

METODOLOGIA DE ENSINO DE ALGORITMOS E PROGRAMAÇÃO NA UNIVERSIDADE FEDERAL DO SUL DA BAHIA: uma análise comparativa em diferentes ambientes

João Rodrigo Pereira de Almeida¹
Diogo Rodrigues Sales da Silva²
Kennedy Morais Fernandes³

RESUMO

Este trabalho tem como objetivo apresentar a metodologia de ensino adotada pela Universidade Federal do Sul da Bahia (UFSB) nos cursos de algoritmos e programação de computadores. Alguns trabalhos na literatura mostram que as disciplinas que introduzem o ensino de programação são ofertadas logo no início da graduação, fazendo com que o estudante tenha muitas dificuldades de assimilação do conteúdo, tornando este tipo de curso como o de maior índice de evasão e reprovação no país. A metodologia adotada pela UFSB baseia-se na utilização de três ambientes de programação: *Scratch*, *Visualg* (algoritmo estruturado em Portugol) e *DevC++* (Linguagem C). O primeiro contato dos estudantes é com o raciocínio computacional, com o foco nos seguintes processos (comandos): entrada e saída de dados, estruturas condicionais e de repetição. Em cada ambiente de programação, a mudança acontece apenas na sintaxe dos comandos. Os processos a partir desses ambientes são descritos ao longo do trabalho, detalhando os mesmos e comparando os ambientes entre si, destacando suas principais peculiaridades e como estes processos funcionam de modo gradativo na aprendizagem para garantir bons resultados. Uma situação-problema e sua implementação é apresentada nos três ambientes de programação aqui abordados. Essa metodologia se mostrou mais eficaz para o ensino de algoritmo e programação do que as metodologias já utilizadas em grande parte dos cursos de ciências e tecnologias das universidades do país.

Palavras-chave: Algoritmos; Scratch; Visualg; Linguagem C.

INTRODUÇÃO

Os cursos das áreas de computação e informática são estigmatizados como cursos de elevado grau de dificuldade, estudos revelam que estão entre as graduações com maior índice de evasão e reprovação no país (DIAS, SERRÃO, 2014).

A Universidade Federal do Sul da Bahia (UFSB) utilizou junto aos seus estudantes,

¹ Discente do curso Licenciatura Interdisciplinar em Matemática, Computação e suas Tecnologias - Universidade Federal do Sul da Bahia. E-mail: jrodrigopa@gmail.com

² Discente do curso Licenciatura Interdisciplinar em Matemática, Computação e suas Tecnologias - Universidade Federal do Sul da Bahia. E-mail: diogorodrigues.sales@gmail.com

³ Professor Doutor em Modelagem Computacional –Universidade Federal do Sul da Bahia. E-mail: kennedy.fernandes@gmail.com

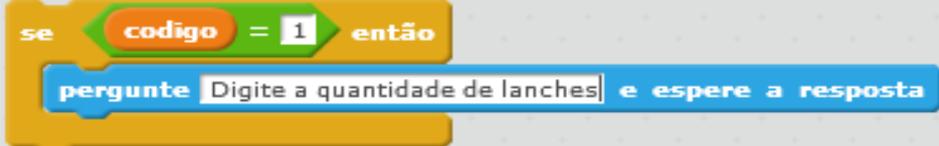
ferramentas que apresentassem ambientes de programação dinâmicos e acessíveis. Uma ferramenta conhecida como *Scratch*^[1], ambiente de programação desenvolvido pelo *Massachusetts Institute of Technology* (MIT), tem como principal objetivo a introdução da programação de computadores para usuários que tenham pouca ou nenhuma experiência. Outra ferramenta utilizada pela UFSB como forma de introdução ao aprendizado de programação é o *VisuAlg*^[2], seu principal objetivo é dispor aos usuários um ambiente de programação facilitado que utiliza o idioma português do Brasil, o qual foi batizado por “Portugol” (ALENCAR, et al, 2013).

