



MINISTÉRIO DA EDUCAÇÃO
SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA BAIANO
CAMPUS SENHOR DO BONFIM
COLEGIADO DE LICENCIATURA EM CIÊNCIAS DA COMPUTAÇÃO

FELLIPE OLIVEIRA RAMOS

**O *SCRATCH* COMO FERRAMENTA PEDAGÓGICA PARA O
DESENVOLVIMENTO DO PENSAMENTO COMPUTACIONAL SOB A
PERSPECTIVA DA APRENDIZAGEM SIGNIFICATIVA**

Senhor do Bonfim/BA
2015

FELLIPE OLIVEIRA RAMOS

**O *SCRATCH* COMO FERRAMENTA PEDAGÓGICA PARA O
DESENVOLVIMENTO DO PENSAMENTO COMPUTACIONAL SOB A
PERSPECTIVA DA APRENDIZAGEM SIGNIFICATIVA**

Trabalho de Conclusão do Curso
apresentado ao Instituto Federal de
Educação, Ciência e Tecnologia,
Campus Senhor do Bonfim, como
requisito para obtenção do título de
Licenciado em Ciências da
Computação.

Orientadora: Prof^a Msc. Lilian da Silva
Teixeira.

Senhor do Bonfim/BA
2015

FELLIPE OLIVEIRA RAMOS

**O SCRATCH COMO FERRAMENTA PEDAGÓGICA PARA O
DESENVOLVIMENTO DO PENSAMENTO COMPUTACIONAL SOB A
PERSPECTIVA DA APRENDIZAGEM SIGNIFICATIVA**

FOLHA DE APROVAÇÃO

Aprovado em: ____/____/2015.

Banca Examinadora:

Prof^a. Msc. Lilian da Silva Teixeira
Orientadora
Instituto Federal Baiano - IF Baiano

Prof^a. Msc. Elane Souza da Silva
Instituto Federal Baiano - IF Baiano

Prof. Msc. José Aurimar dos Santos Angelim
Instituto Federal Baiano - IF Baiano

Senhor do Bonfim/BA
2015

O SCRATCH COMO FERRAMENTA PEDAGÓGICA PARA O DESENVOLVIMENTO DO PENSAMENTO COMPUTACIONAL SOB A PERSPECTIVA DA APRENDIZAGEM SIGNIFICATIVA

Fellipe Oliveira Ramos¹
Lilian da Silva Teixeira²

Resumo

Este artigo é resultado da análise de um minicurso intitulado “*Scratch* e Desenvolvimento de Jogos” oferecido a alunos de classes do Ensino Médio da Educação Básica. O objetivo principal foi analisar o desempenho de práticas que exploram o Pensamento Computacional com o *Scratch*. A metodologia que orientou o trabalho foi embasada na abordagem qualitativa, de caráter exploratório, e como principal instrumento utilizou-se a realização de minicurso com os alunos do Ensino Médio. Conclui-se que os referidos alunos são capazes de interagir e produzir objetos digitais num ambiente de linguagem de programação com êxito apresentando especificidades peculiares à sua condição de estudantes da educação básica.

Palavras-Chave: Pensamento Computacional, Aprendizagem Significativa, *Scratch*.

INTRODUÇÃO

A difusão das Tecnologias Digitais de Informação e Comunicação (TDIC) nas atividades humanas nestas últimas décadas, tem provocado discussões relacionadas à inserção dessas tecnologias na educação desde às etapas iniciais da educação básica. É possível registrar em muitas escolas a tímida inserção do ensino de Informática como componente curricular. Geralmente o trabalho desenvolvido na disciplina de Informática apresenta uma perspectiva meramente instrucionista, de forma que os estudantes são conduzidos a uma postura passiva diante do computador, manipulando a máquina superficialmente, aprendendo as instruções básicas de editores de texto e/ou planilhas eletrônicas.

O CSTA K-12 Computer Science Standards (CSTA, 2011) destaca a importância do desenvolvimento de habilidades computacionais na Educação Básica

¹ Licenciando em Ciências da Computação - Instituto Federal Baiano. Bolsista PIBID - LCC.

² Professora orientadora. Doutoranda em Educação - Universidade do Estado da Bahia. Docente da Licenciatura em Ciências da Computação - Instituto Federal Baiano.

que promovam a capacidade de resolução de problemas, para que possam auxiliar outras ciências, e motivar os alunos.

Para tanto, inúmeros estudos têm sido desenvolvidos a fim de demonstrar as possibilidades e a necessidade de inserção de conceitos de Computação na educação básica, pois diferentemente da Informática, a Computação abarca a busca de soluções para problemas, a criação de softwares e o desenvolvimento dessas ferramentas.

Algumas iniciativas de inserção da Computação relatam experiências bem sucedidas do ensino de lógica de programação, estruturação de algoritmos e abstração de conceitos computacionais a alunos do Ensino Fundamental e Médio. Por meio da exploração de ambientes de linguagem de programação e da exploração de jogos digitais, os alunos são estimulados à exploração do raciocínio lógico necessário para a construção de algoritmos (MARQUES *et al.*, 2011; RAPKIEWICZ *et al.*, 2006).

Estudos como os de Oliveira (2009); Souza e Lencastre (2013) comprovam a eficiência da promoção do ensino que promove as habilidades do Pensamento Computacional por meio do *Scratch*. A inserção da Computação na educação Básica, pode se dar por meio da perspectiva do desenvolvimento de atividades que exploram o Pensamento Computacional. De acordo com as Diretrizes Curriculares Nacionais para os cursos de graduação em Computação, no Parecer nº 136/2012, “a introdução do Pensamento Computacional e algorítmico na educação básica fornece os recursos cognitivos necessários para a resolução de problemas, transversal a todas as áreas do conhecimento” (BRASIL, 2012).

Diante do exposto, o presente trabalho apresenta como problemática principal a seguinte questão: Como se dá o desempenho de estudantes do Ensino Médio no desenvolvimento de atividades com a linguagem de programação *Scratch*? Para tanto, objetivou-se analisar o desempenho de práticas que exploram o Pensamento Computacional sob a perspectiva da Aprendizagem Significativa por meio de um minicurso sobre a linguagem de programação *Scratch* desenvolvido com alunos de turmas do 1º ao 3º ano do Ensino Médio do Colégio Estadual Luís Eduardo Magalhães, na cidade de Senhor do Bonfim/BA.

