



ANÁLISE DAS FERRAMENTAS SCRATCH E LOGO NO AUXÍLIO DO ENSINO DA LÓGICA COMPUTACIONAL PARA ENGENHARIA

Stéphanie A. Braga – stephanie.abraga@gmail.com

Jorge F. M. C. da Silva – jf.engtelecom@gmail.com

Guilherme C. Jales – guilhermejales@gmail.com

Francisco D. C. Rebouças – davi.reboucas91@gmail.com

Antônio H. S. Lira – lira.hugo92@gmail.com

Caio R. F. Barbosa – caio.raveli@gmail.com

José W. M. Menezes – wally@ifce.edu.br

Carlos M. J. M. D. Júnior – cmauriciojd@gmail.com

Instituição Federal de Educação, Ciência e Tecnologia do Ceará

Avenida 13 de maio, 2081, Benfica.

60040-531 – Fortaleza - Ceará

Resumo: *O profissional de engenharia está diariamente enfrentando problemas que requerem diversificados métodos de soluções que têm como base a lógica computacional, vê-se então a importância das disciplinas que abrangem esse conceito. Nesse artigo serão apresentadas duas ferramentas que podem auxiliar no aprendizado da lógica computacional para alunos que estão iniciando nos cursos de engenharia, o Scratch e o Logo. São ferramentas distintas, mas que possuem vantagens similares que podem ajudar ao estudante na adaptação com o conceito da lógica, contribuindo, posteriormente, para o aprendizado de outras linguagens de programação.*

Palavras-chave: *Lógica computacional, Scratch, Logo, Linguagem de programação.*

1. INTRODUÇÃO

A lógica está presente em quase tudo que realizamos e a sua importância pode ser percebida quando temos que ordenar os pensamentos para agir corretamente. É uma forma de organizar os passos que levam a resolução de um objetivo determinado, tornando-a mais simples.

As disciplinas de Lógica Computacional e Algoritmos estão presentes na grade curricular da grande maioria dos cursos de Engenharia e outros da área da ciência, de forma que os estudantes que ingressam nesses determinados cursos precisam se submeter a esse tipo de disciplina.

O objetivo desta disciplina é fazer o aluno compreender a forma de “raciocínio” do computador e dominar as técnicas para solucionar problemas propostos. Estas soluções podem ser um programa de computador ou só um pseudocódigo – o Algoritmo. Um



algoritmo é um conjunto finito de comandos que resolverão um problema proposto (FARRER, 1989).

O que se torna um agravante é que a maioria dos estudantes apresenta muitas dificuldades para desenvolver o raciocínio lógico, não se adaptando bem a esse tipo de disciplina, por diversos motivos, fato que pode até ser considerada uma das causas para o problema de evasão nos cursos de Engenharia das instituições de ensino superior brasileiras (CABRAL, 2007).

Nesse artigo serão apresentadas duas ferramentas que podem ser utilizadas para auxiliar no ensino da Lógica Computacional, o “Scratch” e o Logo, sendo mostrados comparativos que ressaltam as vantagens de cada uma.

2. FUNDAMENTAÇÃO TEÓRICA

O início da ciência da Lógica encontra-se na antiga Grécia [Kne68] [Boc66]. As polêmicas geradas pela teoria de Parmênides e os famosos argumentos de Zenão, que negavam a realidade do movimento fazendo um uso indevido do princípio da não contradição, contribuíram para a distinção dos conceitos, para se ver a necessidade de argumentar com clareza, mediante demonstrações rigorosas, e assim responder às objeções dos adversários. Mais tarde, as sutilezas dos sofistas, que reduziam todo o saber à arte de convencer pelas palavras, levaram Sócrates a defender o valor dos conceitos e tentar defini-los com precisão. Assim a Lógica como ciência vai se formando pouco a pouco, principalmente com Sócrates e Platão. (FONSECA, 2007).*

Diariamente, os profissionais da engenharia enfrentam vários tipos de problemas, que exigem diversas habilidades além da disciplina e desenvoltura, como também o uso de métodos específicos de acordo com o nível de complexidade de cada problema. Com o avanço da área da eletrônica e Engenharia de Software, os programas voltados para a área da engenharia foram também crescendo, resultando no imenso mercado de softwares que temos hoje, que requerem dos profissionais pelo menos o uso da lógica ou outros conhecimentos na área de programação (CAETANO, 2011).

