

UM ESTUDO SOBRE A UTILIZAÇÃO DO SOFTWARE SCRATCH COMO FERRAMENTA DE APOIO AO ENSINO DA DISCIPLINA DE LÓGICA DE PROGRAMAÇÃO

A STUDY ON THE USE OF SCRATCH SOFTWARE AS AN AID TOOL FOR TEACHING IN THE DISCIPLINE OF PROGRAMMING LOGICS

Eduardo Martins Morgado¹
João Ricardo André²

1. Bacharel em Análise de Sistemas – USC, Especialista em Sistemas de Informação para Internet – USC, Licenciatura plena em Matemática – Claretiano, Licenciatura plena em Informática – FATEC Sorocaba, Mestrando da FAAC no Programa de TV Digital da Universidade “Júlio de Mesquita Filho” - UNESP Campus Bauru / SP
Docente na Etec Astor de Mattos Carvalho
Rodovia Lourenço Lozano Km 4 – Bairro Restinga – Cabrália Paulista / SP, CEP: 17480-000
Fone: (14) 3285-1210

2. Docente na Etec Rodrigues de Abreu
Rua Virgílio Malta, 12-70 – Centro – Bauru / SP, CEP: 17015-220
Fone: (14) 3234-4252
joao.andreo@etec.sp.gov.br

Recebido em: 08/11/2012
Aceito em: 13/02/2013

MORGADO, Eduardo Martins; André, João Ricardo. *Um estudo sobre a utilização do software SCRATCH como ferramenta de apoio ao ensino da disciplina de Lógica de Programação*. Mimesis, Bauru, v. 34, n. 1, p. 45-62, 2013.

RESUMO

Aprender a programar ou dominar uma linguagem de programação não é uma tarefa fácil, e a base desse processo deve iniciar pelo estudo da Lógica de Programação, principiando-se o aprendizado, através da elaboração de pseudocódigos, algoritmos e fluxogramas, isto é, técnicas que facilitam a construção e o entendimento do que se deseja realizar, para que, posteriormente, seja desenvolvido ou elaborado um programa.

Refletindo sobre o aprendizado da Lógica de Programação, este artigo apresenta informações sobre o *software* SCRATCH em relação à lógica de programação (pseudocódigo) e, de como este fornece apoio à compreensão do discente no aprendizado desta disciplina.

Palavras-chave: Lógica, Programação, SCRATCH, Aprendizado.

ABSTRACT

Learning to program or master a programming language is not an easy task, and the base of process should be by study Logic Programming, beginning to learning through the development of pseudocodes, algorithms and flowcharts, for, techniques that facilitate the construction and understanding of what you want to accomplish, to that is subsequently developed or elaborated a program.

Reflecting on the learning of Logic Programming, this article presents information about software SCRATCH in relation to logic (pseudocodes), and how it provides support to the understanding of student learning in this course.

INTRODUÇÃO

A disciplina de lógica de programação é a base do raciocínio a ser desenvolvido no discente para que o mesmo tenha êxito e um futuro promissor na área de desenvolvimento de *software*. No entanto, é a disciplina que apresenta maior dificuldade pelos alunos da área de computação / informática, pois os discentes necessitam mudar sua forma de raciocinar e adquirir a percepção de que devem pensar como realizar uma tarefa ou ação, passo-a-passo, e que isso seja transformado em uma linguagem de programação, para que o computador possa então realizar a tarefa seguindo etapa por etapa do que foi determinado.

Outra dificuldade, também sentida pelo discente, é a forma de manipular as informações ou dados abstraídos e como isso se comporta no computador.

Os processos de aprendizagem vêm sendo aprimorados com o passar dos anos em virtude das pesquisas realizadas na área e o *software* SCRATCH é um exemplo de facilitador do aprendizado como: ALICE, ASA, LOGO, SQUEAK, VISUALG, WEBPORTUGOL, entre outros, sendo cada qual com sua área de aplicação.

Em Silva (2007), jogos interativos compostos por interface gráfica propiciam um ambiente lúdico para a compreensão e a construção dos primeiros programas de computador, ao explorarem aspectos de cognição e de colaboração. Por outro lado, o desenvolvimento de jogos pode ser utilizado como uma forma descontraída e atrativa para o ensino de lógica de programação (*apud* FERREIRA *et al*, 2010, p.982).

MORGADO, Eduardo Martins; Andréo, João Ricardo. *Um estudo sobre a utilização do software SCRATCH como ferramenta de apoio ao ensino da disciplina de Lógica de Programação*. Mimesis, Bauru, v. 34, n. 1, p. 45-62, 2013.

MORGADO, Eduardo Martins; Andrêo, João Ricardo. *Um estudo sobre a utilização do software SCRATCH como ferramenta de apoio ao ensino da disciplina de Lógica de Programação*. Mimesis, Bauru, v. 34, n. 1, p. 45-62, 2013.

O intuito das considerações descritas, não é apenas apresentar as semelhanças e as proximidades entre o raciocínio lógico das instruções utilizadas na disciplina de lógica de programação e o *software SCRATCH*, mas também de exibir as características lúdicas que possam estimular o discente a investigar cada vez mais o entendimento do raciocínio lógico, expandindo seus conhecimentos e atingindo os objetivos explorados pela disciplina de lógica de programação.