1. PENSAMENTO COMPUTACIONAL: UMA AMPLA COMPREENSÃO DO AMBIENTE DIGITAL E TECNOLÓGICO

O termo Pensamento Computacional surge pela primeira vez com Wing (2006), ao afirmar que ele se constrói nos poderes e nos limites dos processos computacionais, sejam eles executados por um ser humano ou por uma máquina. Para a autora, os métodos computacionais proporcionam a resolução de problemas e o desenho de sistemas que não seriam resolvidos pelo ser humano. Portanto, o Pensamento Computacional envolve a formulação de problemas e suas respectivas soluções, que são representadas de forma que possam ser realizadas por agentes de processamento de informação. Sob essa perspectiva as ferramentas computacionais são utilizadas a fim de transformar determinado problema aparentemente difícil em um processo que, auxiliado pelas ferramentas computacionais, possa ser resolvido mais facilmente.

A The Royal Society (2012, p. 29) concebe Pensamento Computacional como “o processo de reconhecimento de aspectos da computação no mundo que nos rodeia, e de aplicação de ferramentas e técnicas da Ciência da Computação³ a fim de entender e analisar sistemas e processos naturais e artificiais”. Ou seja, pensar através de elementos computacionais proporciona ao ser humano a compreensão do ambiente digital e tecnológico em que este está inserido, além de dar meios para que ele interfira neste ambiente.

Na educação básica, especificamente no Ensino Médio, atividades na área de Pensamento Computacional promovem uma formação mais ampla do sujeito para enfrentar os desafios da contemporaneidade. Além de proporcionar ao educando aprendizagens na área de leitura, aritmética, assim como um acentuado desenvolvimento do pensamento analítico. (CARVALHO *et al.*, 2013)

Brennan e Resnick (2012, p. 3) identificaram sete conceitos úteis no desenvolvimento de atividades com o Pensamento Computacional, que são comuns em linguagens de programação, mas que também podem ser explorados por meio da linguagem de programação visual *Scratch*. São eles:

- Sequências: instruções que expressam uma tarefa ou atividade a ser executada pelo computador;
- Loops: execução de uma mesma sequência várias vezes por meio de iterações;

³ Ciência que estuda técnicas, metodologias e instrumentos computacionais a fim de automatizar processos, desenvolvendo soluções baseadas no uso do processamento digital.

- Eventos: para que determinada ação ocorra é necessário que uma outra ação aconteça previamente;
- Paralelismo: sequências de instruções que acontecem ao mesmo tempo;
- Condicionais: tomada de decisões baseadas em certas condições;
- Operadores: suporte a operações matemáticas;
- Dados: armazenagem, recuperação e atualização de valores por meio de variáveis e listas.

Por meio da implementação desses conceitos o sujeito será capaz de desenvolver animações, jogos, ou outras atividades no *Scratch*, de forma acessível e criativa. Essas atividades devem proporcionar ao aluno além do pensar e aprender, o aprender como se aprender.

2. **SCRATCH: UMA FERRAMENTA CRIATIVA E ACESSÍVEL**

O *Scratch* tem sido uma das ferramentas mais utilizadas na promoção do Pensamento Computacional em ambientes escolares. É uma linguagem de programação *open source*⁴, colaborativa idealizada e desenvolvida por uma equipe de investigação do *Media Laboratory* do *Massachusetts of Institute of Technology* (MIT). Por ser um ambiente de programação visual permite a criação de projetos ricos por meio de mídias interativas, sendo possível a criação de games, histórias animadas, tutoriais, simulações, e outros, por meio de seus blocos programáveis.

Divulgado em maio de 2007, o *Scratch* começou a ser desenvolvido em 2003. Sua primeira versão era indicada para jovens de 8 a 16 anos, todavia segundo informações do site do MIT é “utilizado por pessoas de todas as idades, inclusive crianças mais jovens”, que “aprendem a pensar de forma criativa, trabalhar de forma colaborativa e pensar de forma sistemática”. Segundo Maloney *et al.* (2010), este projeto inicial de *Scratch* foi motivado por necessidades e interesses de jovens de centros pós-escolares de computação como o *Intel Computer Clubhouse*⁵ que, por meio de seus ambientes criativos estimula jovens ao desenvolvimento de novas habilidades e ideias tecnológicas.

Um dos principais objetivos do *Scratch* é introduzir noções de linguagem de programação àquelas pessoas que não possuem experiência com ambientes de linguagens de programação. Ao exigir de seu usuário decisões óbvias através da

⁴ Termo destinado a softwares de código aberto que disponibilizam seu código-fonte publicamente para fins de estudo, mudança e melhoria de seu design

⁵ Programa comunitário da Intel, no qual uma comunidade de aprendizes utiliza a tecnologia como uma ferramenta de aprendizado e expressão criativa.

utilização de seus blocos visuais, ele apresenta *layout* simples, com janela única, e utilização mínima de comandos, proporciona aos jovens a manipulação de mídias através da exploração e partilha de suas produções, sem a necessidade de compreensão de uma sintaxe complexa como possuem outras linguagens de programação. (MALONEY *et al.*, 2010).

Os projetos em *Scratch* são formados por um palco e vários objetos. O palco é como um plano de fundo onde serão executadas as ações dos objetos, que são associados a sons, imagens, variáveis. Os comandos que dão vida aos objetos precisam somente ser arrastados e soltos no local específico, onde são encaixados uns nos outros. Ao serem combinados sempre formam programas corretos, inexistindo erros sintáticos. (MALAN; LEITNER, 2007).

Nas imagens a seguir percebe-se as sintaxes necessárias para emitir uma mesma informação nas linguagens Java⁶ e *Scratch*. Além disso identifica-se a utilização de uma grande quantidade de códigos na primeira, e a praticidade apresentada pela segunda, ao utilizar somente dois blocos para tal.