Vê-se então a importância da devida compreensão e adequação às disciplinas de Lógica Computacional ofertadas, geralmente no início dos cursos de engenharia. O objetivo geral da disciplina é permitir que o aluno desenvolva o raciocínio lógico aplicado a solução de problemas em nível computacional, além de introduzir os conceitos básicos de desenvolvimento de algoritmos, de forma a propiciar aos alunos uma visão crítica e sistemática sobre resolução de problemas e prepará-los para a atividade de programação (CAETANO, 2011).

Entretanto, diversos fatores contribuem para a má adaptação dos alunos nessas disciplinas: a dificuldade de adaptação dos alunos em desenvolver raciocínio lógico, a falta de motivação gerada pelo despreparo, a ausência de uma boa base lógico-matemática e os métodos didáticos de alguns professores (PEREIRA, 2012).

Faz-se necessário a busca de métodos que melhorem o desempenho dos alunos nesses tipos de disciplinas, pois além de contribuir com a resolução do problema de reprovações e desistências nas universidades, fará com que o profissional da área tenha um bom desempenho em outras disciplinas que venham a usar a lógica computacional como base, além de auxiliá-lo na carreira futura.

3. SCRATCH E LOGO



3.1. O Logo

A Linguagem Logo foi desenvolvida em 1968 pelo sul-africano Seymour Papert e é uma linguagem de programação que tem como característica a possibilidade do usuário dar instruções ao computador para que ele execute as ações determinadas por ele. O objetivo de Papert ao criar a Logo é oportunizar aos usuários a aprender com prazer a programar e assim potencializar a aprendizagem (POCRIFKA, 2009).

A Linguagem Logo se constitui em um programa computacional aberto, de autoria e programação, onde o usuário pode apresentar os conhecimentos adquiridos no estudo de um determinado conteúdo, tornando-se o autor do trabalho realizado.

Uma característica importante e que faz da Logo uma ferramenta voltada para o ensino não só na Engenharia, com também em outros cursos é ter como concepção teórica a abordagem Construcionista de aprendizagem. Segundo os estudos de Seymour Papert, esta teoria tem por objetivo promover a construção do conhecimento por meio do uso das tecnologias.

A linguagem Logo é constituída de uma parte gráfica e outra de texto. No ambiente gráfico o usuário pode desenhar na tela do computador, movimentando um cursor em forma de tartaruga, que é o símbolo da linguagem Logo, e executa as instruções dadas pelo usuário, por meio dos comandos.

3.2. O Scratch

O Scratch é uma linguagem gráfica de programação criada no Media Lab do Instituto de Tecnologia de Massachusetts, inspirada nas linguagens Logo e Squeak, mas que pretende ser mais simples, fácil de utilizar e mais intuitiva. Ela foi partilhada com o mundo em 15 de maio de 2007. É um software gratuito, possui uma IDE onde não é preciso digitar funções, endereços, etc. Seu objetivo primário é facilitar a introdução de conceitos de matemática e de computação, enquanto também induzindo o pensamento criativo, o raciocínio sistemático e o trabalho colaborativo (SCRATCH, 2013).

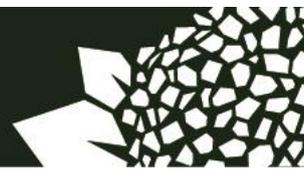
Com o Scratch é possível misturar diferentes recursos (gráficos, fotos, músicas, sons) de uma forma bastante criativa, daí provém a origem se deu nome. O termo Scratch provém da técnica de scratching utilizada pelos Disco-Jockeys do Hip-Hop que giram os discos de vinil com as mãos para frente e para trás, fazendo misturas musicais criativas.