O contexto desse artigo, busca amparar os leitores de que é possível o aprendizado da disciplina de lógica de programação, usando novas técnicas ou ferramentas que estimulem o discente a um contato mais intenso com a disciplina, potencializando o aprendizado com as animações produzidas no *software*, despertando as relações lógicas e matemáticas nos alunos, seduzidos pela ação de jogar ou divertir-se aprendendo.

FUNDAMENTAÇÕES

Definições sobre pseudocódigo/ algoritmo/ fluxograma

Na construção de um programa computacional, pode-se desenvolver o raciocínio lógico do mesmo, utilizando as técnicas de pseudocódigo, algoritmo ou fluxograma – diagrama de blocos.

O pseudocódigo é escrever em linguagem nativa (português) as ações ou atos que devem ser realizados para efetuar ou executar determinada tarefa. Pode ser chamado de PORTUGUÊS ESTRUTURADO ou PORTUGOL.

No pseudocódigo, deve-se utilizar qualquer método significativo para especificar com clareza e concisão um dado algoritmo, utilizando-se a linguagem comum que é a maneira mais prática do raciocínio lógico. (CORMEN *et al.* 2002, p.11)

Pode-se definir como algoritmo computacional, uma forma estruturada de resolução de um problema, isto é, uma sequência lógica de ações a serem realizadas para a resolução de um problema, muitas vezes comparado com uma “receita de bolo”.

Entende-se como algoritmo, um procedimento para computador escrito de forma clara que recebe valor ou valores como dados de entrada e retorna ou produz um valor ou conjunto de valores

como saída, isto é, um algoritmo, são ações executadas passo a passo pelo computador que transforma o dado que entrou em informação na saída. (CORMEM *et al.* 2002, p.3)

O fluxograma por sua vez é uma representação gráfica que faz uso de símbolos pré-definidos para a descrição do fluxo ou sequência de ações e/ou informações ou processos que representam a ENTRADA, PROCESSAMENTO e a SAÍDA dos dados trabalhados. Este busca padronizar a representação dos métodos ou procedimentos, facilitando a leitura e o entendimento por parte do profissional de desenvolvimento de *software*.

Para Souza (2000), pode-se compreender fluxograma por um conjunto de estruturas de programação, com representações distintas, utilizando-se de figuras geométricas, representando a estrutura lógica sequencial, as estruturas de decisão, e as estruturas de repetição e o fluxo sequencial, representado por setas direcionais, facilitando o entendimento da solução. (*apud* DAZZI e SANTIAGO 2003, p. 114)

Exemplos de um fluxograma e um pseudocódigo nas figuras 1 e 2, que seguem.

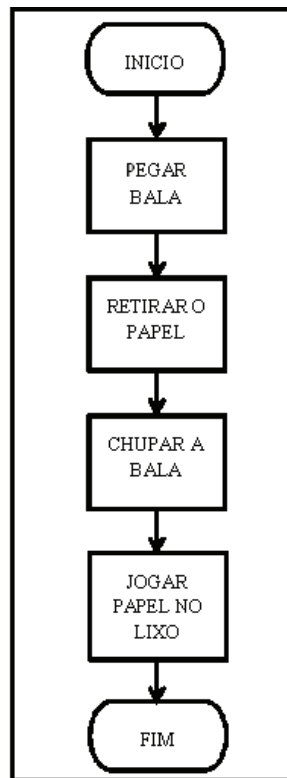


Figura 1 - Diagrama “Chupar Bala”

Fonte: Crédito do Autor João Ricardo Andréo, a partir de aulas ministradas em Lógica de Programação.

MORGADO, Eduardo Martins; Andréo, João Ricardo. *Um estudo sobre a utilização do software SCRATCH como ferramenta de apoio ao ensino da disciplina de Lógica de Programação*. Mimesis, Bauru, v. 34, n. 1, p. 45-62, 2013.

MORGADO, Eduardo Martins; Andrêo, João Ricardo. *Um estudo sobre a utilização do software SCRATCH como ferramenta de apoio ao ensino da disciplina de Lógica de Programação*. Mimesis, Bauru, v. 34, n. 1, p. 45-62, 2013.

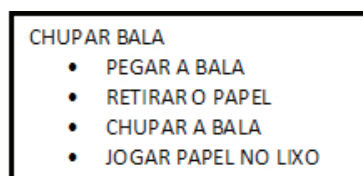


Figura 2 - Pseudocódigo “Chupar Bala”

Fonte: Crédito do Autor João Ricardo Andrêo, a partir de aulas ministradas em Lógica de Programação.

Percebe-se que tanto o pseudocódigo como o fluxograma é escrito de maneira direta e objetiva, obedecendo a uma sequência lógica de acontecimentos ou ações. Isto ocorre por que alguns métodos são obedecidos:

- Usar somente um verbo por frase,
- Usar frases curtas e simples,
- Ser objetivo,
- Não usar palavras com sentido dúbio,
- Imaginar que o algoritmo deve ser interpretado por qualquer pessoa, isto é, aquelas que não são da área de informática.

Outros exemplos, um pouco mais complexos, podem ser formulados como a soma de dois números. Neste caso os valores devem ser armazenados ou atribuídos a variáveis e também compor uma expressão matemática capaz de realizar o cálculo de forma comum – genérica ou independente dos valores recebidos / atribuídos.

A figura 3 exemplifica um algoritmo com a soma de dois números.