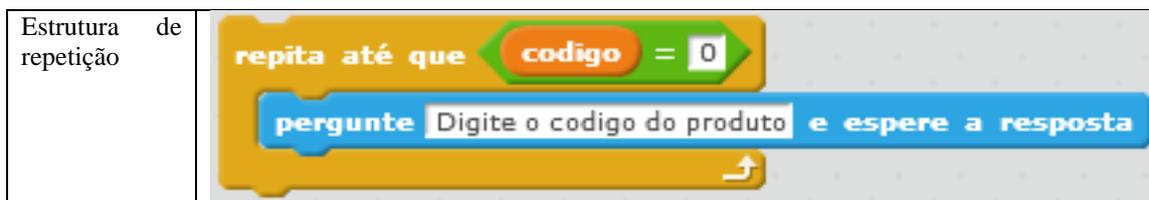
O objetivo deste trabalho é apresentar a metodologia utilizada pela UFSB, mostrando uma forma mais eficaz de aprendizagem para o ensino de linguagem de programação do que as metodologias habitualmente utilizadas nos tradicionais cursos de computação das universidades do país. Os processos (comandos) realizados através dos ambientes de programação citados neste trabalho são descritos com o funcionamento detalhado dos mesmos e comparando os ambientes entre si, destacando suas principais peculiaridades e como estes processos funcionam de modo gradativo na aprendizagem para garantir bons resultados.

METODOLOGIA

A metodologia adotada pela UFSB baseia-se na utilização de três ambientes de programação: *Scratch*^[1], *Visualg*^[2] (Algoritmo em *Portugol*) e *DevC++*^[3] (Linguagem C). O primeiro contato dos estudantes é com o raciocínio computacional, com o foco nos seguintes processos: entrada e saída de dados, estruturas condicionais e de repetição. As Tabelas 1 e 2 apresentam e descrevem a função de cada processo utilizado neste trabalho.

Tabela 1: Processos (comandos) em *Scratch*

PROCESSO	SCRATCH
Entrada	
Saída	
Estrutura Condicional	



O Ambiente *Scratch* utiliza as noções básicas de programação de forma bastante simples, através de blocos ou “peças de quebra-cabeça” (Tabela 1), bastando apenas mover os mesmos e os encaixar a outros para formar a estrutura do código. Após a montagem basta clicar na bandeira verde para executar o código montado. O ambiente *VisuAlg* é utilizado como auxílio na fixação dos conceitos de algoritmo e de lógica de programação, pois os comandos estão em português, não distingue letras maiúscula de minúscula, não há necessidades de inserção de caracteres especiais no final das linhas, como “;”.

A linguagem C, demanda de um rigor maior para se programar, exigindo um pouco mais de atenção na escrita do código. Todas as palavras-chave (comandos) em linguagem C são minúsculas. Além disso, maiúsculas e minúsculas são diferentes: `else` é uma palavra-chave, mas `ELSE` não (SCHILDT, 1996). A Tabela 2 detalha os processos em português e também em linguagem C, apresentando os comandos equivalentes ao *Scratch* (Tabela 1)

Tabela 2: Processos (comandos) em algoritmo português e linguagem C

PROCESSO	VISUALG	LINGUAGEM C
Entrada	<code>leia(quantidade)</code>	<code>scanf("%d", &quantidade);</code>
Saída	<code>escreval("Sua conta é: R\$ ")</code>	<code>printf("Sua conta é: R\$ ");</code>
Estrutura Condicional	<code>se(codigo =1) entao escreval("Digite a quantidade de lanches") fimse</code>	<code>if(codigo==1){ printf("Digite a quantidade de lanches: "); }</code>
Estrutura de repetição	<code>enquanto codigo <> 0 faca escreval("Digite o código do produto") leia(codigo) fimenquanto</code>	<code>while(codigo!=0){ printf("Digite o código produto: "); scanf("%d", &codigo); }</code>

RESULTADOS E DISCUSSÕES

Apresentamos uma situação-problema e sua implementação nos três ambientes de programação abordados.