O Scratch traz os conceitos de informática e lógica de uma forma diferenciada para seus usuários, pois a programação do Scratch é feita por meio de blocos de comandos que são encaixados uns aos outros, formando uma sequência desejada. Esses blocos somente se encaixam de uma única forma, não ocorrendo, assim, erros de sintaxe.

4. METODOLOGIA

Serão mostrados como exemplo dois algoritmos feitos tanto no Scratch como na linguagem Logo. Os ambientes de desenvolvimento utilizados para a execução dos algoritmos foram a IDE (*Integrated Development Environment*) do Scratch, disponibilizada no próprio site, e o ambiente Xlogo.

4.1. Média Final



Esse é um exemplo típico de algoritmo para quem está iniciando com a Lógica de programação, pois utiliza os conceitos de média aritmética e ponderada. Utilizou-se o mesmo método usado pelo IFCE - Instituto Federal de Educação, Ciência e Tecnologia do Ceará para calcular a média final dos seus alunos. A média é dividida em duas etapas, sendo que cada etapa é a média aritmética de duas provas. A média final é a média ponderada das etapas 1 e 2, com os pesos 2 e 3, respectivamente.

Dados de entrada: $N1$, $N2$, $N3$ e $N4$.

Variáveis Auxiliares: $Etapa1$, $Etapa2$.

Saída: $Média$. Se $Média \geq 7.0$, mostra a mensagem “aprovado”, se $Média < 7.0$, mostra “reprovado”.

Procedimentos:

$$Etapa1 = \left(\frac{N1 + N2}{2} \right) (1)$$

$$Etapa2 = \left(\frac{N3 + N4}{2} \right) (2)$$

$$Média = \frac{(Etapa1).2 + (Etapa2).3}{5} (3)$$

Após definidos os procedimentos acima citados, iniciou-se o processo de execução dos algoritmos. A Figura 1 ilustra a etapa inicial do algoritmo usando o Scratch. Foram criadas as variáveis, citadas acima, e foram-se encaixando os blocos de acordo com a sequência. Os blocos de cor roxa integram a seção *Aparência* no Scratch e foram usados para mostrar as mensagens ilustradas, requerendo do usuário o valor das notas. Os blocos de cor amarela são da seção *Controle* e executam uma instrução quando ocorre algo. No bloco de controle usado foi encaixado um outro bloco de cor azul, da seção *Sensores*, e a sequência só prossegue se a instrução que estiver contida nele for executada.



Figura 1 - Parte inicial do algoritmo média no Scratch.

A Figura 2 ilustra a etapa do algoritmo que calcula a média final. Os blocos de cor laranja correspondem às *Variáveis* e os de cor verde aos *Números*, que executam a parte matemática do algoritmo.



Figura 2 - Segunda etapa do algoritmo média no Scratch.

Na Figura 3 está ilustrado o código feito na linguagem Logo, no editor do ambiente Xlogo. Na linguagem Logo existem palavras definidas que realizam determinadas ações, sendo interpretadas pela IDE, porém é possível definir uma sequência de passos, como foi feito com o algoritmo “media”. No editor da IDE inicia-se com a palavra “aprenda” seguido do nome do algoritmo desejado e os dados de entrada, conforme mostrado na Figura 3, definindo assim um novo comando.

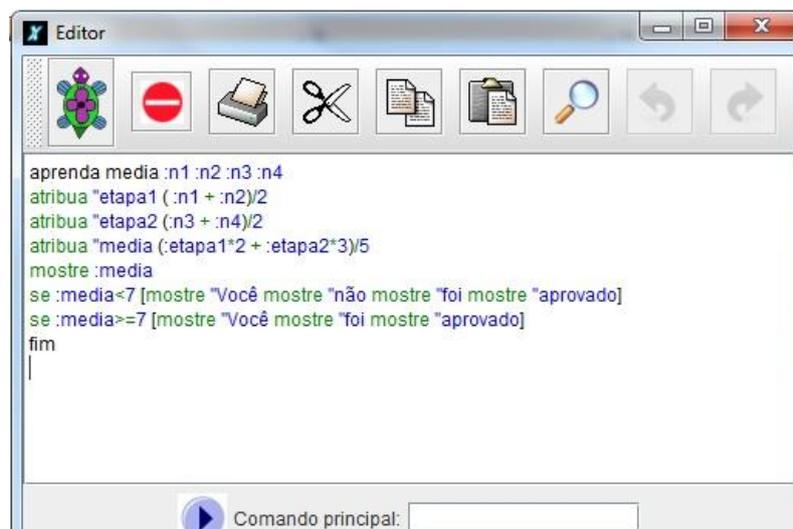


Figura 3 - Algoritmo “media” definido no editor do Xlogo.