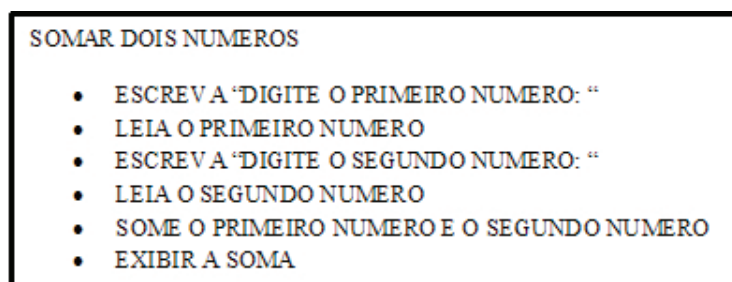


Figura 3 – Algoritmo: Somar dois Números

Fonte: Crédito do Autor João Ricardo Andrêo, a partir de aulas ministradas em Lógica de Programação.

Na área computacional, através do VISUALG, tem-se a seguinte apresentação:

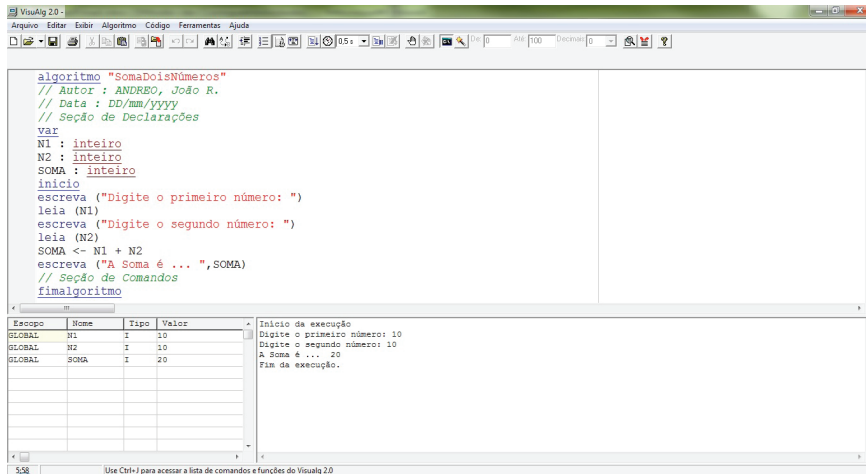


Figura 4 – Algoritmo: Somar Dois Números, em Português Estruturado no VISUALG. Fonte: Crédito do Autor João Ricardo Andréo, a partir de aulas ministradas em Lógica de Programação.

O VISUALG é um *software* ou uma ferramenta elaborada para o desenvolvimento de algoritmos, permitindo que as instruções digitadas sejam executadas e acompanhadas passo a passo, exibindo os conteúdos de variáveis e constantes na janela abaixo do código, ao lado da janela que apresenta as informações processadas.

Segue na Figura 5 cópia do trecho codificado da figura 4, para facilitar a leitura.

```
algoritmo "SomarDoisNumeros"
// Autor : ANDREO, João R.
// Data : DD/MM/YYYY
// Seção de Declarações
var
N1 : inteiro
N2 : inteiro
SOMA : inteiro
início
escreva ("Digite o primeiro número: ")
leia (N1)
escreva ("Digite o segundo número: ")
leia (N2)
SOMA <- N1 + N2
escreva ("A Soma é ... ", SOMA)
// Seção de Comandos
```

Figura 5 - Cópia da pseudoprogramação no VISUALG

MORGADO, Eduardo Martins; Andréo, João Ricardo. *Um estudo sobre a utilização do software SCRATCH como ferramenta de apoio ao ensino da disciplina de Lógica de Programação*. Mimesis, Bauru, v. 34, n. 1, p. 45-62, 2013.

MORGADO, Eduardo Martins; Andrêo, João Ricardo. *Um estudo sobre a utilização do software SCRATCH como ferramenta de apoio ao ensino da disciplina de Lógica de Programação*. Mimesis, Bauru, v. 34, n. 1, p. 45-62, 2013.

Observando a figura 5, pode-se ver a instrução VAR, que faz uma marcação de que as linhas que seguem tratam-se das variáveis utilizadas e de seu tipo ou qualificação.

As variáveis ou constantes na programação de computadores, são porções da memória que recebem uma IDENTIFICAÇÃO, isto é, um nome e também a indicação de qual tipo de dado é armazenado. Podem ser visualizados (compreendidos) como retângulos que dividem porções da memória e cada porção com um nome, uma identificação, como pode ser visto na figura 7.

A diferença entre variável e constante pode ser percebida pelo próprio nome, isto é, a variável pode ter seu conteúdo que foi armazenado, alterado durante o processamento. Já na constante não ocorre esta alteração; seu conteúdo permanece o mesmo do início ao fim da execução do programa.

Para a nomeação - identificação seguem-se algumas regras:

- O primeiro caracter deve ser sempre uma letra.
- Não usar caracteres especiais (*, @ ! # \$ % & > < ' ` { [] + - <ESPAÇO EM BRANCO>) ou palavras reservadas – aquelas que fazem parte de comando da linguagem de programação em uso
- Permitido usar o UNDERLINE
- O nome deve designar o que está armazenando

VÁLIDOS	NÃO VÁLIDOS
NOME	6CENTOS
RUA	NR*CASA
CEP	C-E-P
CODIGO	CHA VE
NR_CASA	OLA!

