- Applicable Info: Algorithms (Program Techniques); Program (Practice Subject); Program Language; Data Structure; Nets; Data Storage (data/information); Conceptual Modeling; Machines (Computer Architecture); Distributed Systems and Operational Systems.

This group can also be found on the list of basic subjects requirements for Computer Science schools proposed by the Brazilian government [5], where this basic subjects requirements are separated on 4 different formation fields:

- Basic Formation: which belongs the basic steps of this field, the Computer Science, the Math requested to formally define them, the physics and electricity requested to allow the understanding and the projects of technically usable computers and the pedagogic formation that introduces the basic knowledge of teaching Computer Subjects.
- Technical Formation: which applies the basic knowledge on the technical development of computers.
- Complementary Formation: which allows an integration of the courses subjects with other professionals and the

- Humanistic Formation: which gives the view of a social and human dimension.

About these requirements, the Computer Science is the most important area of the composition of the courses requirements, its central point are on the concepts of machine algorithms and its sub-areas are: Programming, Computer and Algorithms, Computer Architecture, Math, Physics, Electricity and Pedagogic.

III. WHAT IS BEING MADE?

To know what is being made on the teaching process of Computer Science and improve the lately scenario of Presidente Prudente-SP-Brazil, a field research was made on private schools of Londrina-PR and Presidente Prudente-SP with a questionnaire to the responsible people of the Educational Computer Area . This research included the aspects of the utilization of computers only on the teaching process, of the hardware and software resources, on the same way the classes are teach and how is made the integration of computers with the lately scenario of this schools

A. *The research of schools in Pres. Prudente-SP and Londrina-PR.*

Between the schools analyzed, 75% works on a traditional way.

They have Computer Labs. with a lot of computer resources, such as: printers, scanners, multimedia resources, intranet and internet.

On these labs. The classes are teach by computer professionals, periodically through almost all the grades of fundamental and intermediate schools, according to the school's plans.

The classes are always separated of the rest of the subjects. It consists on teaching basic computer tools, as text editors, electronic tables and products for multimedia presentation. Joining the teachers of basic subjects they work with products related to educational meanings, going through some subjects of the school requirement.

Some schools work on a little different way and have some complementary considerations.

One of the schools uses only the computer , either to the teachers as for the students on the creation of Home Pages. The teachers had some training each teachers builds the Home Page of your own subject. The students also create their own pages and the teachers use the best pages on their subjects.

Another visited school had the idea of installing computers on the classroom, to the teacher use it during the presentation classes, making the computer a instrument of research and support for illustration of their subject.

The last school that works on a different way does not have basic training, because they believe that the student should already know how to use the computer. The student is registered to the classes, and the computers are used as a support on the assembling of robots, for example: a robot that moves itself with solar energy. There are software products with educational meanings, but they are not so used. Who choose these products is the computer professional. This programs are chosen because they are simulation programs, where a challenge is proposed for the student.

B. *Research on Brazilian schools.*

On a research made by the newspaper "Folha de S.Paulo" em [3] about the structure of the Brazilian schools it was verified that in 1997 only 28% of the schools had computer labs. And going to 54% in 1999.

This same research denotes that the intermediate schools have better equipment than the fundamental schools, but, however, show numbers that proves the Brazilian level on educational technology: only 22% of the schools have internet.

The research concludes that the intermediate school level, that, according to the requirements of the Brazilian government the student should be prepared to work or go to college, few public schools (5%) and almost half of the privates offer the opportunity of this students use the internet.

C. *Considerations*

According to [4] the teachers can improve the use of computers and teach on a technology way the students, and , at the same time make sure that the information technologies are used to develop and improve the learning procedure. On such a way that the alphabetization on information technologies improve the critical sense and make the students able to learn not only how to use these technologies but to understand how it works.

This view is used by several authors such as on [2] and [6] where is found that the traditional methods for teaching Math make the student waste too much time on calculus instead of making concepts and develop a abstracts thinking.

Therefore, in Math is teach how to handle a calculator, the Computer Science teaching can not limitate itself for a training procedure or even teach a computer tool such as a specific text editor, because if a change happens on the editor type, this training must be restarted.

IV. RESEARCH WITH CHILDREN

The last research was performed with 25 children, between 9 and 11 years old, students of private and public schools.

After questions about the name, age and grade of each child a talk was performed with the children about what did they think about the school, their preferences about the subjects and what did they like to do when they were not at school. After it was asked what did they wanted to learn from computer and Computer Science.

To the children be able to distinguish between to concepts, it was given to examples:

- "When the teacher teach calculus does she teach how to use a calculator?"
- "About a car, can be learned how to drive it or how to build it?"

Besides this private talking it was asked for them to draw a environment that they think it was nice to learn Computer Science.

The children were placed on groups of (5 to 10) on a large table with material for drawing. Each one made a draw according to their wish.

A. Considerations.

The answers about what did they like to learn with computer was: learn how to play, internet chats, type texts, draw, and learn about subjects of the school. And when asked about Computer Science they answered: build a computer understanding its components and build their own games.

During the talk it can be realized that the children were impressed to know that they can make their own games and they can make that with computers. However one of the children added that only wanted to build a computer with someone else.

The environment imagined by the children is full of happiness, has furniture, windows and flowers. On the computer monitor there is always a colored draw and a child smiling.

In some draws was noticed that some children have a larger notion of how the computer works, because some of the draws had computers plugged on the electricity.

It was verified very similar draws on its structure, concluding that several children prefer to only copy the friend's draw.

CONCLUSION

The data collected with the teachers on the institutions researched shows the basic requirements for Computer Science, what mainly denotes the concepts related to the algorithms notion.

The research with the schools reflects the environment that involve the computer teaching on Presidente Prudente-SP and Londrina-PR schools, where a great number of are limited to the current technologies. This research was extended to some other regions of the country with the porpoise of giving a national status.

The research with children denotes nice perspectives of the Computer Science learning and the positive perception of the computer influence on their routine.

With the improvement of researches on this field, in the schools and also with teachers and students, is pretended to define the basic contents of the Computer Science teaching procedure and the ways to use it in relation of the already existing subjects.

ACKNOWLEDGMENTS

The authors would like to thank psychologist Regina Gioconda for the orientations with children and interpretation of the research data; to Roges Grandi and Julio Pereira Machado for the great suggestions; and Marco Eduardo Cordeiro for his help with the earlier version of this paper.

REFERENCES

- [1] T. A. Diverio and P. B. Menezes, *Teoria da Computação: Máquinas Universais e Computabilidade*, 1st ed., Porto Alegre: Sagra Luzzato, 1999.
- [2] A. A. Esmín, “O que um Aluno de Ensino Médio deve Conhecer de Computação? Uma Proposta de Currículo”, in: *Anais do XIX Congresso Nacional da Sociedade Brasileira de Computação*, Rio de Janeiro, July, 1999, pp. 787-793.
- [3] “Infra-estrutura das escolas de ensino médio”, in: *Journal “Folha de São Paulo”*, São Paulo, may 1, 2000, pp. 4-3.
- [4] A. C. L. Jurema and M. O'Rourke, “Critical consciousness and Information Technology (IT) – An International Approach to Developing IT Literacy in Scho Twenty-two points, plus triple-word-score, plus fifty points for using all my letters. Game's over. I'm outta here.ols”, in: *Anais do VIII Simpósio Brasileiro de Informática na Educação*, São José dos Campos, November, 1997, pp. 531-541.
- [5] MEC, *Diretrizes Curriculares de cursos da área de Computação e Informática*. Disponível por www em http://www.mec.gov.br/sesu/ftp/cur-diretriz/computacao/co_diretriz.rtf
- [6] F. A. Silveira and A. C. R. Costa, “O MATHEMATICA em uma experiência com Alunos de 2º Grau”, in: *Anais do VIII Simpósio Brasileiro de Informática na Educação*, São José dos Campos, November, 1997, pp. 849-852.