Após definido no editor, o nome do algoritmo pode ser digitado como um comando, seguido dos dados de entrada definidos e a resposta será mostrada na tela, conforme indica a Figura 4.

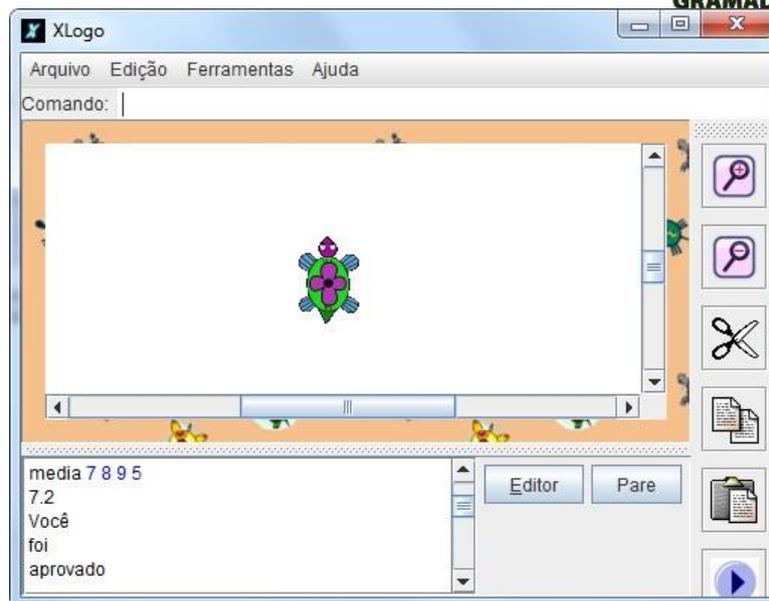


Figura 4 - Resposta do algoritmo “médica” no ambiente Xlogo.

4.2. Sequência de Fibonacci

Nesse exemplo é apresentada a conhecida sequência de Fibonacci, onde o número subsequente é a soma dos dois anteriores. Ex. 1, 2, 3, 5, 8, 13, 21, 34... e assim infinitamente. O objetivo do algoritmo é calcular qual será o valor da sequência quando calculada *lim* vezes, sendo que o valor para *lim* será informado pelo usuário. Nesse algoritmo foram introduzidas as idéias de repetição e recursividade.

Dado de entrada: *lim*, variável cujo valor será informado pelo usuário e indicará quantas vezes será executada a soma da sequência de fibonacci.

Variáveis Auxiliares: *a* e *b*. Essas variáveis serão inicializadas com os valores 0 e 1, respectivamente, porém seus valores serão mudados ao longo do algoritmo.

Saída: *res*, variável que indicará o último número resultante da sequência. O seu valor também será mudado no decorrer do algoritmo e será impresso a cada instrução de soma realizada, formando assim a sequência completa.

Procedimentos: Após serem inicializadas as variáveis citadas a cima, inicia-se o processo de repetição quantas vezes for o valor atribuído para a variável *lim*. Dessa forma, as instruções descritas abaixo serão repetidas *lim* vezes.

$$\begin{aligned} res &= a + b \\ a &= b \\ b &= res \\ \text{mostre:} &res \end{aligned}$$

A Figura 5 ilustra como foi executado o algoritmo no Scratch. Os blocos utilizados foram do mesmo tipo dos que foram usados no algoritmo da média, diferindo somente com a adição do bloco *repita*, da seção Controle. Dentro desse bloco há um espaço para ser inserido outro bloco ou valor; no caso foi inserido a variável *lim*. A variável de saída, *res*, é mostrada na tela a cada repetição.