Figura 6 - Exemplos de IDENTIFICADORES

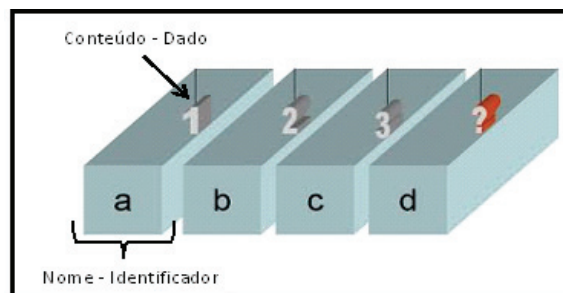


Figura 7 - Exemplificando uma Memória

Fonte: (Figura 6 e 7) Crédito do Autor João Ricardo Andrêo, a partir de aulas ministradas em Lógica de Programação.

TESTE DE MESA

O algoritmo desenvolvido necessita ser testado, para que a análise dos resultados obtidos comprove o funcionamento deste, de maneira abrangente, ou seja, os dados inseridos devem chegar ao resultado esperado atingindo desta forma o mais alto nível de abstração.

Na figura 4 do *software VISUALG*, tem-se uma amostra das variáveis e seus respectivos valores, passo a passo, na execução do programa. Isto permite que o discente visualize em tempo real o que ocorre com os valores das variáveis e conseqüentemente das expressões.

Entretanto, antes de atingir esta evolução, este processo era desenvolvido manualmente, conhecido como “teste de mesa”. Vejamos o exemplo que segue: Calcular a média de dois números.

O algoritmo pode ser escrito da seguinte forma:

```
//Algoritmo que Calcula a MEDIA de dois NUMEROS
var
P1, P2: inteiro
Md: real
Inicio
    Escreva ("Informe o Primeiro Número: ")
    Leia P1
    Escreva ("Informe o Segundo Número: ")
    Leia P2
    Md <- (P1 + P2) / 2
    Escreva ("A MEDIA dos Números é: ", Md)
Fim
```

Figura 8 - Algoritmo / Pseudocódigo - Calcular Média de Dois Números

Para realizar o teste de mesa, constrói-se uma tabela indicando o nome das variáveis no cabeçalho das colunas e os respectivos conteúdos atingidos durante o teste. Caso o resultado dos testes sejam os valores corretos do cálculo realizado manualmente, tem-se a ratificação de que o pseudocódigo está correto, conforme a tabela 1.

P1	P2	Md
4	2	3
5	5	5
2	1	1,5
2	3	2,5

Tabela 1 - TESTE DE MESA do Algoritmo / Pseudocódigo Calcular Média de Dois Números

Fonte: Crédito do Autor João Ricardo Andréo, a partir de aulas ministradas em Lógica de Programação.

MORGADO, Eduardo Martins; Andréo, João Ricardo. *Um estudo sobre a utilização do software SCRATCH como ferramenta de apoio ao ensino da disciplina de Lógica de Programação*. Mimesis, Bauru, v. 34, n. 1, p. 45-62, 2013.

MORGADO, Eduardo Martins; Andrêo, João Ricardo. *Um estudo sobre a utilização do software SCRATCH como ferramenta de apoio ao ensino da disciplina de Lógica de Programação*. Mimesis, Bauru, v. 34, n. 1, p. 45-62, 2013.

Para que a tabela 1 apresente o resultado da Média de maneira correta, os testes são efetuados passo a passo:

- Escreve-se o valor de P1 => na linha 1 coluna 1
- Escreve-se o valor de P2 => na linha 1 coluna 2
- Calcula-se a SOMA de P1 com P2 e a DIVISÃO por 2
- O Resultado FINAL será atribuído a Md, na linha 1 coluna 3

O processo é repetido para as demais linhas.

A tabela 1 apresentou o resultado do cálculo da média de maneira íntegra, isto é, o algoritmo escrito vai atingir seu objetivo para os valores desejados.

Esta é a maneira de efetuar o TESTE DE MESA manualmente.

SOFTWARES PARA O APRENDIZADO DE LÓGICA DE PROGRAMAÇÃO

Existem outros *softwares* que são utilizados não só como ferramentas para o aprendizado da lógica de programação em cursos de nível superior ou técnico, como também para o desenvolvimento do raciocínio lógico de outros discentes, por vezes até de faixa etária menor, através da ludicidade que os *softwares* proporcionam.

WEBPORTUGOL (2012) – foi desenvolvido pela Universidade do Itajaí, permitindo o desenvolvimento de um algoritmo *on-line* através da escrita estruturada, utilizando o idioma português, conforme a qualificação do referido nome. Desenvolvido no ano de 2006 por Higor Hostins em seu trabalho de conclusão de curso, o Webportugol utiliza um *applet* Java, necessitando de um *browser* para ser executado.

VISUALG (2012) – desenvolvido pela Apoio Informática Ltda – Consultoria e Desenvolvimento de Sistemas, é um software que permite o desenvolvimento de algoritmos em língua portuguesa, de maneira estruturada, e que pode ser armazenado e editado.

ASA (2012) – uma sigla que significa: Ambiente de Simulação e Animação de algoritmos. É um produto / *software* que auxilia no desenvolvimento da lógica, através da construção de fluxogramas que, posteriormente, podem ser transformados/traduzidos/convertidos em linguagens de programação como PASCAL, Linguagem C e o CLIPPER.