Anexo 27 Artigo SITE'2001 – AACE

Computer Science for Children

Cláudia Santos Fernandes
Faculdade de Informática de Presidente Prudente
Universidade do Oeste Paulista
Brazil
claudia@apec.unoeste.br

Paulo Fernando Blauth Menezes
Departamento de Informática Teórica
Instituto de Informática
Universidade Federal do Rio Grande do Sul
Brazil
blauth@inf.ufrgs.br

Abstract: This work shows a propose of teaching computer science for children. It is intended that the “Information Technology Teaching” sharpens the critical sense and capacitate students to learn not only to use currents technologies, but to understand how it works. To find out which Computer Science Topics are more suitable to teach children, it was realized a quiz, answered by Computer Science and Information Technology teachers which thought disciplines. A solid basis to the practice of Computer Science has the objective to develop the demanded abilities to think, to express clearly and precisely, to answer problems and create concepts.

Introduction

Many efforts are being developed to use today’s technologies into the teaching-learning process. These efforts are aimed at the use and classification of educational software products, distance communication techniques, the use of multimedia and virtual reality resources to aid teaching, and also in the construction of computer environments that could change the educational paradigm, though from all known papers up until know there is none focusing on the basic Computer Science concepts involved.

Between all the aspects involved during the learning process, the development of logical reasoning has fundamental importance, thus it is proposed that the teaching of fundamental concepts of Computer Science could enhance the basic school learning and develop today’s demanded abilities.

This article is organized in three sections: the discussion of teaching concepts, the aspects of work in the teaching of computer science or children and the final conclusions.

Teaching Concepts

According with Menezes and Diverio (1999) the computer theory is very important to the Computer Science, because it gives a right theory support requested to a correct and great learning of the science involved on computers, creates the development of a logical and formal thinking, and also, introduces fundamental concepts that are developed in other areas.

To find out which Computer Science Topics are more suitable to teach children, (Fernandes and Menezes, 2000) it was realized a quiz, answered by Computer Science and Information Technology teachers which thought disciplines like: Software Engineering, Data Bases, Programming Languages, Math and General Systems Theory, in many public and private institutions of Brazil.

It was asked, independent of the application to answer in an intuitive and objective way which are the basic fundaments of Computer Science. The identified concepts where separated and summed in relation to the quantities of answers.

From all the concepts cited, the top ones where: Algorithms, Programming, Formal Languages and Automats, Computability, Complexity, Formalisms and Machines. It is noticed that those comprehend the basic formation in the Computer Science area.

It is intended that the “Information Technology Teaching” sharpens the critical sense and capacitate students to learn not only to use this technologies, but to understand how it works.

This vision is corroborated by many authors such as on (Esmín, 1999) and (Silveira, 1997) in which there is evidence that the traditional methods of teaching math bring the student to use most of the time to elaborate calculations as opposed to build concepts and develop abstract reasoning.

Therefore, as math does not teach how to manipulate a calculator, the teaching of Computer Science cannot be limited to be a training or to teach how to use a tool, like a specific text editor. Because if something changes in this type of editor this training must be restarted. Additionally as it does not fit in this context, the teaching of typing and shorthand writing.

Aspects of Work

After identified the fundamental concepts of computer science it is necessary to study how they will be introduced and how to approach them in conformity with the existing disciplines. We are studying on the history of education and how today's disciplines were introduced to the actual school curriculum for to be based

History is the understanding of the transformations made by men during time. Pedagogy is the art and the science of teaching, or the action of men when transmitting its knowledge. It is focused on the man in its whole and it is centered on the formation of thought. Thus Pedagogy in the teaching of Computer Science can help the formation of computational thought, use other's science methods, and identify and eliminate learning obstacles. The path to the teaching of Computer Science must be worried on the solution of problems and must be centered on the student's experience to act in a specific knowledge domain.

Understand and study the history of computer is before all understand the evolution of mathematics. The result of building techniques that allowed us to calculate in the past, for example, area and tangent coincides with the technological development of mankind. Thus the first year of Computer Science must be based on the teaching of mathematics, for example: basic in logical math and conditional idea of Computer Science.

Conclusions

The data collected with the teachers on the institutions researched shows the basic requirements for Computer Science, what mainly comprehend the basic formation in the Computer Science area.

These concepts must be worked in a multidisciplinary matter, focused in mathematics. Thus, algorithms are used for development of calculus, logic's notion, of theory of set etc.

The transformation of these concepts in useful applications is of fundamental importance, bringing motivation to children for the computer science's study.

References

- Boyer, C. B. (1991). *A History of Mathematics*. Broklyn, New York: John Wiley & Sons.
- Diverio, T. & Menezes, P. B. (1999). *Teoria da Computação: Máquinas Universais e Computabilidade*. Porto Alegre, RS: Sagra Luzzato.
- Fernandes, C. S. & Menezes P. B. (2000). A Propose of Teaching Computer Science for Children. In: *International Conference on Engineering and Computer Education*. São Paulo.
- Esmín, A. A. (1999). O que um Aluno de Ensino Médio deve Conhecer de Computação? Uma Proposta de Currículo. In: *Anais do XIX Congresso Nacional da Sociedade Brasileira de Computação*. (pp. 787-793). Rio de Janeiro, 1999.
- Silveira F. A. & Costa A. C. (1997). O MATEMATICA em uma experiência com Alunos de 2º Grau. In: *Anais do VIII Simpósio Brasileiro de Informática na Educação*. (pp. 849-842). São José dos Campos, 1997.

Anexo 28 Artigo SBC'2001 – WIE

Metodologia do Ensino de Ciência da Computação: uma Proposta para Crianças

Cláudia Santos Fernandes¹, Paulo Blauth Menezes²

¹ Faculdade de Informática de Presidente Prudente - Universidade do Oeste Paulista (UNOESTE) - Rua José Bongiovani, 700 - 19.050-680 - Presidente Prudente –SP

² Instituto de Informática – Universidade Federal do Rio Grande do Sul (UFRGS) -Caixa Postal 15.064 – 91.501-970 – Porto Alegre – RS – Brasil

claudia@apec.unoeste.br, blauth@inf.ufrgs.br

Resumo. Devido à disseminação dos computadores e de seu uso como parte da formação dos alunos, muito tem-se discutido sobre a união entre os recursos tecnológicos e a educação. No entanto, não adianta tornar disponível toda esta tecnologia se rapidamente são abandonadas e são substituídas ferramentas e técnicas dentro da Computação. Por isso, mais importante do que o ensino das tecnologias correntes é o ensino dos conceitos fundamentais da Ciência da Computação. Estes conceitos além de proporcionar um embasamento teórico para entendimento da ciência envolvida na computação, também propiciam o desenvolvimento de um raciocínio lógico e formal e de habilidades que são exigidas no mundo atual. Este trabalho apresenta teoria e exercícios e fornece subsídios para o ensino dos conceitos fundamentais da Ciência da Computação para crianças.

Abstract. Due to the dissemination of computers and their use as part of students' formation, very discussion about join between technology recourses and education. However, it's not enough to leave disposable all technology if tools and technics are quickly abandoned and substitute in the computation. Therefore, more important that teaching current technologies is teaching Computer Science fundamental concepts. These concepts beyond propitiates an adequate theoretic base needed to understanding of the science involved in computers, and also, proposes the development of a logical and formal reasoning and develop skills nowadays required. This work presents theories and exercises and provide subsidy for teaching fundamental concepts of Computer Science to children.