Figura 1 – Programa “Hello” em Java

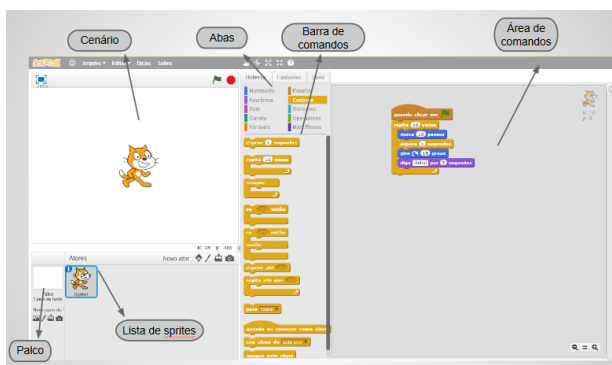
```
class Hello
{
    public static void main(String [] args)
    {
        System.out.println("Hello!");
    }
}
```

Figura 2 – Programa “Hello” em *Scratch*



A imagem abaixo mostra a tela principal do *Scratch*, e destaca os seus principais ambientes: Cenário, Abas, Barra de Comandos, Área de Comandos, Palco e Lista de Sprites.

Figura 3 – Tela principal do *Scratch*



⁶ Linguagem de programação orientada a objetos, desenvolvida na década de 90 pela empresa *Sun Microsystems*. É uma das linguagens de programação mais utilizadas na atualidade.

Apesar do *Scratch* ser uma produção relativamente recente, pode-se afirmar que sua difusão em todo o mundo foi muito rápida. Graças à comunidade de colaboradores, atualmente está disponível em inúmeros idiomas. Atualmente o *Scratch* encontra-se na versão 2.0, e pode ser utilizado de forma *online*⁷, e também, pode ser feito o *download* gratuito do editor *offline*⁸, disponível para os sistemas operacionais *Mac OS*, *Windows* e *Linux*.

O *Scratch* pode ser considerado um ambiente de ensino e aprendizagem potencialmente significativo, mas para que isso ocorra se faz necessário compreender os fundamentos teóricos do conceito de Aprendizagem Significativa.

3. APRENDIZAGEM SIGNIFICATIVA NA PRÁTICA PEDAGÓGICA CONTEXTUALIZADA

Concebida por David Ausubel na década de sessenta, a teoria da Aprendizagem Significativa é representada pela interação cognitiva existente entre um conhecimento prévio e um novo conhecimento. Dessa forma, ela ocorre em “qualquer atividade na qual a representação cognitiva de experiência prévia e os componentes de uma situação problemática apresentada são reorganizados a fim de atingir um determinado objetivo” (AUSUBEL, 1968).

Moreira (1999) considera que a Aprendizagem Significativa opera a partir do momento em que o sujeito faz uso de conceitos aprendidos anteriormente como base para a aprendizagem de outros conceitos.

Para que a Aprendizagem Significativa ocorra de fato, o material utilizado no processo de ensino e aprendizagem deve ser potencialmente significativo; o aluno precisa possuir subsunçores que possam ser relacionados com os novos conteúdos de forma não arbitrária e substantiva; e para que esses novos conteúdos sejam absorvidos dessa forma, o aluno precisa estar disposto a relacionar o novo conteúdo a algum conteúdo já existente em sua estrutura cognitiva, portanto ele precisa estar predisposto a aprender. (AUSUBEL *et al.*, 1980).

Além dos princípios apresentados, é preciso que o professor apresente os conteúdos aos alunos de uma forma que estes percebam utilidade naquilo que está

⁷ <https://scratch.mit.edu>

⁸ <https://scratch.mit.edu/scratch2download>

sendo aprendido. Isso contribuirá para que ele se interesse nos novos conteúdos, e se interesse no aprendizado.

Uma prática pedagógica que se inspira nos princípios da teoria da Aprendizagem Significativa proporciona que os novos conceitos demorem de ser esquecidos, e caso esquecidos sejam mais facilmente lembrados. Desse modo, ao trabalhar conteúdos que requerem do sujeito novas competências e habilidades, no contexto da Aprendizagem Significativa, o indivíduo transferirá rapidamente estes conceitos para a resolução de problemas outros a serem apresentados, mesmo que estes estejam configurados em contextos diferentes daqueles aprendidos.

4. CAMINHO METODOLÓGICO

O presente trabalho foi desenvolvido sob a metodologia de abordagem qualitativa. “A pesquisa nessa área lida com seres humanos que, por razões culturais de classe, de faixa etária, ou por qualquer outro motivo, têm um substrato comum com o investigador.” (MINAYO, 2011, p. 21)

A pesquisa qualitativa é mais utilizada pelas ciências da área de humanas, no caso em questão, por se tratar de uma pesquisa que nasce da conclusão de um curso de licenciatura, é oportuna pela necessidade de se fazer uma análise voltada para um grupo particular, alunos de Ensino Médio, de contexto específico que permite ao pesquisador compreender também realidades mais amplas, porém, sem atribuir determinismos. Um trabalho com uma problemática dessa natureza requer uma ação interventiva junto aos sujeitos a fim de constatar-se a aplicabilidade e viabilidade da inserção do Pensamento Computacional no Ensino Médio através do *Scratch*.

Desse modo, foi desenvolvida a metodologia de pesquisa de campo de caráter exploratório, que apresenta um aspecto de pesquisa empírica, em que aplicou-se um procedimento sistemático (execução de minicurso) para a obtenção de dados de análise. Na pesquisa de campo de caráter exploratório,

obtem-se frequentemente descrições tanto qualitativas quanto quantitativas do objeto de estudo, e o investigador deve conceituar as inter-relações entre as propriedades do fenômeno, fato ou ambiente observado. (MARCONI; LAKATOS, 2010, p. 71).

Justifica-se a escolha da metodologia da pesquisa de campo exploratória por se tratar de sujeitos pertencentes a uma escola em que um dos pesquisadores já atua

há aproximadamente dois anos com atividades referentes ao ensino de informática e computação, através do Programa de Bolsa de Iniciação a Docência - PIBID, da Licenciatura em Ciências Computação do Instituto Federal Baiano - *campus* Senhor do Bonfim.

A experiência do trabalho com conceitos básicos da área do Pensamento Computacional por meio do *Scratch* se deu através do minicurso intitulado “*Scratch* e Desenvolvimento de Jogos”, oferecido a 12 alunos de uma escola pública estadual, e com duração de 16 horas. O minicurso foi dividido em duas etapas: Definições, Conceitos Iniciais de Computação, Algoritmos; Raciocínio Lógico; e, Programação com o *Scratch*.

Após a execução das atividades do minicurso, aplicou-se um questionário semiaberto com a finalidade de obter uma devolutiva dos alunos, pois para perceber o impacto dessa nova aprendizagem para a formação desses sujeitos fazia-se necessário verificar suas interpretações sobre a experiência com o *Scratch*.

