



ANALISE E DESENVOLVIMENTO DE TÉCNICAS DE ENSINO DE PROGRAMAÇÃO PARA JOVENS E ADOLESCENTES

Acadêmico: Flavio Malvestiti Junior

Orientadora: Lilian J. Meyer Riveros

RESUMO

A disseminação da tecnologia da informação vem trazendo novos desafios, a falta de profissionais qualificados para trabalhar em um nível mais baixo com a nova tecnologia faz com que a mesma seja subutilizada. Programadores têm tido um papel fundamental, desenvolvendo aplicativos e soluções que facilitam nosso dia-a-dia, mas a prática de programação ainda é uma matéria que não está incluída nas grades curriculares das escolas. O objetivo deste projeto de pesquisa é justamente verificar a aceitação de ferramentas e técnicas de programação por parte de crianças do ensino fundamental, analisando a maneira como elas interagem com a ferramenta disponibilizada questionando-as sobre as possibilidades que elas veem no novo conhecimento adquirido em relação aos procedimentos metodológicos, a pesquisa escolhida foi a qualitativa com recorrência em técnicas quantitativas para a obtenção de dados. A análise destes dados será feita por meio de ferramentas estatísticas para os questionários. Os resultados são, a geração de conhecimento e a identificação do interesse pessoal de cada aluno no ensino de programação nas escolas.

Palavras-chave: Programação. Ensino. Escola. Ensino fundamental.

ABSTRACT

The dissemination of information technology has been bringing new challenges, the lack of skilled professionals to work at a lower level with the new technology makes it underutilized. Programmers have had a key role, developing applications and solutions that facilitate our day-to-day, but the programming practice is still a matter that is not included in the school curricular grids. The objective of this research project is precisely to verify the acceptance of tools and techniques of programming by children of elementary school, analyzing the way they interact with the tool provided by questioning them about the possibilities that they see in the new

knowledge acquired in relation to methodological procedures, the survey chosen was qualitative with a recurrence of quantitative techniques for obtaining data. The analysis of these data will be done through statistical tools for the questionnaires. The results are, the generation of knowledge and the identification of each student's personal interest in teaching programming in schools.

Keywords: programming. Teaching. School. Elementary School.

1 INTRODUÇÃO

Os computadores estão cada dia mais presentes em nossas vidas e nas vidas das crianças de nossa sociedade, nas escolas eles veem sendo utilizados como ferramentas que auxiliam o ensino, ferramentas iterativas e aplicativos são utilizados mas para além de se utilizar apenas os aplicativos e programas são disponibilizados, muitas das crianças querem criar seus próprios aplicativos e programas, mas não sabem por onde começar ou não possuem um incentivo ou alguém que os ajude com essa tarefa, há outras nem sabem que há esta possibilidade e os benefícios que saber programar podem lhes trazer.

Dentre os muitos benefícios que a atividade de programação pode trazer as crianças e adultos em geral está a melhora nas atividades envolvendo o raciocínio lógico e formal.

Mas para além de ensinar o uso das ferramentas, é necessária que se ensine os conceitos da computação e posteriormente alguma forma de programação, a teoria da computação se torna de fundamental importância, pois ela permitirá que se tenha um grande embasamento teórico e um profundo entendimento sobre a ciência envolvida na computação (FERNANDES, 2002 p.13).

Este Trabalho visa a elaboração e aplicação de atividades que envolvam a prática de técnicas de programação para crianças e adolescentes na faixa etária de 08 a 13 anos e analisar os impactos que estes conhecimentos proporcionaram em seu desenvolvimento escolar.

Permitir que crianças compreendam melhor o funcionamento das tecnologias que estão presentes em seu dia a dia, encorajando-as a se aprofundar nas áreas relacionadas à Tecnologia da informação principalmente no desenvolvimento de softwares.