1. Introdução

Atualmente, o avanço tecnológico é vertiginoso, sendo mais tentador as pesquisas dentro da área de Ciência da Computação que se concentrem em termos de software e hardware. Porém, segundo [Fonseca 1998] pode-se observar que rapidamente se abandonam e se substituem ferramentas e técnicas dentro da Computação.

Também é observado que a inclusão da informática no processo educativo, atualmente vem sendo direcionada na utilização e classificação de produtos de software educacional, em técnicas de comunicação à distância, no uso de recursos de multimídia ou de realidade virtual para apoio ao conteúdo abordado pelo professor e também na construção de ambientes computacionais que possam proporcionar uma mudança de paradigma educacional, porém os trabalhos que se tem conhecimento não abordam os conceitos básicos envolvidos na Ciência da Computação.

Pesquisas voltadas à área educacional por pessoas da área de Ciência da Computação são importantes devido a dois panoramas: a nova geração de crianças que surge já incorporou o uso das tecnologias correntes e estas pesquisas servirão de base e referência para que esta geração relacione conceitos entre a Computação e outras ciências como a Filosofia, as Ciências Cognitivas e principalmente com a Matemática.

As motivações para este trabalho estão na convicção de que o ensino dos princípios básicos das áreas da Ciência da Computação é mais importante do que o ensino das tecnologias correntes e que o ensino dos conceitos fundamentais da Ciência da Computação pode melhorar o aprendizado escolar e desenvolver habilidades que são exigidas no mundo atual.

O desenvolvimento deste trabalho está apoiado na seguinte seqüência. Primeiro são apresentadas pesquisas de campo sobre o uso da Informática Educativa em escolas; a identificação, através de professores, dos conceitos fundamentais de Ciência da Computação e uma pesquisa sobre o que crianças gostariam de aprender de Ciência da Computação. Uma vez identificados os conceitos fundamentais, foram feitos estudos e propostos teoria e exercícios nos itens algoritmos, programas, máquinas e autômatos, procurando, sempre, utilizar-se de uma linguagem simples e clara. Por último são apresentados alguns dos resultados parciais e sugestões para continuação deste trabalho.

2. Identificação dos Conceitos Fundamentais da Ciência da Computação

Com o objetivo de definir quais conteúdos devem ser trabalhados com as crianças, foram realizadas três pesquisas de campo. Os dados coletados com os professores nas instituições pesquisadas sinalizam o conteúdo fundamental para a Ciência da Computação, o qual prestigia conceitos relativos a noção de algoritmo. A pesquisa com as escolas reflete o ambiente que envolve o ensino de Informática, na região de Presidente Prudente – SP e Londrina – PR, onde em grande parte delas está limitado ao ensino de tecnologias correntes. A pesquisa com as crianças demonstra expectativas favoráveis ao aprendizado da Ciência da Computação e a percepção positiva da influência do computador em seu cotidiano. Estas pesquisas estão detalhadas em [Fernandes 2000].

3. Conceitos a serem ensinados

Identificados os conceitos fundamentais da Ciência da Computação faz-se necessário um estudo de como serão introduzidos estes conceitos e as formas de abordá-lo estabelecendo relações com outras disciplinas, especialmente a matemática, exemplo: noções de lógica da Matemática e noção de condicional da Ciência da Computação.

O caminho para o ensino de Ciência da Computação deve ter como preocupação a solução de problemas e deve ser centrada na experiência do aluno e atuar em domínio específico do conhecimento, pois a lógica da ciência da computação tem relação forte com a lógica do cotidiano. Os exercícios de trabalho devem ser simples, claros, com exemplos práticos do cotidiano, que são mais motivadores que aplicações técnicas. O ambiente imaginado deve apoiar-se em um trabalho cooperativo e reflexivo.

Segundo [Brookshear 2000], um *algoritmo* é um conjunto de passos que definem como uma tarefa é executada. Uma vez que se tenha o algoritmo, deve-se ter uma forma de representá-lo para que a *máquina* entenda, assim, os algoritmos, em Ciência da Computação, são representados através de *programas*. Esta representação deve ser através de um conjunto de comandos facilmente compreensíveis e sem ambigüidades. Existe uma grande diversidade de esquemas para a representação de

algoritmos, conhecidos como *linguagens de programação* e são formalizadas de duas maneiras: gramática (*mecanismos de geração*) e reconhecedores (*autômatos*).

A busca de algoritmos para controlar tarefas cada vez mais complexas conduz à investigação da existência ou não de algoritmos que solucionem determinados problemas. Segundo [Divério 1999] o objetivo do estudo da solucionalidade de problemas é investigar a existência ou não de algoritmos que solucionem determinada classe de problemas. Ou seja, investigar os limites da *computabilidade* e, conseqüentemente, os limites do que pode efetivamente ser implementado em um computador. Assim, se não existir um algoritmo que leve à solução de um problema, este problema não poderá ser resolvido por uma máquina, portanto são problemas não computáveis. A seguir são trabalhados os conceitos de forma simples e acrescentados exemplos de como foram apresentados às crianças.

3.1. Algoritmos

O estudo de procedimentos efetivos é um dos principais temas da Ciência da Computação. Em [Sagatume 1986] afirma que, em geral, todo problema computacional se reduz a busca de um procedimento, que pode ser escrito através de modelos matemáticos. A palavra algoritmo na matemática [Fonseca 1998] designa um procedimento geral de cálculo, que se desenvolve, por assim dizer, automaticamente, poupando-nos esforço mental durante o seu curso. A máquina teórica de Turing é aceita como um modelo formal de algoritmo, através dela, pela primeira vez, se deu uma definição do que significa computar algo. Assim, por se tratar de um conceito que marca fortemente a Computação e também é utilizada para estudar a capacidade dos processos algoritmos até hoje em dia, sua definição e funcionamento são abordados no item 3.3.

As características principais dos algoritmos em [Brookshear 2000], [Forbellone 1996] e [Guimarães 1985] são: não podem ser ambíguos, devem ser finitos e se os passos definidos forem seguidos deverão conduzir à solução de um determinado problema. Portanto, não se tem a preocupação de fornecer o resultado da solução do problema. Os exemplos clássicos deste fato são as receitas de bolos. Quando se possui uma receita de bolo, temos os passos para que seja possível obter o resultado final: o bolo. Nenhuma receita traz o bolo pronto.

3.1.1. Lógica

Uma primeira etapa para o ensino de algoritmo para crianças deveria ser a introdução do conceito de *lógica* que de acordo com [Forbellone 1996] é a matéria-prima para elaboração de algoritmos. Isto é verdadeiro, pois a lógica fornece os instrumentos mentais necessários para enfrentar qualquer tipo de investigação.

Segundo [Fonseca 1998], a lógica foi considerada na tradição clássica e medieval como instrumento indispensável ao pensamento científico. Da Ciência Lógica nasceu a Lógica Matemática e, dentro desta, várias filosofias da lógica que interpretam os cálculos simbólicos e sua sistematização axiomática. Para a Computação interessa abordar em particular a questão do pensamento dedutivo e matemático, seus limites, o problema da mecanização do pensamento quantitativo e o problema da Inteligência Artificial.