4. 1 Execução do Minicurso “*Scratch* e Desenvolvimento de Jogos”

4.1.1 Etapa 1: Definições, Conceitos Iniciais de Computação, Algoritmos e Raciocínio Lógico

Nesta etapa abordou-se fundamentos introdutórios da computação, como evolução do computador, números binários, *bits* e *bytes*, *hardware*, *software*, *peopleware* e *firmware* e processamento de dados. Um dos objetivos desta etapa foi apresentar aos alunos, mesmo que de forma introdutória, fundamentos básicos da computação, por meio de conceitos discutidos por aqueles que fazem cursos superiores na área de computação. Estes conceitos foram apresentados aos alunos de forma teórica, mas relacionados a situações práticas, e contextualizadas. Posteriormente eles foram instigados a conceituar computação e informática e a identificar o uso de elementos computacionais na sociedade, seus benefícios e malefícios.

Também foram apresentados aos alunos conteúdos diretamente ligados à programação, conceitos e exemplos de algoritmos, bem como exemplos e utilização de linguagens de programação. Foi discutido que os algoritmos representam uma sequência lógica do passo a passo necessário para a execução de determinadas

tarefas, que são primordiais na área de programação, inclusive na a construção de programas no *Scratch* e estão presentes no conceito de Pensamento Computacional.

Após conhecer estes conceitos os alunos foram desafiados a utilizar o raciocínio lógico a fim de desenvolver algoritmos: “ir para a escola”, “trocar pneu” e “trocar uma lâmpada”.

4.1.2 Etapa 2: Programação com o *Scratch*

Por meio de desafios lançados aos alunos, que envolviam o desenvolvimento de jogos e programas no *Scratch*, buscou-se seguir os conceitos de sequências, *loops*, eventos, paralelismo, condicionais, operadores e dados, propostos por Brennan e Resnick (2012) no desenvolvimento de atividades com o Pensamento Computacional. Dessa forma, esta etapa foi dividida em dois momentos que abordaram conceitos teóricos de programação seguidos de desafios no *Scratch* a serem desenvolvidos por meio da utilização desses conceitos.

No primeiro momento abordou-se os conceitos de Estruturas de Controle de Seleção (simples e composta) e Estruturas de Controle de Repetição (pré-teste, pós-teste e variável de controle), relacionando-os com os blocos existentes no *Scratch*. No Desafio 1 foi proposto o desenvolvimento de uma animação composta por dois personagens, e um palco da escolha do aluno, sendo que os personagens deveriam trocar suas fantasias por meio da utilização das Estruturas de Seleção e/ou Repetição. Nesse momento inicial os alunos foram motivados a descobrir as funcionalidades do *Scratch* de forma autônoma.

Na sequência foram discutidos os conceitos de programação: paralelismo, operadores e armazenagem, recuperação e atualização de dados por meio de variáveis e listas, mas também foram desenvolvidas práticas no ambiente do *Scratch* juntamente com uma explanação sobre a importância da compreensão e da aplicação de coordenadas no *Scratch* por meio da utilização de blocos específicos.

Para aplicar estes conceitos foi proposto o Desafio 2, no qual os alunos desenvolveram um jogo simples de labirinto. Teriam que desenhar um caminho por meio da ferramenta “plano de fundo”; inserir 7 atores (um teria movimentos e percorreria o caminho, afim de caçar os outros seis atores que estariam espalhados no palco); implementar paralelismo; utilizar blocos com operadores; e implementar um placar por meio da criação de uma variável que contabilizaria os atores tocados no caminho.

A importância das coordenadas foi observada por meio do estabelecimento de uma regra que garantia ao ator principal do jogo retornar ao seu ponto inicial quando tentasse sair do caminho estabelecido.

5. CRIATIVIDADE E LIBERDADE DE CRIAÇÃO NA EXPLORAÇÃO DO SCRATCH

Todo o trabalho foi organizado em quatro momentos. Dois momentos referentes a primeira etapa, e dois momentos referentes a segunda etapa.

Na primeira etapa do minicurso as atividades foram teóricas, e culminaram com a produção de algoritmos. A princípio os alunos demonstraram resistência, principalmente em relação aos conceitos computacionais no plano teórico, porém a medida que estes conceitos eram relacionados às atividades do cotidiano os alunos envolviam-se mais demonstrando surpresa ao compreenderem a gênese de construção de softwares a partir da lógica das sequências que compõe um algoritmo.

Todos os alunos envolvidos na atividade produziram algoritmos relacionados a atividades do cotidiano. É interessante observar que cada sujeito produziu sua própria lógica sequencial, alguns mais elaborados que outros, mas todos alcançaram o resultado esperado.

A problematização foi um fator importante para fazer os sujeitos refletirem sobre o que estavam produzindo, a medida que organizavam suas sequências, questionava-se se estes não estavam esquecendo de inserir na sequência “passos” importantes para se alcançar o fim pretendido.

Na figura a seguir apresenta-se o algoritmo “trocar pneu”, desenvolvido por um dos alunos que participaram do minicurso.

Figura 04 - Produção Algoritmo “Trocar Pneu”- Aluno A⁹

The image shows a handwritten list of steps for changing a tire, written on lined paper. The steps are numbered 1 through 8. The handwriting is in cursive and somewhat informal. The steps are: 1- Pegar o macaco, 2- Pôr o embaixo do carro e levanta-lo, 3- Pegar a chave de roda, 4- Desmontar os parafusos de roda, 5- Tirar o pneu, 6- Encaixar o outro pneu, 7- Enroscar os parafusos, 8- Baixar o macaco e tirá-lo debaixo do carro.

Trocar pneu:
1- Pegar o macaco
2- Pôr o embaixo do carro e levanta-lo
3- Pegar a chave de roda
4- Desmontar os parafusos de roda
5- Tirar o pneu
6- Encaixar o outro pneu
7- Enroscar os parafusos
8- Baixar o macaco e tirá-lo debaixo do carro

Observa-se que o Aluno A, trouxe todos os passos necessários para executar a ação de forma correta, porém no passo 2, e no 8, ele acaba integrando duas ações distintas que poderiam estar separadas.