As ferramentas permitiram apresentar conceitos de lógica e programação voltados ao desenvolvimento e análise de software de maneira simples e divertida o que permitirá a elas se alfabetizarem tecnologicamente aprendendo em um primeiro momento as linguagens de programação básicas fornecidas pelas ferramentas utilizadas e conceitos de ciência da computação.

2 LINGUAGEM DE PROGRAMAÇÃO

Com o avanço das máquinas, tornou-se necessário também o avanço na maneira como seriam desenvolvidos os programas para os computadores, do cartão perfurado e da movimentação de válvulas houveram muitos avanços até chegar nas linguagens modernas que possuímos hoje.

Pode-se separar as linguagens de programação em duas categorias, linguagens de Alto e Baixo nível, o que define o nível da linguagem é a sua proximidade com a máquina a nível de gerenciamento de hardware, ou seja quanto mais próxima da linguagem binária das máquinas, mais baixo nível é a linguagem e quanto mais próximo das linguagens humanas, mais alto nível é a linguagem.

As linguagens de programação de alto nível são um meio que nos permite dizer as máquinas o que elas devem fazer e como elas devem fazer, sem que para isso tenhamos que escrever diretamente em linguagem de máquina, conforme Manssour (200?, p.4)

Linguagem de Alto Nível: é a LP mais próxima da linguagem do homem, não requer conhecimento da arquitetura da máquina e é portátil, isto é, independente da máquina. Neste caso, o programa ou código fonte precisa ser traduzido para linguagem de máquina para poder ser executado. O código ou programa objeto é o resultado da tradução quando é utilizado um compilador ou interpretador

Como exemplo de linguagem de alto nível podemos citar o Java cujo a sintaxe se aproxima muito a linguagem humana onde os comandos são dados de forma imperativa.

Contrariamente as linguagens de Alto nível, as linguagens de baixo nível se aproximam mais da linguagem de máquina, o que torna o seu entendimento mais difícil para as pessoas como cita Langa (2006, p.1)

Programa que se realiza com este tipo de linguagem não pode ser migrado ou utilizado em outras máquinas. Ao estar praticamente desenhado a medida do hardware, aproveitam ao máximo as características do mesmo.

A exemplo de linguagem de baixo nível pode-se citar o Assembly que trabalha diretamente com os registradores do processador executando operações diretamente na máquina.

As linguagens de programação deram uma maneira de se comunicar com as máquinas permitindo criar aplicações de forma mais ágil e amigável sem que para isso faça-se a programação direto em linguagem de máquina como era feito com os cartões perfurados.

2.1 JAVA

Java é uma linguagem que foi derivada do C++, financiada originalmente pela SUN Microsystems que mais tarde viria ser incorporada pela Oracle, que mantém e estimula o uso da linguagem, “O Java é uma poderosa linguagem de programação. Os programadores experientes as vezes sentem-se orgulhosos por criarem algum uso exótico, distorcido e complexo de uma linguagem” (DEITEL e DEITEL 2009, p.11).

Java é uma Linguagem de programação cujo seus arquivos compilados rodam sobre uma máquina virtual, permitindo que os códigos escritos para uma plataforma sejam executados em outra sem a necessidade do processo de *rebuild* ou de adaptação de código, segundo Romanato (200?)

Uma máquina virtual é um software que simula uma máquina física e consegue executar vários programas, gerenciar processos, memória e arquivos. Resumindo, ele constitui de uma plataforma, onde a memória, o processador e seus outros recursos, são totalmente virtuais, não dependendo de hardwares.

A JVM é a máquina virtual Java responsável pela execução dos códigos escritos e compilados, ela gerencia todos os recursos necessários, memória, processador, controle de threads acesso a disco entre outras, permitindo assim que a aplicação seja executada. Essa máquina é capaz apenas de executar *byte Codes* gerados pelo compilador Javac, mesclando trechos de códigos que são compilados e outros que serão apenas interpretados, dependendo da necessidade de permanece e repetições de instruções que ocorrem durante o código (FROTA, 2013).