Neste momento são apresentados alguns filósofos em [Jostein 1995], suas contribuições e pensamentos, especialmente Aristóteles (384 a.C. – 322 a.C.), pois com ele é que se dá o verdadeiro nascimento da Lógica. Também é feito um relato de como era a vida das crianças nesta época, da forma como Aristóteles organizou os conceitos

do homem, de como criou a “Lógica” e de como estes conceitos podem ser usados em outras situações.

3.1.2. Álgebra

Segundo [Fonseca 1998] o termo álgebra vem do árabe al-jabr, ou literalmente, a "reunião de partes quebradas". Álgebra é a arte dos raciocínios perfeitos, é a ciência que fundamenta o desenvolvimento da Computação. Neste momento é apresentado às crianças um histórico de algumas máquinas e de seus respectivos funcionamentos, como: régua de cálculo, máquina de Pascal, a calculadora de Leibniz, a máquina analítica de Babbage. Os exemplos a seguir contêm exercícios usuais a alguns livros (figura 1) e exercícios sugeridos por Piaget em [Furth 1972] (figura 2).

Exercícios I. Lógica e aritmética (Autores desconhecidos)

I.1. Se sua mesada é R\$10,00 por semana e você gasta R\$8,00 por semana, quanto tempo demorará para juntar R\$20,00 em economia?

I.2. Tenho 1 caixa, dentro delas 2 outras caixas, dentro de cada uma delas, outras 2. Quantas caixas tenho?

I.3. Tendo 2 vasilhas de 5 e 3 litros, como consigo exatamente 7 litros de água?

I.4. O relógio A marca mais 12 minutos que o B, que marca menos 7 minutos que o de C, que tem atraso de 4 minutos sobre o D, que está atrasado 9 minutos. Um relógio certo marca 14:57. Quantos marcam os de A, B, C e D?

I.5. A caminhou 150 metros para a frente. Dobrou para a direita e caminhou 50 metros. Dobrou para a direita e caminhou 100 metros. Dobrou para a direita e caminhou 50 metros. A que distância está do ponto de partida?

I.6. A está à direita de B e à esquerda de C, que está à esquerda de D. Se E ficar entre A e B, e F entre C e D, qual a posição de C e D, qual a posição de B e D, considerando os que estão de um lado e outro de cada uma?



I.7. **A é mais alto 5 centímetros do que B, que é mais baixo 12 centímetros do que C e este mais alto 3 centímetros do que D. Qual a diferença de altura entre A e D?**


I.8. A teria mais 50,00 do que B, se obtivesse mais 30,00, porém, tem menos 40,00 do que C, que tem mais 80,00 do que D, que com mais 20,00, ficará com 150,00. Quanto tem cada um?


Figura 1. Exemplos de exercícios de lógica e aritmética (autores desconhecidos)


Todos estes exercícios devem ser trabalhados de forma prática, exemplos: onde é abordado dinheiro são utilizados dinheiro de mentira, montados pelas próprias crianças no computador e impressos; quando se tratou de quantidade de caixas e volume de água são utilizadas caixas de papel e vasilhas com marcações de volumes. Nos exercícios em são utilizadas as letras A, B, C, são substituídas pelo nome das crianças.

Exercícios II - Lógica símbolo-imagem (Piaget em [Furth 1972])

Exemplo: C →  ? → 

 ? → 

 ? → 

 ? → 

As respostas seriam S, A, M, C, respectivamente, mas são válidas outras respostas alternativas.

Figura 2. Exemplos de exercícios de lógica símbolo-imagem (Piaget em [Furth 1972])

O exercício inicia com a visualização de um exemplo e depois prossegue de acordo com a demonstração. Os exercícios prosseguem aumentando o grau de dificuldade. Todo o conjunto de exercícios se encontra em [Furth 1972], pág. 131 a 184.

3.1.3. Representação de Algoritmos

Para se escrever algoritmos é necessário saber qual a linguagem que será utilizada. Se for uma comunicação homem-homem utiliza-se a linguagem natural (português, inglês etc.). Também pode ser utilizada a linguagem das figuras, como em [Brookshear 2000] onde há o algoritmo para se construir um pássaro por meio de dobradura (figura 3).

Porém estas formas de comunicação podem conduzir a enganos, como na frase "O artesão produz formas perfeitas". Nesta frase poderíamos achar que o artesão produz esculturas com f(ó)rmas perfeitas ou fabrica f(ô)rmas para bolos.

No caso do algoritmo para a construção do pássaro é provável que alguns não conseguissem construí-los somente seguindo as instruções sendo necessário a definição precisa de uma linguagem, como sugerida em [Brookshear 2000] (figura 3). Estas definições precisas são chamadas de primitivas.

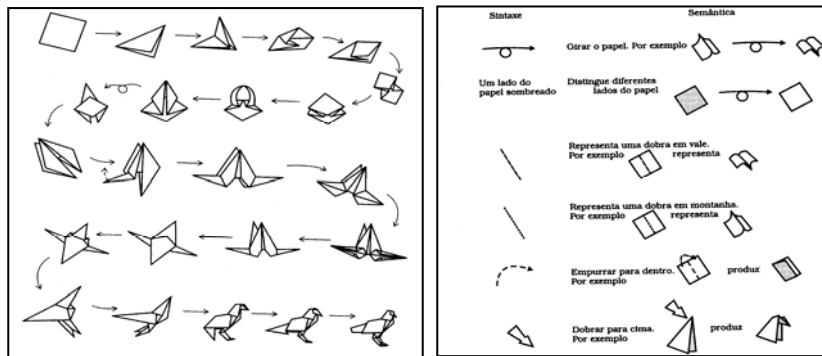


Figura 3. Algoritmo e primitiva de um pássaro de dobradura em [Brookshear 2000].

Exercícios para serem trabalhados com as crianças:

- Propor que, utilizando as primitivas da figura 3 e acrescentando novas primitivas, construam algoritmos para construir avião, barco, copo etc.;
- Utilizando as primitivas definidas para cada um dos problemas construa os algoritmos para:
- Em [Forbellone 1998], suponha que você possua um robô e queira fazê-lo trocar uma lâmpada, sendo que o mesmo foi programado para obedecer as seguintes primitivas:
 - *pegue <objeto>*
 - *pressione <objeto>*
 - *gire garras 180° no sentido horário*
 - *gire garras 180° no sentido anti-horário*
 - *mova <objeto> para <lugar>*
 - *desloque-se para <lugar>*
- e ainda é capaz de:*
 - *perceber quando algum comando não pode mais ser executado*
 - *sentir alguma fonte de calor*
- Três jesuítas e três canibais precisam atravessar um rio para tal dispõem de um barco com capacidade para duas pessoas. Faça o algoritmo utilizando as seguintes primitivas para atravessar todos, mas tome cuidado para que em nenhuma das margens a quantidade de jesuítas seja menor que a de canibais.
 - *Jesuíta entra no barco*
 - *Canibal entra no barco*
 - *Jesuíta sai do barco*
 - *Canibal sai do barco*
 - *Barco atravessa rio da margem direita para esquerda*
 - *Barco atravessa rio da margem esquerda para direita*
- Utilizando as primitivas abaixo, elabore um algoritmo que execute um jogo da velha.
 - *Enquanto todos os quadrados não estiverem ocupados e X ou O não tenham marcado três quadrados em linha*
 - *Escolha o <número do quadrado>*
 - *Se o quadrado não estiver ocupado, marque o quadrado <no.do quadrado> com <X ou O>*
 - *Se X ou O tem dois quadrados quaisquer em linha marque no terceiro desocupado*
- Utilizando-se de um computador e de ferramentas para edição de textos e figuras propor a construção de um algoritmo que ensine a fazer um papel de carta. Neste trabalho foram utilizados o editor de textos *Word* e a ferramenta *Paint* e foram criados algoritmos para esta finalidade.