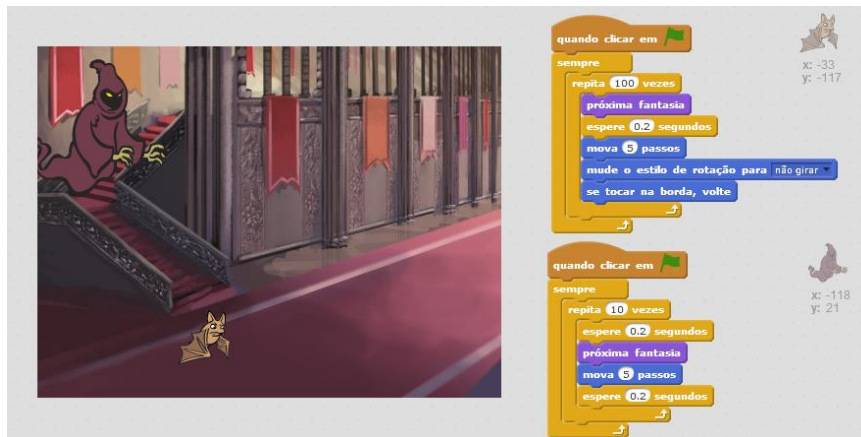
Ao encerrar as atividades da primeira etapa, iniciou-se a fase de aproximação com os conteúdos de programação necessários para a prática com a linguagem *Scratch*. Ao apresentar os conceitos computacionais voltados à programação, os sujeitos voltavam a mostrar as mesmas resistências do momento teórico da primeira etapa.

Os alunos no ambiente do laboratório de informática apresentam um comportamento de grande expectativa para a interação direta com a máquina e seus recursos digitais, ao serem conduzidos a uma discussão teórica nesse espaço físico ficam inquietos e ansiosos para estabelecer o contato direto com a máquina.

A seguir apresentam-se as figuras 05 e 06, que são produções dos alunos após o momento de apresentação do *Scratch* e execução do Desafio 1. Os alunos foram desafiados a produzir uma animação, conforme a descrição aqui apresentada na subseção 4.1, sem a intervenção do orientador do minicurso, os alunos foram estimulados à exploração do ambiente da linguagem *Scratch*.

⁹ Para manter sigilo quanto à identificação dos sujeitos utilizou-se das letras do alfabeto em maiúsculo para identificar cada um dos alunos que participaram do minicurso.

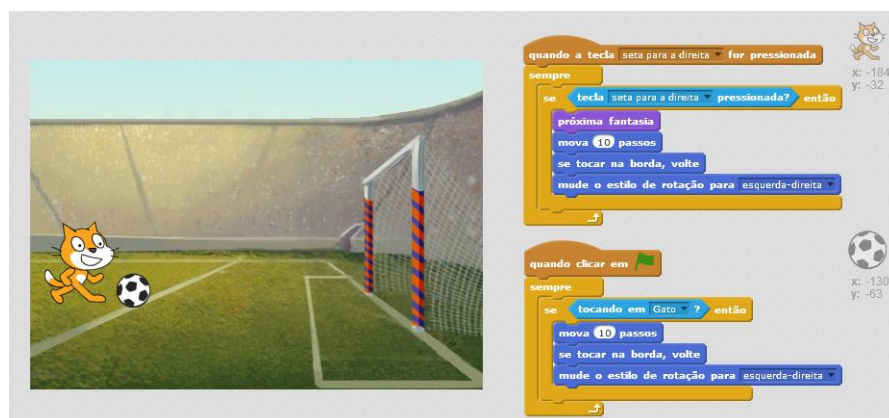
Figura 05 – Palco e Comandos - Produção do Aluno B



Ao observar-se a estrutura construída pelo Aluno B percebe-se que ele cumpriu a atividade proposta, pois modificou o plano de fundo, inseriu dois atores e implementou a troca de suas fantasias por meio da utilização das Estruturas de Repetições “sempre” e “repita”.

Na animação abaixo, o Aluno C, assim como o Aluno B, concluiu o desafio proposto. O Aluno C descobriu de forma autônoma a aplicação dos operadores na Estrutura de Seleção “se”, o que proporcionou o controle dos blocos a serem executados, e também descobriu que para utilizar o bloco “próxima fantasia” era necessário utilizar o bloco “mova x passos”. Além disso, ele utilizou blocos que não foram solicitados como “se tocar na borda volte”, o que promoveu uma melhor execução da animação. Observamos aqui, portanto, a criatividade e a liberdade de criação proporcionadas pelo Scratch, definidas por Pinto (2010) como algumas das potencialidades desta linguagem de programação.

Figura 06 – Palco e Comandos - Produção do Aluno C

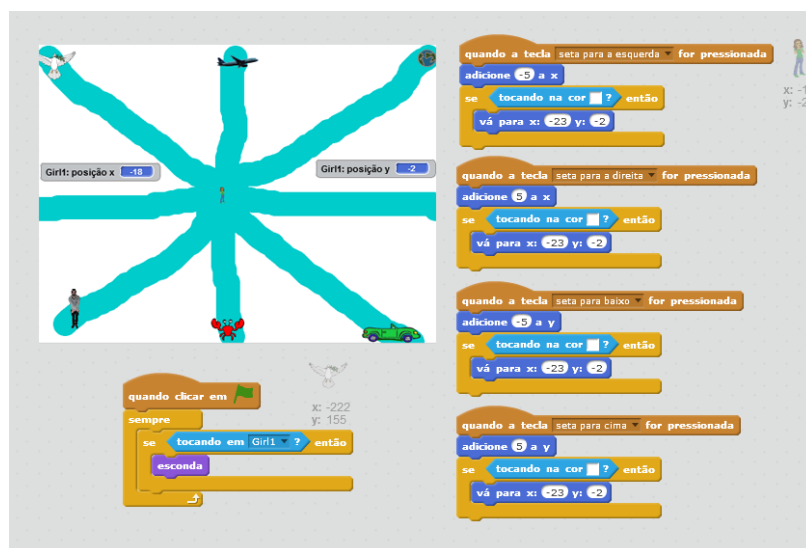


O que mais chamou atenção no desenvolvimento dos desafios foi a postura criativa e autônoma dos alunos, quando se deparavam com os conflitos na organização da estrutura lógica dos blocos rapidamente encontravam as soluções mais adequadas e prosseguiram com seus projetos de animação.

Na sequência, após a apresentação teórica de conteúdos relativos a programação, partiu-se para a execução do Desafio 2, também descrito aqui na subseção 4.1. As figuras a seguir apresentam os produtos obtidos através desta atividade.