LOGO (2012) – é uma ferramenta para auxiliar o desenvolvimento do raciocínio lógico. Foi desenvolvido por pesquisadores do MIT – USA, sendo considerada uma linguagem mais complexa,

por possuir características que abrangem três formas ou técnicas de programação: a procedural, a objeto e a funcional.

ALICE (2012) – *software free* que pode ser utilizado em ambiente *Linux* ou *Windows*, empregado no ambiente educacional para desenvolver o raciocínio lógico. Projetado pela Universidade Carnegie Mellon nos Estados Unidos, possui ambiente gráfico em 3D, de fácil utilização e interação, trazendo semelhanças com *games*.

SCRATCH (2012) – desenvolvido pelo Instituto de Tecnologia de Massachusetts (Media Lab), de plataforma livre e que pode ser instalado em sistema operacional *Windows* ou *Linux*. Fácil e ágil para utilizar, o SCRATCH apresenta blocos de comandos que devem ser encaixados uns aos outros, formando uma sequência lógica de execução e mostrando ao usuário a maneira correta das estruturas, em analogia às mesmas existentes na lógica de programação. Permite construir jogos, histórias ou animações. No *layout* da tela do *software*, abaixo do menu ao lado esquerdo, localiza-se um quadro com grupos de opções, composto por uma variedade de comandos / ações, que permitem a elaboração de jogos ou animações, facilitando a compreensão lógica de quem o utiliza.

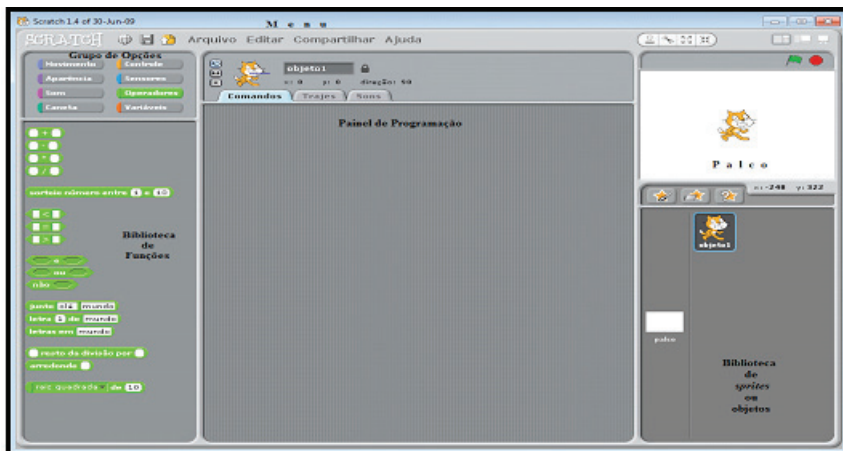


Figura 9 - Tela do software SCRATCH

Fonte: Crédito do Autor João Ricardo Andréo, a partir de anotações de aulas ministrada na disciplina “Game Design: Desenvolvimento de Jogos Digitais” no programa de mestrado em TV Digital.

DESENVOLVIMENTO

Para alcançar a compreensão deste objeto de estudo, foi realizada uma pesquisa exploratória e bibliográfica para fundamentar os

MORGADO, Eduardo Martins; Andréo, João Ricardo. *Um estudo sobre a utilização do software SCRATCH como ferramenta de apoio ao ensino da disciplina de Lógica de Programação*. Mimesis, Bauru, v. 34, n. 1, p. 45-62, 2013.

MORGADO, Eduardo Martins; Andrêo, João Ricardo. *Um estudo sobre a utilização do software SCRATCH como ferramenta de apoio ao ensino da disciplina de Lógica de Programação*. Mimesis, Bauru, v. 34, n. 1, p. 45-62, 2013.

conceitos utilizados em lógica de programação, seguindo também confrontações com a construção de exemplos no *software SCRATCH* e em pseudocódigo, evidenciando que o produto pode ser utilizado como objeto de aprendizagem para a disciplina de lógica de programação.

A figura 10 apresenta o mesmo exemplo da figura 8 (algoritmo “Somar dois Números”) aplicado ao SCRATCH.

Na figura 10, notam-se as semelhanças dos conteúdos com a figura 8, tem-se a presença das variáveis, a programação apresenta a atribuição dos conteúdos (valores), as variáveis (“mude N1 para resposta” o.u “mude N2 para resposta”), a expressão matemática (“mude MD para $(N1+N2)/2$ ”) e ao fim a apresentação do resultado.

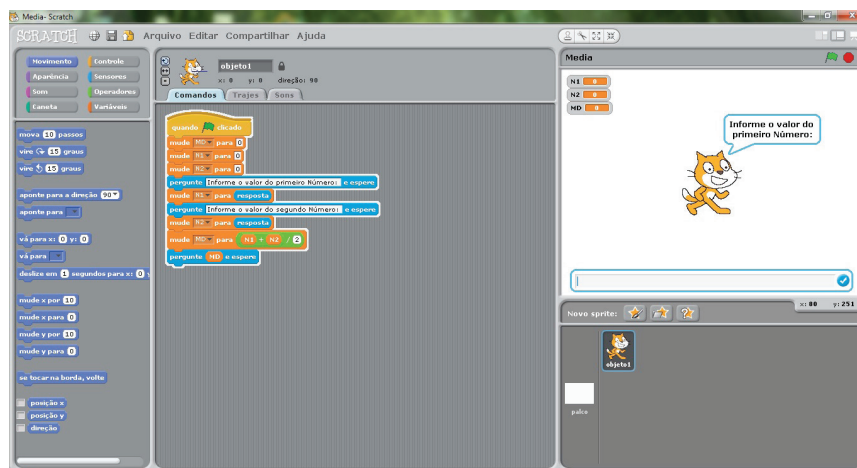


Figura 10 - Exemplo para Calcular a Média de Dois Números no SCRATCH
Fonte: Crédito do Autor João Ricardo Andrêo.