3.2. Programas

Para aproximação com a linguagem usual da Ciência da Computação, o termo *primitiva* é substituído por *comando* que devem ter claramente definidos sua sintaxe e semântica. O conjunto de comandos utilizados para escrever algoritmos constitui uma *linguagem de programação*. Os algoritmos escritos utilizando os comandos de uma linguagem de programação são chamados de *programas*.

Devido a diversidade de máquinas e de linguagens de programação em [Divério 1999] a formalização de conceitos de programas e de máquina não são baseadas em qualquer linguagem ou computador real.

"A distinção entre programa e máquina é importante na Ciência da Computação, uma vez que o programa (ou algoritmo) independe da máquina e possui uma complexidade estrutural e computacional (quantidade de trabalho necessário para resolver o problema)". [Divério 1999] pág. 192

Em [Divério 1999] o conceito de programa é visto como um conjunto de operações e testes que são compostos de acordo com uma estrutura de controle. Os tipos de programas são determinados de acordo com esta estrutura e são três: Monolítico, com apenas desvios condicionais e incondicionais; Iterativo, possui estruturas de iteração de trechos de programas e Recursivo, baseado em sub-rotinas recursivas.

O programa do tipo *monolítico* é estruturado utilizando desvios condicionais (dependendo da resposta é executada uma tarefa) e incondicionais (após uma instrução segue-se para outra). Pode ser especificado na forma de texto, através do uso de instruções rotuladas e na forma diagramática: fluxograma. Neste trabalho somente são utilizados programas do tipo monolítico em forma de fluxograma. Não se trabalha com os programas do tipo iterativo e recursivo por três razões: o programa monolítico é o tipo mais fundamental de programas; apesar de não possuir mecanismos explícitos de repetição, pode-se simular uma repetição utilizando suas estruturas e recursão é um conceito importante e difícil de ser trabalhado por possibilitar níveis diversos de abstração, assim é recomendado como trabalho futuro.

3.2.1. Identificadores

Para se trabalhar com os programas é necessário o trabalho com os identificadores e para poder entendê-los e defini-los é necessário o entendimento da "Teoria dos Conjuntos". Existem vários produtos de software que abordam este assunto e podem ser utilizados como reforço sobre a Teoria dos Conjuntos. Neste instante são feitas relações entre identificadores e incógnitas, devendo ser muito bem trabalhadas as suas características: nome, conteúdo e o tipo. São sugeridos exercícios para definir o nome, o tipo e dar exemplos de conteúdos de identificadores (figura 4).

<i>Exemplo: Definir o identificador para o conjunto de nome das frutas</i>		
<i>Nome</i>	<i>Tipo</i>	<i>Exemplos de conteúdos</i>
<i>Frutas</i>	<i>Caracter</i>	<i>"maçã", "pêra", "goiaba"</i>
<i>- Definir o identificador para o conjunto de valores de sua mesada, para o conjunto dos valores das idades das pessoas, guardar a resposta para a pergunta: "Seu pai é advogado?"</i>		

Figura 4. Exercícios para definição de identificadores.

3.2.2. Comandos

Como definido anteriormente os comandos são as primitivas utilizadas para definir os algoritmos. Os comandos iniciais a serem utilizados para a construção de algoritmos

serão: atribuição e seleção. A figura 5 contém alguns exemplos de exercícios para o trabalho com o comando de atribuição.

Exemplo: Definir o conteúdo do identificador *Número* do tipo *Inteiro* após o comando de atribuição: $Número \leftarrow 2$

Número
2

Definir os conteúdos do identificador *Número* e *X* que são do tipo *Inteiro*, *Nome1* e *Nome2*, do tipo *Caracter* após os comandos de atribuição.

Comandos	Número	X	Nome 1	Nome 2
Número \leftarrow 10 X \leftarrow 5				
Número \leftarrow 8 - 3				
Número \leftarrow Número - X				
Nome1 \leftarrow Nome1 + Nome2				

Figura 5. Exemplos de exercícios utilizando o comando de atribuição.

Na figura 6 encontra-se o início de um exemplo de construção de um programa para conjugação de verbos (modo indicativo e vogal temática 'a') e na figura 7 são apresentados exercícios que através da utilização de fatos rotineiros e das estruturas de seleção e repetição devem ser desenvolvidos os programas.

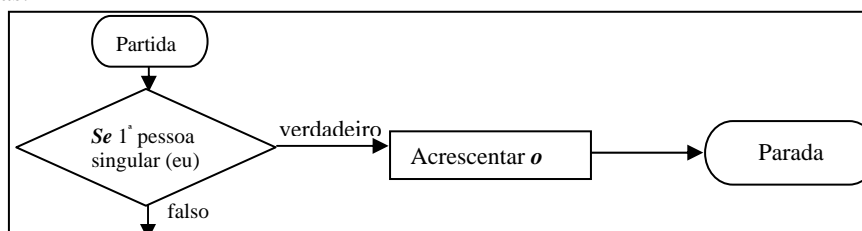


Figura 6. Início do programa para conjugação de verbos.

1. Construir um programa para eleger um representante da classe.
Obs.: - Irão ser fornecidos os nomes de 3 candidatos
- Fazer uma repetição para ir totalizando os votos de cada candidato
- Fazer uma repetição para ir totalizando os votos de cada candidato
- exibir qual candidato ganhou (fez o maior número de votos)
2. Construir um programa que descubra a qual signo do horóscopo pertence uma pessoa.
Obs.: - Irá ser fornecida a data de nascimento da pessoa
- Estender para: - descobrir o signo no horóscopo Chinês
- exibir as principais características de cada signo
3. Construir um programa para somar a quantidade de pontos de um jogo de boliche. Para cada partida ir pedindo os pinos derrubados para cada bola jogada (os pontos variam de 0 (canaleta) a 10 (strike))

Figura 7 – Exercícios utilizando os comandos de seleção e repetição.

3.3. Máquinas

Diversas pesquisas foram realizadas por diversos matemáticos (Boole, Frege, Peano, Hilbert, Newman, Gödel, Turing entre outros) com o objetivo de reduzir todo raciocínio a um processo mecânico.

O primeiro sistema lógico foi definido por Boole. Segundo [Fonseca 1998], foi quem forneceu uma idéia clara de formalismo, concebendo a lógica como uma construção formal à qual se busca posteriormente uma interpretação.

"Um sistema formal pode ser visto como uma espécie de "jogo" rigorosamente definido, que especifica regras para manipulação de símbolos. [Fonseca 1998]"

Na exemplificação de um sistema formal são utilizados os jogos de xadrez e de dama, pois devemos saber quais são as peças e qual o seu formato, os movimentos destas peças e como elas estão dispostas inicialmente no tabuleiro.

3.3.1. Máquina de Turing

A Máquina de Turing [Fonseca 1998] é um conceito abstrato que efetivamente deu início à era dos computadores baseou-se no princípio de que a simples aplicação de regras permite passar mecanicamente de uns símbolos a outros.

Turing analisou: Qual a forma utilizada por nós homens para executar um cálculo? Partindo de regras definidas, vai modificando os estados dos números até que se chegue à solução. Dependendo da quantidade de dígitos necessitamos guardar os estados intermediários, uma memória, que pode ser papel e lápis. A Máquina de Turing em [Hopcroft 1979] e [Papadimitriou 1999] é composta de uma fita dividida em células, uma cabeça que lê e grava dados na fita, um conjunto de regras que definem o movimento da cabeça e o estado inicial da fita, exemplificada na figura 8.