Como pode ser visto na figura 07, o Aluno D cumpriu o Desafio 2 ao inserir todos os elementos solicitados. O conceito de paralelismo é aplicado nos atores que possuem evento comum para serem executados, no caso, “quando clicar em bandeirinha”. Esses atores ficam estáticos na tela, e somem da tela ao serem tocados pelo ator “girl”. Para tanto o Aluno D descobriu de forma autônoma o bloco “esconda” e utilizou a mesma sequência de blocos para os atores estáticos, que pode ser vista no ator “pomba”. Além disso, um fato relevante foi que o Aluno D agregou os conceitos aprendidos por meio do Desafio 1 ao implementar blocos de Estruturas Controle, utilizando-se de operadores.

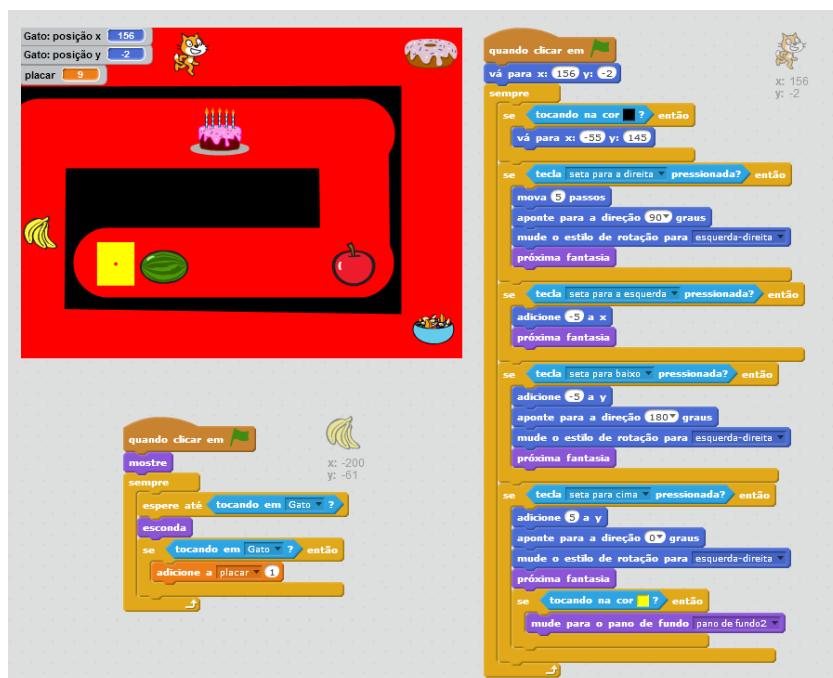
Figura 07 – Palco e Comandos - Produção do Aluno D



Ao desenvolver o Desafio 2, o Aluno E, assim como o Aluno D, utilizou conceito de paralelismo também fez uso de mecanismos aprendidos anteriormente ao utilizar os blocos “próxima fantasia” e Estruturas Condicionais e Estruturas de Repetição.

O que difere a produção do primeiro (Aluno D), para a do segundo (Aluno E) foi a forma com que ele utilizou a Estrutura de Controle “se”, e agrupou todos os blocos de comandos em eventos distintos. Esse fato revela que a Aprendizagem Significativa ocorre de maneira muito específica para cada sujeito. Na experiência aqui descrita todos os sujeitos foram envolvidos no mesmo trabalho, o conteúdo foi direcionado igualmente a todos, assim como as instruções indicadas foram as mesmas, todavia cada aluno traçou suas próprias estratégias para construir um novo conhecimento, pois, como afirma Ausubel (1968), cada sujeito desenvolve a situação proposta de uma forma específica, levando em consideração seus conhecimentos prévios.

Figura 08 – Palco e Comandos - Produção do aluno E



O Aluno E (figura 08) conseguiu implementar o bloco “mudar estilo de rotação”, e também uma variável capaz de armazenar o número de atores, a fim de exibi-la na tela e informar o número de atores tocados pelo ator que se movimenta. Dessa forma o Aluno E concluiu o desafio proposto em sua totalidade. O Aluno D, apesar de ter feito uso de uma estrutura lógica diferente, e não ter implementado a variável também teve êxito em sua produção.

Após o desenvolvimento de todas as atividades previstas no minicurso, foi aplicado um questionário semiaberto aos alunos cursistas a fim de avaliar o trabalho desenvolvido sob a perspectiva dos próprios sujeitos.

Nesse instrumento foi possível perceber as dificuldades, as situações mais significativas e os impactos da experiência com o minicurso para o desenvolvimento intelectual dos alunos do Ensino Médio.

Ao serem questionados se *Scratch* pode ser utilizado nas disciplinas curriculares do Ensino Médio, 100% dos alunos afirmaram acreditar que isso é possível. Esse fato nos chama atenção, demonstrando que os sujeitos conseguiram fazer uma relação das atividades desenvolvidas no *Scratch* com os conteúdos curriculares das disciplinas propedêuticas. Observemos a resposta do Aluno B: *“Pode ser utilizado na forma de raciocínio... É isso que o Scratch faz, você raciocinar para montar algo”*.

No que se refere aos aspectos de usabilidade do ambiente da linguagem de programação *Scratch*, observa-se como um aspecto positivo o fato dele ser lúdico e de fácil manipulação. O Aluno B afirma que: *“é legal estar produzindo algo como uma atividade e ao mesmo tempo se divertindo”*. Já o Aluno F considerou *“a facilidade para construir os jogos”*. Por ser uma linguagem de programação mais simples, o *Scratch* possibilita que o aluno pense na produção por ele mediada, sem que pense, necessariamente, na programação em si (RESNICK, 2007).

Quando indagados se após o minicurso ainda continuariam utilizando o *Scratch*, todos os alunos relataram que se identificaram com o ambiente e continuarão a utilizá-lo, conforme percebe-se na fala do Aluno C: *“É provável que sim. Achei divertido mexer com a criação de um jogo, ao invés de apenas jogar.”*. Enquanto isso, o Aluno D atesta que pretende *“tentar melhorar o que foi aprendido”*.

Um dos limites desta produção é o fato de não ter sido possível trabalhar com todos os alunos da escola pesquisada, desse modo, o posicionamento dos alunos pesquisados possivelmente não revela a totalidade, uma vez que os sujeitos foram selecionados a partir do critério de identificação com atividades voltadas as TDIC.