Problema: Consiste na elaboração de um algoritmo que simula um “Sistema de Lanchonete”. O código de cada lanche é o parâmetro de entrada do sistema. Dados com códigos e valores: Código 1 – Item: *Hamburguer* – Valor: 3,50; Código 2 – Item: *X-Burguer* – Valor: 4,10. Quando o código 0 for inserido, a conta será encerrada e o seu valor será impresso na tela.

A Figura 1 apresenta a resolução do problema descrito em *Scratch*. O programa se inicia com o comando quando clicar em bandeira verde, no segundo comando temos a variável codigo iniciando com um valor diferente de 0, neste problema inserimos o valor 1000 para código.

Em seguida temos um bloco com uma estrutura de repetição (*loop*) utilizando o comando repita até que código = 0, ou seja, o programa será executado até que o usuário encerre a conta, inserindo o valor 0 para código. Dentro do *loop* temos duas estruturas condicionais (se... então), uma para quando o usuário digitar código igual a 1 e uma para quando o usuário digitar código igual a 2. O valor da conta é calculado dentro de cada estrutura condicional, de acordo com a quantidade de itens que o usuário inseriu e o valor de cada lanche. O *loop* é interrompido quando o valor 0 for inserido para código. A parte final do programa apresenta o comando de saída de dados diga, onde 2 mensagens são impressas na tela: total da conta (valor da variável conta) e total da conta com a taxa de serviço do garçom (valor da operação conta*1,1).

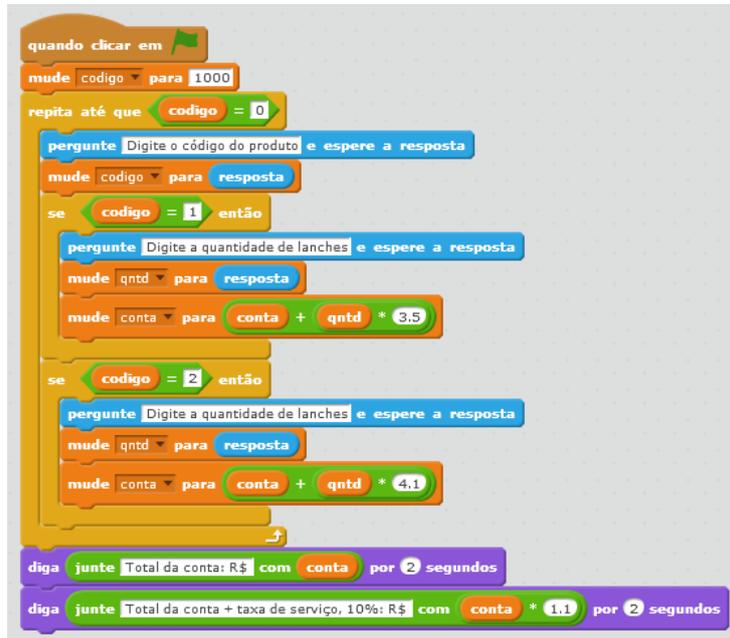


Figura 1: Programa em Scratch para o Sistema de Lanchonete.

```

1 algoritmo "Sistema de Lanchonete"
2 var
3   codigo, quantidade: inteiro
4   conta: real
5 inicio
6   codigo <- 1000
7   enquanto codigo <> 0 faça
8     escreval("Digite o código do produto")
9     leia(codigo)
10    se(codigo =1) entao
11      escreval("Digite a quantidade de lanches")
12      leia(quantidade)
13      conta <- conta + (quantidade * 3.5)
14    fimse
15    se(codigo =2) entao
16      escreval("Digite a quantidade de lanches")
17      leia(quantidade)
18      conta <- conta + (quantidade * 4.1)
19    fimse
20  fimenquanto
21  escreval("Total da conta: R$ ", conta)
22  escreval("Total da conta + taxa de serviço de 10%: R$ ", conta * 1.1)
23 fimalgoritmo