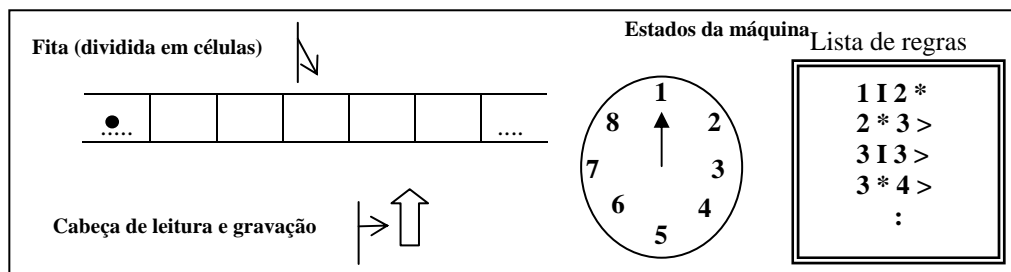


Figura 8. Componentes de uma Máquina de Turing.

Em [Fonseca 1998], há um exemplo específico onde estão definidas as seguintes regras, ou primitivas, ou comandos, como: substituir branco (*) por símbolo (I). Com estas regras cria-se o conjunto de instruções ou lista de regras. A simulação desta máquina de Turing pode ser feita em papel ou lousa, após o trabalho com autômato finito, podem ser utilizados produtos de software para construir e simular a máquina de Turing.

3.4. Máquina de estados finitos ou autômato finito

Em [Menezes 1998] um *Sistema de Estados Finitos* é exemplificado através do funcionamento de um elevador: onde cada estado possui as informações relativas ao andar corrente e direção do próximo movimento e não é capaz de memorizar as informações anteriores.

O estudo dos autômatos finitos permite projetar vários tipos comuns de computador, algoritmos e programas. O autômato finito em [Hopcroft 1979] e [Papadimitriou 1999] é especificado formalmente através do seu conjunto de estados, do conjunto dos terminais, do conjunto das produções construídas, do estado inicial do autômato e do conjunto de seus estados finais.

Como em [Menezes 1998], neste trabalho a representação da função programa será feita através de um grafo, onde os estados são representados através de circunferências e os arcos contêm as ações ou o símbolo lido para atingir o próximo estado. A figura 9 contém exemplo de autômato para simulação dos estados civis de uma pessoa.

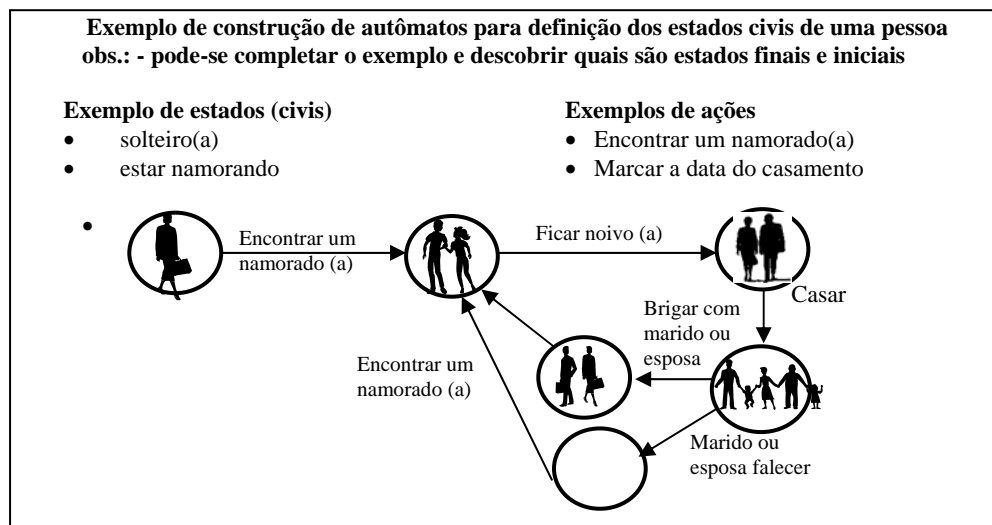


Figura 9. Exemplo de um autômato finito que simula os estados civis de uma pessoa.

4. Resultados Parciais

Foi realizado um trabalho inicial com 3 crianças com 9, 10 e 11 anos de idade. Tomando como base a primeira pesquisa realizada com crianças, seguiu-se da seguinte forma. Depois de um diálogo com perguntas sobre a vida de cada criança e suas preferências a respeito das disciplinas e o que a criança gostava de fazer quando não estava na escola. Após foi pedido que respondessem o que gostariam de aprender do Computador e de Ciência da Computação.

Para que as crianças pudessem distinguir entre esses dois conceitos, foram dados dois exemplos: “Quando a professora ensina somar, subtrair, ela ensina a mexer com a calculadora?”; “Sobre um carro, pode-se aprender a dirigi-lo, ou a construí-lo”.

Durante a conversa pôde ser percebido que as crianças ficaram impressionadas em saber que podem fazer seu próprio jogo e que podem construir um computador. Após seguiu-se apresentando os conceitos fundamentais, na mesma ordem apresentada neste trabalho. Utilizou-se além do computador, materiais simples, como caixas de papelão; garrafas de água etc. Procurou-se em todos exercícios trabalhar de forma lúdica, onde foram estimuladas a cooperação, autonomia e o trabalho prático, vivenciando situações comuns ao cotidiano destas crianças, como forma de motivar a criança em aprender.

Até este momento foram trabalhados os conceitos de lógica, álgebra e algoritmos. Estas crianças conseguiram definir o que é um algoritmo e distinguem sua representação e execução, também conseguiram construir seus próprios algoritmos de acordo com suas primitivas.

5. Conclusão

Na elaboração dos exercícios para demonstração e ensino dos conceitos fundamentais da Ciência da Computação houve preocupação em trabalhar de forma interdisciplinar, principalmente com a matemática. Assim, algoritmos são utilizados para desenvolvimento de cálculos, noções de lógica e conjuntos, história da filosofia, conjugação de verbos etc. Outra preocupação foi transformar estes conceitos em aplicações que fossem utilizados elementos mais simples possíveis, coloridos e presentes em nosso cotidiano. A transformação destes conceitos em aplicações úteis é

de fundamental importância, pois leva motivação para o estudo de Ciência da Computação.

Sem a pretensão de um curso completo, este trabalho é uma introdução para alguns temas básicos, cuidadosamente selecionados por profissionais da área de Ciência da Computação com a pretensão de formação do pensamento instigante sobre problemas computacionais. Assim, o próximo conceito a ser acrescido deve ser computabilidade, pois se refere à busca de que tipo de problemas se consegue resolver utilizando um computador.

Atualmente, este trabalho está concentrado na aplicação dos exercícios em crianças. Esta aplicação, além de testes e correções necessárias nos exercícios propostos, faz com que vislumbremos que o ensino de Ciência da Computação para crianças é viável e muito gratificante.