CONSIDERAÇÕES FINAIS

A realização do minicurso sobre *Scratch* com os alunos do Ensino Médio proporcionou uma rica experiência interventiva aos alunos que foram contemplados. As atividades desenvolvidas foram voltadas a exploração do Pensamento

Computacional, e a contextualização constante dos conceitos e conteúdos apresentados contribuíram para a aprendizagem significativa.

Como foi possível perceber, a utilização do *Scratch* teve uma grande aceitação por parte dos alunos, de forma que quando indagados, todos afirmaram ter aprovado da experiência com a linguagem de programação, confirmaram o desejo em continuar manipulando-a, não relataram grandes dificuldades na sua utilização, e o mais importante, sinalizaram perceber que estavam desenvolvendo atividades que proporcionavam o raciocínio lógico, e conseqüentemente o Pensamento Computacional.

Através da análise foi possível perceber que os alunos de turmas do Ensino Médio são capazes de interagir e produzir objetos digitais num ambiente de linguagem de programação, mesmo sem conhecimentos prévios específicos da área da Computação, e ainda num espaço reduzido de tempo, visto que todas as atividades desenvolvidas ocorreram em 4 dias de trabalho que totalizaram 16 horas de minicurso.

Observou-se ainda, que o *Scratch* é potencialmente significativo, e também que os alunos que participaram do minicurso o frequentaram de iniciativa própria, de forma extraclasse, e estavam predispostos a conhecer os novos conteúdos apresentados. Dessa forma, acredita-se que a realização deste trabalho proporcionou aos alunos a aprendizagem do Pensamento Computacional por meio do *Scratch*, de forma significativa.

Ratificando o que afirma Phillips (2009), constatou-se que os indivíduos que desenvolvem o Pensamento Computacional adquirem aptidão para o desenvolvimento de aplicações, e também competências como o pensamento abstrato, o pensamento algorítmico, o pensamento lógico e o pensamento dimensionável. Conseqüentemente, se continuarem a explorar o *Scratch*, poderão tornar-se sujeitos com competências e habilidades nem sempre alcançadas por sujeitos que nunca tiveram contato com ambientes de linguagem de programação.

Concluindo, foi evidenciado o quanto os conceitos computacionais, e ainda, conceitos da matemática podem ser facilmente aprendidos quando essa aprendizagem ocorre de forma significativa, pois a interação com a linguagem de programação do *Scratch* provoca no sujeito a organização do seu pensamento amparado em noções subsunçoras num processamento de significação do conhecimento a ser construído.

REFERÊNCIAS

AUSUBEL, David Paul. **Educational psychology**: a cognitive view. New York and Toronto: Holt, Rinehart and Winston, p. 685, 1968.

AUSUBEL, David Paul; *et al.* **Psicologia educacional**. Tradução de Eva Nick et al. 2ª ed. Rio de Janeiro: Interamericana, 1980.

BRASIL, Diretrizes Curriculares Nacionais para os cursos de Graduação em Computação. **Parecer nº 136/2012**. Conselho Nacional de Educação. Disponível em < http://portal.mec.gov.br/index.php?option=com_content&id=12991:diretrizes-curriculares-cursos-de-graduacao>. Acesso em: 20 abr. 2015.

BRENNAN, Karen; RESNICK, Mitchel . **New frameworks for studying and assessing the development of computational thinking**. Proceedings of annual meeting of the American Educational Research Association. p. 1-25, Vancouver, Canadá, 2012.

CARVALHO, Luiz Bunte de; *et al.* **Pensamento Computacional no Ensino Médio Mineiro**. Disponível em <<http://www.lbd.dcc.ufmg.br/colecoes/wei/2013/0034.pdf>>. Belo Horizonte, 2013.

MALAN, David J.; LEITNER, Henry H. **Scratch for budding computer scientists**. Proceedings do 38th SIGCSE'07, p. 223-227. Kentucky, 2007.

MALONEY, John; *et al.* **The scratch programming language and environment**. In: Communications of The ACM, 2010. v. 10, n. 4, Article 16.

MARCONI, Marina de Andrade; LAKATOS, Eva Maria. **Técnicas de pesquisa**: planejamento e execução de pesquisas, amostragens e técnicas de pesquisa, elaboração, análise e interpretação de dados. - 7 ed. - São Paulo: Atlas, 2010.

MARQUES, Diego Lopes; *et al.* **Atraindo alunos do ensino médio para a computação**: Uma Experiência Prática de Introdução à Programação utilizando Jogos e Python. In: Anais Do XVII Workshop sobre Educação Informática. Aracaju: Universidade Federal de Sergipe, 2011.

MINAYO, Maria Cecília de Souza; DESLANDES, Suely Ferreira; GOMES, Romeu. **Pesquisa social**: teoria, método e criatividade. – 30 ed. Petrópolis, Rio de Janeiro: Vozes, 2011.

MIT - MASSACHUSETTS INSTITUTE OF TECHNOLOGY. **Para os Pais**. s.d. Disponível em: <<https://scratch.mit.edu/parents/>>. Acesso em: 01 abr. 2015.

MOREIRA, Marco Antonio. **Aprendizagem significativa**. Editora Universidade de Brasília: Brasília, 1999.

OLIVEIRA, Elaine Cecília de Lima. **O Uso do Software Scratch no Ensino Fundamental**: Possibilidades de Incorporação Curricular Segundo Professoras dos Anos Iniciais. PUC-Minas: Belo Horizonte, 2009.