Outro exemplo que pode apontar as semelhanças está relacionado com a ESTRUTURA DE DECISÃO que faz parte do conteúdo abordado na disciplina de lógica de programação.

A estrutura de decisão é organizada através da comparação de uma ou mais condições, podendo ser constituída por uma complexa expressão e até uma simples verificação de igualdade.

Basicamente existem duas condições nesta verificação “VERDADEIRO”, no caso da expressão estar correta; ou “FALSO”, quando a resposta da expressão não é correta.

O exemplo da figura 11 apresenta novos comandos e alguma complexidade, porém, ilustra e esclarece a estrutura de decisão.

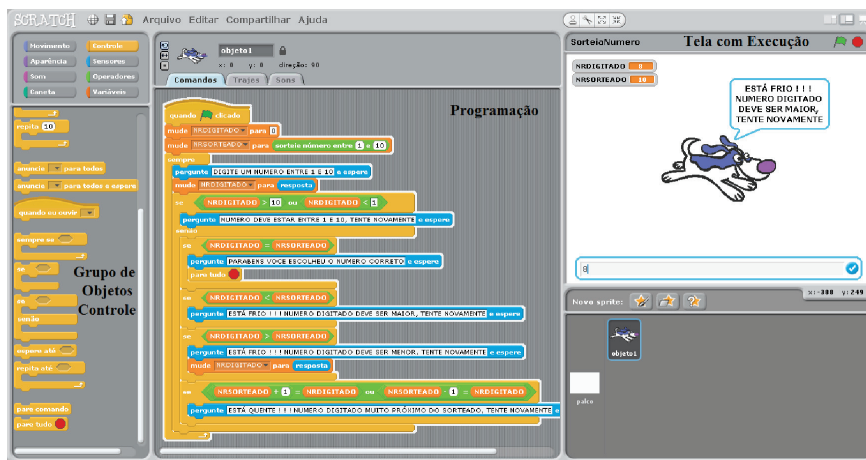


Figura 11 - Exemplo no SCRATCH sobre Estrutura de Decisão

Observando a janela de programação, pode-se notar a presença da palavra “SE” e logo à frente, uma simples comparação ou uma expressão, onde o resultado lógico sendo “VERDADEIRO”, o conteúdo que segue dentro da estrutura será executado. No caso do resultado lógico ser “FALSO”, o conteúdo dentro da estrutura não será executado, seguindo o fluxo para o próximo comando.

As figuras 11 e 12 ilustram a utilização da estrutura de decisão.

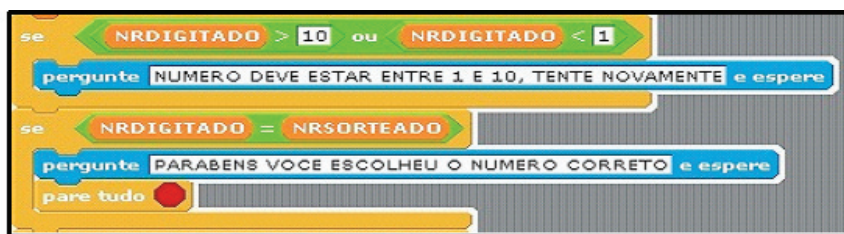


Figura 12 – Estrutura de Decisão - Comando SE no SCRATCH

Na primeira instrução de comparação, têm-se duas expressões (“Se NRDIGITADO > 10 ou NRDIGITADO < 1”). Neste caso, as duas expressões devem ser “VERDADEIRAS” para que seja executado o comando dentro da estrutura.

No segundo trecho, têm-se apenas uma comparação (“Se NRDIGITADO = NRSORTEADO”) e, se esta atingir a condição de “VERDADEIRO”, o comando dentro da estrutura (“pergunte “PARABÉNS VOCÊ ESCOLHEU O NÚMERO CORRETO””) será executado, isto é, a frase aparecerá no vídeo e logo em seguida ocorrerá a finalização da execução, uma vez que o objetivo de adivinhar um número entre 1 e 10, foi atingido pelo participante.

MORGADO, Eduardo Martins; Andréo, João Ricardo. *Um estudo sobre a utilização do software SCRATCH como ferramenta de apoio ao ensino da disciplina de Lógica de Programação*. Mimesis, Bauru, v. 34, n. 1, p. 45-62, 2013.

MORGADO, Eduardo Martins; Andrêo, João Ricardo. *Um estudo sobre a utilização do software SCRATCH como ferramenta de apoio ao ensino da disciplina de Lógica de Programação*. Mimesis, Bauru, v. 34, n. 1, p. 45-62, 2013.