Referências Bibliográficas

- Bird, R. Programs and Machines. London: John-Wiley, 1976.
- Brookshear, J. G., Ciência da Computação - Uma Visão Abrangente. Porto Alegre: Bookman, 2000.
- Divério, T. A., Menezes, P. B. Teoria da Computação: Máquinas Universais e Computabilidade. Porto Alegre: Sagra Luzzato, 1999.
- Fernades, C. S., Menezes, P. B., Accorsi, F. A Propose of Teaching Computer Science for Children. In: Internacional Conference on Engineering and Computer Education – ICECE'2000, São Paulo, agosto de 2000.
- Fonseca, C. História da Computação – Teoria e Tecnologia. Brasília: LTR, 1998. Disponível em <http://www.cic.unb.br/tutores>.
- Forbellone, A. L. V., Eberspächer, H. F. Lógica de Programação – A Construção de Algoritmos e Estrutura de Dados. São Paulo: Makron Books, 1993.
- Furth, H. G. Piaget na Sala de Aula, Prentice-Hall, 1972.
- Guimarães, A. M., Lages N. A. C. Algoritmos e Estrutura de Dados. Rio de Janeiro: Livros Técnicos e Científicos, 1985.
- Hopcroft, J. E., Ullman, J. D. Introduction to Automata Theory, Languages and Computation. Addison-Wesley, 1979.
- Jostein, G. O Mundo de Sofia. Oslo: Schwarcz Ltda, 1995.
- Menezes, P. B. Linguagens Formais e Autômatos. Porto Alegre: Sagra Luzzato, 1998.
- Papadimitriou, C. H., Lewis, H. R. Elementos da Teoria da Computação. Califórnia: Bookman. 1999.
- Sagatume, M., Baum, G. Problemas, Lenguajes y Algoritmos. Campinas: Unicamp. 1986.

Bibliografia

- [AHO 74] AHO, A.; HOPCROFT, J; ULLMAN, J. **The Design and Analysis of Computer Algorithms**. USA: Addison-Wesley, 1974.
- [AMS 2000] AMSON, G. A.; JAMAL, R. M.; TEIXEIRA, J. C. **Álgebra I**. São Paulo: Anglo, 2000.
- [AND 99] ANDRADE, H.; CRESPO, S.; PORTELLA, A. Metodologia para Criação de Roteiros Educativos em Realidade Virtual. **Revista Brasileira de Informática na Educação**, Florianópolis, n.5, p. 53, set.1999.
- [BAR 2002] BARANAUSKAS, M. C. C. **Uma Abordagem Construcionista ao Design de um Ambiente para Programação em Lógica**. Disponível em: <<http://www.nied.unicamp.br/publicacoes/separatas/Sep15.pdf>>. Acesso em: 10 de mai. 2002.
- [BIR 79] BIRD, R. **Programs and Machines – an Introduction to the Theory of Computation**. USA: Addison-Wesley, 1979.
- [BOY 91] BOYER, C. B. **A History of Mathematics**. New York: John Wiley & Sons, 1991.
- [BRO 2000] BROOKSHEAR, J. G. **Ciência da Computação - Uma Visão Abrangente**. Porto Alegre: Bookman, 2000.
- [CAP 2001] COLÉGIO de Aplicação da UFRGS. Disponível em: <<http://www.cap.ufrgs.br/>>. Acesso em: 15 jan. 2001.
- [CAR 94] CARDOSO, C. M.; RAMOS, E. M. F. AALO - Um Ambiente para a Aprendizagem de Lógica. In: CONGRESSO DA SOCIEDADE BRASILEIRA DE COMPUTAÇÃO, SBC, 1994, Florianópolis, **Anais...**[S.l : s.n.],1994.
- [COR 90] CORMEN, T.; LEISERSON, R. **Introduction to Algorithms**. USA: MIT Press/McGraw-Hill, 1990.
- [CRIS 98] CRISTÓVÃO, H. M. Ambientes Computacionais para Apoio à Aprendizagem: Um experimento com frações. **Revista Brasileira de Informática na Educação**, Florianópolis, n.3, p. 69, set.1998.
- [DAV 94] DAVIS, M.D. et al. **Computability, Complexity, and Languages: fundamentals of theoretical computer science**. San Diego: Academic Press, 1994.
- [DIV 99] DIVERIO, T. A.; MENEZES, P. B. **Teoria da Computação: máquinas universais e computabilidade**. Porto Alegre: Sagra Luzzato, 1999.

- [DIV 2001] DIVERIO, T. A.; MENEZES, P. B. Teoria da Computação e os Profissionais da Área da computação. In: ESCOLA REGIONAL DE INFORMÁTICA, 9., 2001, Passo Fundo, Maringá e São José. **Livro Texto**. Porto Alegre. Instituto de Informática da UFRGS, 2001.
- [ESM 99] ESMIN, A. O que um Aluno de Ensino Médio deve Conhecer de Computação? Uma Proposta de Currículo. In: CONGRESSO NACIONAL DA SOCIEDADE BRASILEIRA DE COMPUTAÇÃO, 19., 1999, RIO DE JANEIRO. **Educação e Aprendizagem na Sociedade da Informação**: anais. Rio de Janeiro: EntreLugar, 1999.
- [FAR 2002] FARIA, C. V.; PEREIRA, J.; CAMBOIM, L. **Teorias de Aprendizagem**. Disponível em: <<http://www.ufv.br/dpe/edu660/edu660.htm>. Acesso em: 20 fev. 2002>.
- [FER 98] FERNANDES, C. S.; PERIOTTO, A. J. **Desenvolvimento e Avaliação de Produtos de Software Educacional**. 1998. Dissertação (Mestrado em Educação)-Unoeste, Presidente Prudente.
- [FER 2000] FERNANDES, C. S.; MENEZES, P. B.; ACCORSI, F. A Propose of Teaching Computer Science for Children. In: INTERNATIONAL CONFERENCE ON ENGINEERING AND COMPUTER EDUCATION, 2000. **Proceedings...** São Paulo: Senac, 2000.
- [FER 2000a] FERNANDES, C. S.; MENEZES, P. B. Computer Science for Children. In: INTERNATIONAL CONFERENCE SOCIETY FOR INFORMATION TECHNOLOGY & TEACHER EDUCATION AND ASSOCIATION FOR THE ADVANCEMENT OF COMPUTING IN EDUCATION, 2001. **Proceedings...** Flórida: SITE, 2000.
- [FER 2001] FERNANDES, C. S.; MENEZES, P. B.; ACCORSI, F. Metodologia do Ensino de Ciência da Computação: uma proposta para crianças. In: WORKSHOP DE INFORMÁTICA NA ESCOLA - CONGRESSO DA SOCIEDADE BRASILEIRA DE COMPUTAÇÃO, 2001. **Proceedings...** Fortaleza: SBC, 2001.
- [FON 98] FONSECA, C. **História da Computação – Teoria e Tecnologia**. Brasília: LTR, 1998. Disponível em: <<http://www.cic.unb.br/tutores>>. Acesso em: 02 fev. 2001.
- [FOR 93] FORBELLONE, A. L. V.; EBERSPÄCHER, H. F. **Lógica de Programação – A Construção de Algoritmos e Estrutura de Dados**. São Paulo: Makron Books, 1993.
- [FUR 72] FURTH, H. G. **Piaget na Sala de Aula**. São Paulo: Prentice-Hall, 1972.
- [GRA 94] GRAHAM, R.; KNUTH, D.; PATASHNIK, O. **Matemática Concreta. Fundamentos para a Ciência da Computação**. Rio de Janeiro: Livros Técnicos e Científicos, 1994.