```

Figura 2: Programa em Portugol

```

1 main(void){
2   //variáveis
3   int quantidade,codigo;
4   float conta;
5   //Inicio
6   codigo = 1000;
7   while(codigo!=0){
8     printf("Digite o codigo produto: ");
9     scanf("%d", &codigo);
10    if(codigo=1){
11      printf("Digite a quantidade de lanches: ");
12      scanf("%d",&quantidade);
13      conta = conta+(quantidade*3.5);
14    }
15    if(codigo=2){
16      printf("Digite a quantidade de lanches: ");
17      scanf("%d", &quantidade);
18      conta = conta+(quantidade*4.1);
19    }
20  }
21  printf("Total da conta: R$ %f \n", conta);
22  printf("Total da conta + taxa de serviço, 10%: R$ %f", conta*1.1);
23 }

```

Figura 3: Programa em Linguagem C.

A Figura 2 apresenta a resolução do problema em algoritmo estruturado portugol no ambiente *VisuAlg*. Cada linha do código em portugol é equivalente a cada peça ou comando que foi descrito em *Scratch*. A diferença básica das duas plataformas é que em uma o código é digitado (portugol) e em outra o código é construído a partir de blocos ou peças, clicando e arrastando (*Scratch*). A implementação do algoritmo em linguagem C é apresentada na Figura 3. As estruturas de repetição nestes códigos (Figuras 2 e 3) são diferentes do código em *Scratch* (Figura 1), pois o *loop* é executando enquanto codigo for diferente de 0. Os comandos enquanto ... faça e while utilizam, respectivamente, em portugol codigo <> 0, em linguagem C codigo != 0.

CONCLUSÃO

Neste trabalho apresentamos a metodologia de ensino aplicada pela UFSB aos estudantes que nunca tiveram contato com algoritmos e programação de computadores. A metodologia apresentada utiliza as ferramentas *Scratch* e *VisuAlg* (algoritmo em português), aplicando-as em uma situação-problema e comparando-as entre si, demonstrando o processo de aprendizagem e evolução que os estudantes passaram para chegar a aprender definitivamente uma linguagem de programação, no caso a Linguagem C.

Concluimos que a metodologia proposta é uma forma eficaz de aprendizagem, já que as ferramentas utilizadas introduzem uma maneira lúdica, visual e de mais fácil compreensão. Este método faz com que os estudantes consigam identificar e aprender primeiramente a parte lógica do processo, deixando a sua parte sintática ser introduzida aos poucos, causando menos traumas e aumentando o gosto pelo estudo de programação no meio acadêmico.

REFERÊNCIAS

[1] **Acerca do *Scratch***. Disponível em: <https://scratch.mit.edu/about/>. Acesso em: 12 de agosto 2016.

[2] **Acerca do *Visualg***. Disponível em: <http://www.apoioinformatica.inf.br/produtos/visualg> acesso em: 31 de agosto de 2016.

[3] **Acerca do *DevC++***. Disponível em: <https://sourceforge.net/projects/orwelldevcpp/> Acesso em: 06 de setembro de 2016.

ALENCAR, M. A. S.; LOBO, F. L.; Moreira, J. L. G.; Souza, M. B. **Uma Abordagem Metodológica voltada para o Ensino-Aprendizagem de Algoritmos**. Porto Alegre – RS: CINTED-UFRGS Novas Tecnologias na Educação, V. 11 N° 1, Julho, 2013

DIAS, K. L.; SERRÃO, M. L. **A Linguagem *Scratch* no Ensino de Programação: Um Relato de Experiência com Alunos Iniciantes do Curso de Licenciatura em Computação**. Brasília – DF: XXXIV Congresso da Sociedade Brasileira de Computação, Julho, 2014.

SCHILDT, Herbert. **C Completo e Total**. São Paulo: Makron, 3. ed., 1991.