A figura 13 apresenta a estrutura de decisão em pseudocódigo.

```
algoritmo "Adivinhar Número"
// Função :
// Autor : ANDRÊO, João R.
// Data : DD/MM/YYYY
// Seção de Declarações
var
  NRSORTEADO: INTEIRO
  NRDIGITADO: INTEIRO
inicio
  aleatorio 1, 10
  leia (NRSORTEADO)
  aleatorio off
  limpatela
  repita
    NRDIGITADO <- 0
    escreva("Digite um número entre 1 e 10 ")
    leia (NRDIGITADO)
    se (NRDIGITADO > 10) ou (NRDIGITADO < 1) entao
      escreva ("Número deve estar entre 1 e 10, tente novamente")
    senao
      se NRDIGITADO = NRSORTEADO entao
        escreva ("Parabéns, você escolheu o número correto")
        interrompa
      fime
      se NRDIGITADO < NRSORTEADO entao
        escreva ("Está frio, número digitado deve ser maior, tente novamente !!!")
      fime
      se NRDIGITADO > NRSORTEADO entao
        escreva ("Está quente, número digitado deve ser menor, tente novamente !!!")
      fime
      se (NRDIGITADO + 1 = NRSORTEADO) ou (NRDIGITADO - 1 = NRSORTEADO) entao
        escreva ("Está QUENTE, número digitado muito próximo do sorteado, tente novamente !!!")
      fime
    fime
  fimrepeita
// Seção de Comandos
finalgoritmo
```

Figura 13 - Pseudocódigo para Adivinhar Número Sorteado entre 1 e 10

Fazendo uma comparação entre a figura 13 e a figura 11, nota-se a semelhança das estruturas usadas no pseudocódigo e no SCRATCH, bem como a presença dos comandos que compõem a estrutura de decisão.

Outra estrutura utilizada no aprendizado da lógica de programação é a **ESTRUTURA DE REPETIÇÃO**.

Esta estrutura é utilizada para repetir um comando, ou um conjunto de comandos que se encontram dentro da estrutura. A repetição pode ser controlada por uma variável do tipo contador, controlando o valor inicial e o final, ou por uma condição / expressão.

Quando se utiliza uma variável do tipo “contador”, sabe-se qual valor inicial a variável deve assumir e qual o valor para o término da repetição, ou seja, os comandos serão executados uma quantidade finita de vezes, o que não ocorre quando se usa uma “condição”, isto é, os comandos serão executados enquanto a condição for VERDADEIRA.

A figura 14 apresenta as telas iniciais, com a entrada de dados de um programa que realiza a contagem de números de um intervalo informado pelo usuário. Este exemplo utiliza a “ESTRUTURA DE REPETIÇÃO”.

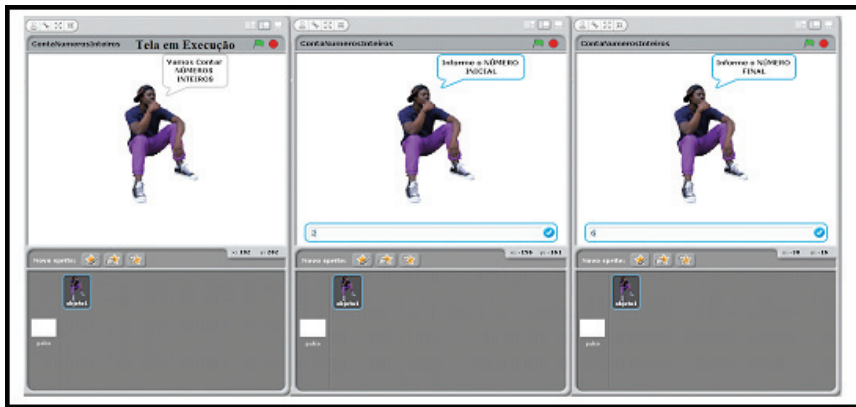


Figura 14 - Telas Iniciais no SCRATCH sobre “ESTRUTURA DE REPETIÇÃO” – Contar

Após a inserção de valores é que a lógica de repetição se faz presente.



Figura 15 - Comando de Repetição usado no SCRATCH

Comparando os comandos do SCRATCH, da figura 15, com o pseudocódigo na figura 16, têm-se as evidências de semelhanças.

```

algoritmo "Contar Números Inteiros"
// Função :
// Autor : ANDRÉO, João R.
// Data : DD/MM/YYYY
// Seção de Declarações
var
NRINICIO :INTEIRO
NRFINAL :INTEIRO
inicio
// Seção de Comandos
Limpatela

escreval ("Vamos CONTAR NÚMEROS INTEIROS")
escreval ("Informe o NÚMERO INTEIRO INICIAL: ")
leia (NRINICIO)
escreval ("Informe o NÚMERO INTEIRO FINAL: ")
leia (NRFINAL)

Enquanto NRINICIO < NRFINAL + 1 faca
    escreval (NRINICIO)
    NRINICIO <- NRINICIO + 1
Fimenquanto
Fimalgoritmo
  
```

Figura 16 - Pseudocódigo com Estrutura de Repetição - Contar Números Inteiros

MORGADO, Eduardo Martins; Andréo, João Ricardo. *Um estudo sobre a utilização do software SCRATCH como ferramenta de apoio ao ensino da disciplina de Lógica de Programação*. Mimesis, Bauru, v. 34, n. 1, p. 45-62, 2013.

MORGADO, Eduardo Martins; Andrêo, João Ricardo. *Um estudo sobre a utilização do software SCRATCH como ferramenta de apoio ao ensino da disciplina de Lógica de Programação*. Mimesis, Bauru, v. 34, n. 1, p. 45-62, 2013.

Na figura 16 têm-se a presença da estrutura de repetição iniciando no comando “Enquanto” e finalizando em “Fimenquanto”.