Figura 5 - Algoritmo da sequência de fibonacci executado no Scratch.

Para executar o algoritmo na linguagem Logo, foi feito o mesmo procedimento que o algoritmo da média, definindo “fibonacci” no editor do Xlogo. Como no Scratch, a única alteração em relação aos tipos de comando do algoritmo da média foi somente a adição do comando *repita*. O algoritmo pronto no editor do Xlogo está ilustrado na Figura 6. Após executado, mostra a sequência quando $lim=6$, na Figura 7.



Figura 6 - Algoritmo da sequência de Fibonacci no editor do Xlogo.

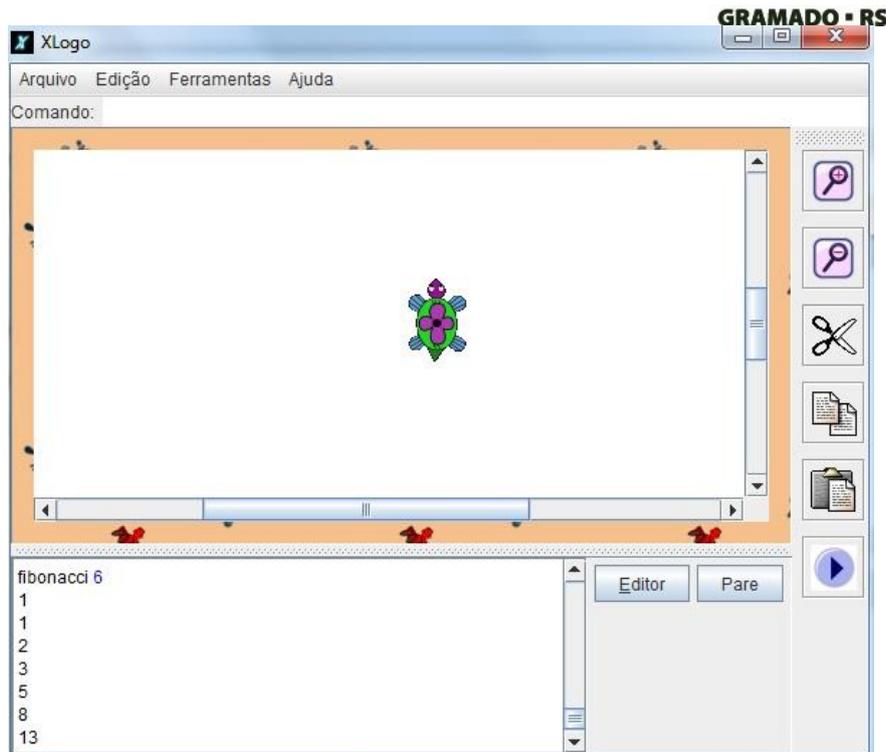


Figura 7 - Resposta à execução do comando “fibonacci” no ambiente XLogo.

5. ANÁLISES DE RESULTADOS

Com a execução dos algoritmos nas duas ferramentas, foi possível observar os recursos das mesmas que podem auxiliar no aprendizado da lógica computacional. Os resultados analisados estão expressos por meio das tabelas 1 e 2.

Tabela 1 – Análise da execução dos algoritmos com o Scratch.

Scratch	
Recurso	Benefícios
Ambiente de desenvolvimento interativo.	Instiga o aprendizado.
Execução dos algoritmos por meio de Blocos.	Facilita o primeiro contato com a Lógica.
Sequências lógicas nomeadas nos blocos.	Leva o estudante a conhecer e se adaptar às sequências lógicas.
Blocos com formatos diferenciados.	Exercita o raciocínio na forma sequencial, já que os blocos só se encaixarão se forem colocados em um local apropriado, com uma sequência correta.



Tabela 2 - Análise da execução dos algoritmos com o Logo/Xlogo.