- [GUI 85] GUIMARÃES, A. M.; LAGES N. A. C. **Algoritmos e Estrutura de Dados**. Rio de Janeiro: Livros Técnicos e Científicos, 1985.
- [HAR 92] HAREL, D. **Algorithmics: the spirit of computing**. USA: Addison-Wesley, 199.
- [HOP 79] HOPCROFT, J. E.; ULLMAN, J. D. **Introduction to Automata Theory, Languages and Computation**. USA: Addison-Wesley, 1979.
- [HUN 87] HUNTER, R. **Compiladores – sua concepção e programação em pascal**. Portugal: Presença, 1987.
- [INF 2000] INFRA-estrutura das escolas de Ensino Médio. **Folha de São Paulo**, São Paulo, p. 4-3, 1^o de maio de 2000.
- [JON 73] JONES, N. D. **Computability Theory: an introduction**. Pennsylvania: Academic Press, 1973.
- [JOS 95] JOSTEIN, G. **O Mundo de Sofia**. Oslo: Schwarcz, 1995.
- [JUR 97] JUREMA, A. C.; O'ROURKE, M. Critical Consciousness and Information Techology (IT) – An International Approach to Developing IT Literacy in School Twenty-two points, plus triple-word-score, plus fifty points for using all my letters.. In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, SBIE, 8., 1997. **Anais...** São José dos Campos: ITA, 1997.
- [KAI 72] KAIN, R. Y. **Automata Theory: machines and languages**. Minnesota: McGraw-Hill Book Company, 1972.
- [KOW 95] KOWALTOWSKI, T.; LUCCHESI, C. L. Applications of Finite Automata in Debugging Natural Language Vocabularies. **Journal of the Brazilian Computing Society**, [S.l.], 1995.
- [LEO 97] LEONARDO, E. **Software para Ensino Fundamental: teoria dos conjuntos**. 1997. Monografia (Bacharelado em Ciência da Computação) Fipp, Unoeste, Presidente Prudente. 1997.
- [LEW 98] LEWIS, H. R.; PAPADIMITROU, C. H. **Elements of the Theory of Computation**. New Jersey: Prentice-Hall, 1998.
- [LUC 79] LUCCHESI, C. et al. **Aspectos Teóricos da Computação**. Rio de Janeiro: Projeto Euclides, 1979.
- [MAN 88] MANBER, U. **Introduction to Algorithms: a creative approach**. USA: Addison-Wesley, 1988.
- [MEC 2002] DIRETRIZES Curriculares de Cursos da Área de Computação e Informática. Disponível em: <http://www.mec.gov.br/sesu/ftp/cur-diretriz/computacao/co_diretriz.rtf>. Acesso em: 10 jan. 2002.

- [MEN 98] MENEZES, P. B. **Linguagens Formais e Autômatos**. Porto Alegre: Sagra Luzzato, 1998.
- [MIS 2002] MISKULIN, R. G. S. **Softwares Educacionais – Ambientes Computacionais Utilizados no Ensino**. Disponível em: <<http://www.cempem.fae.unicamp.br/lapemmec/softwares.htm>>. Acesso em: 10 mai 2002.
- [MOR 98] MORET, B. M. E. **The Theory of Computation**. México: Addison-Wesley, 1998.
- [NET 87] NETO, J. J. **Introdução à Compilação**. Rio de Janeiro: Livros Técnicos e Científicos, 1987.
- [NIT 2001] NITZKE, J. A.; CAMPOS, M. B.; LIMA, M. F. **Teoria de Piaget**. Disponível em: <<http://penta.ufrgs.br/~marcia/teopiag.htm>>. Acesso em 10 dez. 2001.
- [NUN 2001] NUNES, M. G.; DOSUALDO, D. **Teoria da Computação**. Disponível em: <<http://www.icms.sc.usp.br/~mdgvnune/download/sce5832/HDOCMAINTeoria.html>>. Acesso em: 25 ago. 2001.
- [OLI 2000] OLIVEIRA, A. A. F. ENIBAM – Ensino Informaizado em Tópicos Básicos de Matemática. **Revista Brasileira de Informática na Educação**, Florianópolis, n. 6, p. 11, abr.2000.
- [PAN 97] PANQUEVA, A. H. G. Software Educativo Multimídia – Aspectos Críticos no seu Ciclo de Vida. **Revista Brasileira de Informática na Educação**, Florianópolis, n. 3, p. 09, set.1997.
- [PAL 99] PANSANATO, L. T. E.; NUNES, M. G. V. Autoria de Aplicações Hipermissão para Ensino.. **Revista Brasileira de Informática na Educação**, Florianópolis, n.5, p. 09, set.1999.
- [PEQ 2000] PEQUENO, M. C. VDL – Um Ambiente Virtual Cooperativo Voltado para Aprendizagem à Distância. **Revista Brasileira de Informática na Educação**, Florianópolis, n.6, p. 18 abr. 2000.
- [PIP 97] PIPPENGER, N. **Theories of Computability**. USA: Cambridge University, 1997.
- [RAN 2001] RANGEL, S. **Modelagem de Problemas e Otimização**. Disponível em: <<http://www.dcce.ibilce.unesp.br/~socorro/XIIIEMAT/novo/model/sld001.htm>>. Acesso em: 22 nov. 2001.
- [ROD 2000] RODRIGUES. J. A. O Papel da Pedagogia e da Andragogia no Ensino de Computação e Engenharia. In: INTERNATIONAL CONFERENCE ON ENGINEERING AND COMPUTER EDUCATION, 2000. **Proceedings...** São Paulo: Senac, 2000.

- [ROS 2000] ROSATELLI, M.C. et. al. Um Ambiente Colaborativo para o Ensino a Distância. **Revista Brasileira de Informática na Educação**, Florianópolis, n.6, p. 83, abr.2000.
- [SAG 86] SAGASTUME, M; BAUM, G. **Problemas, Lenguajes y Algoritmos**. Campinas: Ed. da UNICAMP, 1986.
- [SBC 2002] **CURRÍCULO de Referência da SBC**-Sociedade Brasileira de Computação para Cursos de Graduação em Computação e Informática. Disponível em: <<http://www.sbc.org.br/educacao>>. Acesso em: 25 jan. 2002.
- [SER 2000] SÉRATES, J. S. **Raciocínio Lógico: lógico matemático, lógico quantitativo, lógico numérico, lógico analítico, lógico crítico**. Brasília: Jofonon, 2000.
- [SID 98] SIDERICOUDES, O. Desenvolvimento de Metodologias de Ensino-Aprendizagem da Matemática em Ambientes Computacionais Baseados na Estética Logo. **Revista Brasileira de Informática na Educação**, Florianópolis, n.2, p. 97, abr.1998.
- [SIL 97] SILVEIRA, F. A.; COSTA, A. C. R. O MATHEMATICA em uma experiência com Alunos de 2º Grau. In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, SBIE, 8., 1997, São José dos Campos. **Anais...** São José dos Campos: ITA, 1997.
- [SIP 97] SIPSER, M. **Introduction to the Theory of Computation**. Boston: PWS Publishing Company, 1997.
- [SUD 98] SUDKAMP, T. **A Languages and Machines: an introduction to the theory of computer science**. USA: Addison-Wesley, 1998.
- [TER 91] TERADA, R. **Desenvolvimento de Algoritmos e Estrutura de dados**. São Paulo: Makron Books, 1991.
- [TOS 2000] TOSCANI, L. V.; NERON, A. L. **Teoria dos Grafos & Análise Combinatória**. Apostila utilizada no curso de Ciência da Computação da UFRGS. Porto Alegre, 2000.
- [TOS 2001] TOSCANI, L. V.; VELOSO, P. A. **Complexidade de Algoritmos**. Porto Alegre: Sagra Luzzato, 2001.
- [VIC 88] VICCARI, R. M. **Tutor Inteligente para a Programação em Lógica - Idealização Projecto e Desenvolvimento**. 1988. Tese (Doutorado em Ciência da Computação)-Universidade de Coimbra, Coimbra.