Quando se verifica a semelhança com a figura 15, percebe-se que o raciocínio lógico utilizado é basicamente o mesmo, apenas por uma ressalva: na estrutura da figura 16, a condição escrita na expressão é diferente da figura 15, porém, o resultado alcançado é o mesmo.

A lógica utilizada nos exemplos anteriores pode ser aprimorada, desenvolvida ou mesmo escrita de outra maneira, utilizando outras técnicas que resultarão no mesmo objetivo.

Programar ou escrever algoritmos, é uma tarefa que depende do raciocínio lógico de cada um desde que o objetivo seja atingido, porém, é desejável que o objetivo final seja alcançado com o mínimo de passos ou ações, para que a tarefa tenha êxito com a maior rapidez possível, considerando que o uso da memória e do processador (processamento) sejam desejavelmente o menor.

CONSIDERAÇÕES FINAIS

Sendo a lógica de programação, a base para o aprendizado nos cursos de ensino técnico ou superior na área de informática, no âmbito do desenvolvimento de *software*, e, de que esta disciplina é uma das principais responsáveis por reprovas ou evasões dos cursos, deve-se fazer uso de vários objetos de aprendizagem e metodologias para despertar no discente uma proximidade ou identificação com o contexto, para que este seja um profissional em potencial na área de desenvolvimento.

Dentre as dificuldades do entendimento ao raciocínio lógico, têm-se a ausência de uma sólida base de conceitos matemáticos e lógico-matemático, o que dificulta o aprendizado cognitivo do discente.

Refletindo sobre as dificuldades e avaliando a possibilidade de utilizar diversos objetos de aprendizagem e metodologias (didática), o *software* SCRATCH apresenta similaridade e proximidade em relação ao pseudocódigo, além de despertar a ludicidade e a participação do discente em aprofundar seu raciocínio lógico e cognitivo, no despertar deste para o mundo dos *games*.

Observando os grupos de objetos existentes no SCRATCH, verificou-se os mesmos operadores matemáticos e lógicos utilizados no pseudocódigo, as estruturas de controle, como a estrutura de decisão e estrutura de repetição, além das estruturas direcionadas para produção de games, como movimento, aparência, caneta, som e sensores.

A produção deste artigo demonstrou significativa relevância na utilização do *software* SCRATCH e no processo de ensino aprendizagem da disciplina de lógica de programação. Outras pesquisas podem ser realizadas cogitando o uso deste como objeto de aprendizagem para outros cursos, como princípios básicos para o desenvolvimento de games ou mesmo a utilização em outros dispositivos eletrônicos móveis (com adaptações) e até mesmo a criação de campeonatos e desafios entre escolas.

REFERÊNCIAS

CORMEN, THOMAS H. et al. **Algoritmos: teoria e prática**. Tradução de segunda edição americana – Vandenberg D. de Souza. Rio de Janeiro: Elsevier, 2002.

DAZZI, RUDIMAR L. S.; SANTIAGO, RAFAEL. **Ferramentas que auxiliam o desenvolvimento da lógica de programação**. In: XII SEMINCO, 2003, Blumenau. Anais do XII SEMINCO 2003, p. 113 – 120. Disponível em: <<http://www.inf.furb.br/seminco/2003/artigos/118-vf.pdf>>.

FERREIRA, CLÁUDIA; GONZAGA, FLÁVIO; SANTOS, RODRIGO. **Um Estudo sobre a Aprendizagem de Lógica de Programação Utilizando Programação por Demonstração**. In: XXX Congresso da Sociedade Brasileira de Computação – XVIII Workshop sobre Educação em Computação (WEI), 2010, Belo Horizonte. Anais do CSBC 2010, p. 981 - 990. Disponível em: <http://www.bcc.unifal-mg.edu.br/lares/files/artigos/Um_estudo_sobre_a_aprendizagem_de_logica_de_programacao_utilizando_pbd.pdf>.

WEBPORTUGOL. Ferramenta para desenvolver algoritmos e aprender lógica de programação. Disponível em: <<http://www.univali.br/webportugol>>.

ASA. Ferramenta para desenvolver fluxogramas e aprender lógica de programação. Disponível em: <<http://www.bracad.com.br/downloads/desenvolvimento-de-software/algoritmo>>.

MORGADO, Eduardo Martins; Andréo, João Ricardo. *Um estudo sobre a utilização do software SCRATCH como ferramenta de apoio ao ensino da disciplina de Lógica de Programação*. Mimesis, Bauru, v. 34, n. 1, p. 45-62, 2013.

MORGADO, Eduardo Martins; Andréo, João Ricardo. *Um estudo sobre a utilização do software SCRATCH como ferramenta de apoio ao ensino da disciplina de Lógica de Programação*. Mimesis, Bauru, v. 34, n. 1, p. 45-62, 2013.

ALICE. Ferramenta para desenvolver fluxogramas e aprender lógica de programação. Disponível em: <<http://www.bracad.com.br/downloads/desenvolvimento-de-software/algorithmo>>.

LOGO. Ferramenta para desenvolver fluxogramas e aprender lógica de programação Disponível em: <<http://pt.scribd.com/doc/97964444/Scratch>>.

SCRATCH. Ferramenta para desenvolver fluxogramas e aprender lógica de programação Disponível em: <<http://pt.scribd.com/doc/97964444/Scratch>>.