- PHILLIPS, Pat. (2009). **Computational Thinking**.
<http://www.csta.acm.org/ProfessionalDevelopment/sub/CSIT09Presentations/Phillips_Computational.pdf>. Acesso em: 10 jan. 2015.
- PINTO, António Sorte. **Scratch na aprendizagem da matemática no 1.º ciclo do ensino básico**: estudo de caso na resolução de problemas. Braga: Universidade do Minho, 2010
- RAPKIEWICZ, Clevi Elena; *et. al.* **Estratégias Pedagógicas no Ensino de Algoritmos e Programação Associadas ao Uso de Jogos Educacionais**. CINTED-UFRGS, 2006. v. 4 nº 2.
- RESNICK, Mitchel. **Rethinking learning in the digital age**. Massachusetts Institute of Technology, 2007. Disponível em:
<<http://www.media.mit.edu/~mres/papers/wef.pdf>>. Acesso em: 01 abr. 2015.
- SOUZA, Rui Miguel; LENCASTRE, José Alberto. **Desenvolvimento do Pensamento Computacional com Recurso ao Scratch**: Uma Experiência com Alunos do 8º Ano. Universidade do Minho. Atas do XII Congresso Internacional Galego-Português de Psicopedagogia. ISBN: 978-989-8525-22-2. Braga: Universidade do Minho, 2013.
- STANDARDS TASK FORCE. **CSTA K-12 Computer Science Standards**. ACM Computer Science Teachers Association. New York, 2011. Disponível em:
<<http://csta.acm.org/Curriculum/sub/K12Standards.html>>. Acesso em: 14 mar. 2015.
- THE ROYAL SOCIETY. (2012) **Desligar ou reiniciar? O caminho a seguir para a Computação no Reino Unido Escolas**. Disponível em:
<http://royalsociety.org/uploadedFiles/Royal_Society_Content/education/policy/computing-nas-escolas/2012-01-12-Computing-in-Schools.pdf>. Acesso em: 30 abr. 2015.
- THIOLENT, Michel. **Metodologia da pesquisa-ação**. 13. ed. São Paulo: Cortez, 2004. p.14-16.
- WING, Jeannette Marie. **Computational thinking**. In: Communications of The ACM, Março 2006. Vol. 49, nº 3. p. 33-35.

APÊNDICES

APÊNDICE A - PLANO DE AULA 01

I. IDENTIFICAÇÃO

Instituto Federal Baiano – *Campus* Senhor do Bonfim
Licenciatura em Ciências da Computação
Fellipe Oliveira Ramos

II. OBJETIVOS

- Compreender conceitos básicos de programação e algoritmos;
- Perceber a importância das linguagens de programação;
- Construir algoritmos.

III. CONTEÚDOS

- Introdução a Computação
 - Evolução do computador;
 - Números binários,
 - *Bits e bytes*,
 - *Hardware, software, peopleware e firmware*;
 - Processamento de Dados;
 - Pensamento Computacional.
- Introdução a Programação
 - Algoritmos;
 - Linguagens de programação.

IV. METODOLOGIA

Aula expositiva e participativa.

V. RECURSOS DIDÁTICOS

- Computadores;
- Datashow;
- Quadro.

VI. ATIVIDADES

- Participação em momentos de discussão;
- Construção de algoritmos.
 - Algoritmo “Ir para a escola”;
 - Algoritmo “Trocar pneu”;
 - Algoritmo “Trocar a Lâmpada”.

VII. REFERÊNCIAS BIBLIOGRÁFICAS

BONATO, Vanderlei. **Conceitos básicos sobre computadores**. s.d. Disponível em: <http://wiki.icmc.usp.br/images/b/be/Aula1_-_Conceitos_B%C3%A1sicos_2011101.pdf> Acesso em: 20 abr. 2015.

BORATTI, Isaias Camilo; OLIVEIRA, Álvaro Borges de. **Introdução à Programação Algoritmos**. - 3ª ed. – Florianópolis: Visual Books, 2007.

TV ESCOLA. **Bits e Bytes**: Que mundo é esse? Disponível em: <<http://tvescola.mec.gov.br/tve/videoteca-series!loadSerie?idSerie=495>> Acesso em: 20 abr. 2015.

APÊNDICE B - PLANO DE AULA 02

I. IDENTIFICAÇÃO

Instituto Federal Baiano – Campus Senhor do Bonfim

Licenciatura em Ciências da Computação

Fellipe Oliveira Ramos

II. OBJETIVOS

- Construir apresentações de slides utilizando Libre Office Impress

III. CONTEÚDOS

- Apresentação do Scratch
 - Interface;
 - Sprite;
 - Stage
 - Blocos;
 - Costumes;
 - Sons;
 - Aba ferramentas.
- Estruturas de Seleção
 - Simples;
 - Composta.
- Estruturas de Repetição
 - Pré-Teste;
 - Pós-Teste;
 - Variável de controle.

IV. METODOLOGIA

Aula expositiva e participativa.

V. RECURSOS DIDÁTICOS

- Computadores;

- Datashow;
- Quadro.

VI. ATIVIDADES

- Participação em momentos de discussão.
- Desafio 1 no Scratch, com as Estruturas de Controle de Seleção e Repetição
 - Inserir 2 Personagens;
 - Um palco qualquer;
 - Ambos devem trocar fantasia.
 - Utilizar Estrutura de Seleção simples e composta;
 - Utilizar Estrutura de Repetição;
 - Utilizar sensores.

VII. REFERÊNCIAS BIBLIOGRÁFICAS

BORATTI, Isaias Camilo; OLIVEIRA, Álvaro Borges de, **Introdução à Programação Algoritmos**. - 3ª ed. – Florianópolis: Visual Books, 2007.

LOPES, Anita; GARCIA, Guto. **Introdução à Programação**. - 1ª ed. - Rio de Janeiro: Campus, 2002.

APÊNDICE C - PLANO DE AULA 03

I. IDENTIFICAÇÃO

Instituto Federal Baiano – *Campus* Senhor do Bonfim

Licenciatura em Ciências da Computação

Fellipe Oliveira Ramos

II. OBJETIVOS

- Compreender conceitos de programação;
- Implementar no *Scratch* os conceitos aprendidos.

III. CONTEÚDOS

- Paralelismo;
- Operadores
- Dados;
- Coordenadas no Scratch.

IV. METODOLOGIA

Aula expositiva e participativa.

V. RECURSOS DIDÁTICOS

- Computadores;
- Datashow;
- Quadro.

VI. ATIVIDADES

- Desafio 2 no Scratch com Paralelismo, Operadores, Dados e Coordenadas;

Desenhar um caminho;

- Inserir 7 atores (um para percorrer o caminho e os outros espalhados no palco);
- Utilizar Corrdenadas;;
- Ao tentar sair do caminho o ator deverá retornar ao seu ponto inicial;
- Implementar paralelismo;

- Utilizar blocos com operadores;
 - Utilizar variáveis;
 - Adicionar uma lista.
- Participação em momentos de discussão.

VII. REFERÊNCIAS BIBLIOGRÁFICAS

BORATTI, Isaias Camilo; OLIVEIRA, Álvaro Borges de, **Introdução à Programação Algoritmos**. - 3ª ed. – Florianópolis: Visual Books, 2007.

LOPES, Anita; GARCIA, Guto. **Introdução à Programação**. - 1ª ed. - Rio de Janeiro: Campus, 2002.