Logo	
Recurso	Benefícios
Por ser uma linguagem, já possui palavras/comandos primitivos.	Faz com que o iniciante se adapte, de maneira simples, ao uso das palavras primitivas, facilitando posteriormente o aprendizado de outras linguagens.
Comandos nomeados de forma simples, semelhante às palavras usadas no cotidiano.	Facilita o entendimento das instruções, quando associadas ao próprio nome da ação.
Interatividade.	Estimula, com os erros, o aprendizado e a exploração de novos métodos, já que quando algum comando é acionado incorretamente, uma resposta imediata é mostrada, informando que houve um problema.
Ampla capacidade, permitindo executar algoritmos de alta complexidade.	Os iniciantes podem expandir os seus conhecimentos à medida que evoluem no aprendizado.

6. CONSIDERAÇÕES FINAIS

O aprendizado em Lógica computacional é fundamental para a formação do profissional de Engenharia. A maior utilização de ferramentas interativas em sala de aula ou laboratório pode ser um catalisador para ampliar os conhecimentos do estudante e servir de estímulo para este se aprofundar no estudo de Programação e áreas afins.

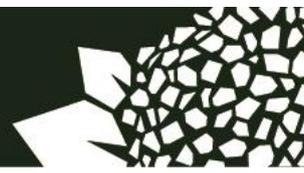
Após as análises feitas ao longo desse artigo, pode-se concluir que as duas ferramentas apresentadas, Scratch e Logo, são bastante eficientes quando se trata do aprendizado da Lógica computacional para os estudantes dos primeiros períodos de Engenharia. As vantagens citadas de cada ferramenta contribuem para o processo de amadurecimento do conceito de lógica e propiciam aos iniciantes um contato mais simples e dinâmico com a programação, servindo como base para o aprendizado de linguagens de programação mais complexas, utilizadas largamente no cotidiano dos engenheiros.

REFERÊNCIAS BIBLIOGRÁFICAS

CABRAL, M. I. C. *et al.* Perfil dos cursos de computação e informática no Brasil. Anais: XXVII – Congresso da SBC - XV WEI. Rio de Janeiro: IME, 2007.

CAETANO, Daniel. Lógica de Programação Para Engenharia, Unidade 1: Introdução à Disciplina. São Paulo: 2011.

FARRER, H. et al. Algoritmos Estruturados, 3.ed. Rio de Janeiro : LTC, 1989. 284 p, il.



FONSECA FILHO, Cléuzio. História da Computação: O caminho do pensamento e da tecnologia, 1. ed. Porto Alegre: EDIPUCRS, 2007. 205 p, il.

FORBELLONE, André Luiz Villar; EBERSPÄCHER, Henri Frederico. Lógica de programação: A construção de algoritmos e estruturas de dados. 3. ed. São Paulo: Pearson Prentice Hall, 2005. 220 p, il.

GUEDES, R. B. M.; UNIVERSIDADE FEDERAL DE PERNAMBUCO, Centro de Ciências Exatas e da Natureza. Inteligência Computacional: Métodos Procedimentais para Pensar, Aprender e Resolver Problemas, 1998. 202p, il. Dissertação (Mestrado).

PEREIRA, P. S. et al. Análise do Scratch como ferramenta de auxílio ao ensino de programação de computadores. Anais: XL – Congresso Brasileiro de Educação em Engenharia. Belém: UFPA, 2012.

POCRIFKA, D. H. Linguagem Logo e a construção do conhecimento. Anais: IX – Congresso Nacional de Educação. Curitiba: PUCPR, 2009.

SCRATCH. **ABOUT Scratch (Scratch Documentation Site)**. Disponível em: < <http://scratch.mit.edu/about/> >. Acesso em: 10 junho 2013.

ANALYSIS OF SCRATCH AND LOGO TOOLS ON SUPPORTING COMPUTATIONAL LOGIC TEACHING FOR ENGINEERING

***Abstract:** Professional engineer deals on a daily basis with problems that require diversified methods of solutions based on computational logic. In result, it is clear the worth of courses including this content. In this paper it will be presented two tools that can help on computational logic learning for students who are beginning engineering courses: Scratch and Logo. These are distinct tools, but they have similar advantages that can support student's adaptation with the concept of logic, contributing for a forward learning of other programming languages.*

***Key-words:** Computational logic, Scratch, Logo, programming language.*