Devido as suas características, a linguagem Java permite ao programador poupar tempo e esforços ao desenvolver suas aplicações, uma vez que ela abstrai o controle de hardware da aplicação e o coloca em uma máquina virtual permitindo assim a escrita e compilação de um mesmo código para diferentes sistemas operacionais, permitindo assim que a aplicação seja executada em qualquer ambiente bastando que para isso haja uma máquina virtual Java sendo executada no sistema em que deseja-se utilizar a aplicação.

2.2 SMALL BASIC

O Small Basic é uma linguagem de programação baseada na linguagem BASIC da Microsoft, contendo uma lista reduzida de palavras chaves (apenas 14), voltado para simplicidade o Small Basic não possui classes, objetos, genéricos ou lambdas, apenas o código imperativo (RAJI, 2008).

Segundo o guia do Small Basic disponibilizado pela Microsoft (200?, p.2)

Small Basic é uma linguagem de programação que é desenhada para tornar a programação extremamente fácil, acessível e divertida para iniciantes. A intenção do Small Basic é derrubar a barreira e servir como trampolim para o incrível mundo da programação.

As variáveis não são fortemente tipadas e sendo dinâmicas e globais todo o tempo, conforme explica a Microsoft (200?, p.12).

Variáveis têm nomes associados a elas e é assim que você as identifica. Existem algumas regras simples e diretrizes muito boas para nomear essas variáveis. Elas são:

O nome deve começar com uma letra e não deve colidir com nenhuma outra palavra-chave como por exemplo **if**, **for**, **then**, etc.

Um nome pode conter qualquer combinação de letras, dígitos e sublinhados (_).

É útil nomear variáveis de forma significativa – já que variáveis podem ser tão compridas quanto você desejar, use nomes de variáveis para descrever sua intenção.

O código gerado pelo Small Basic roda sob a plataforma DotNet, o que permite que o compilador e editor do Small Basic tenha acesso a todas as bibliotecas

disponíveis para o DotNet, bem como permite que novas bibliotecas sejam criadas a partir de qualquer linguagem suportada pela plataforma.

Devido a sua simplicidade nos comandos e ao seu número restrito de palavras chave, o Small Basic se mostra uma excelente opção para a introdução a programação, uma pequena desvantagem desta linguagem é que ela está disponível apenas para o ambiente Windows.

2.3 SCRATCH

O Scratch é uma linguagem de programação criada pelo Instituto de tecnologia de Massachusetts (MIT) em 2003 e desde 2013 o Scratch 2 está disponível como uma aplicação web para Windows, Mac OS X e Linux, o Scratch permite o aprendizado de programação através do conceito visual de blocos, permitindo o controle do fluxo de programação através de uma técnica de arraste e solte que envolve blocos de códigos, sons, imagens e outros tipos de mídia.

A usabilidade do Scratch é baseada nos softwares e mesas de DJs, pois o Scratch se refere aos pedaços de código que podem facilmente serem reutilizados, combinados e adaptados, a linguagem foi criada com o objetivo de ser simples, intuitiva e de fácil aprendizado, permitindo que desenvolvedores de qualquer idade e com qualquer grau de conhecimento em programação possam utilizar-se dela para criar projetos (LAMB e JOHNSON, 2011).

Segundo Ventorini e Fioreze (2014, p.4):

Este software possui uma sintaxe mais intuitiva que as outras linguagens de programação como Pascal, Python, Fortran, Cobol e C++. Seus blocos de comandos são visíveis e possuem maior diversidade de comandos prontos, representados por blocos, os quais facilitam a produção de estórias multimídias interativas ou qualquer outro tipo de programação.

O Scratch demonstra ser uma excelente ferramenta para o ensino de programação a crianças e adolescentes, pois sua interface simples, minimalista e intuitiva, diminui a curva de aprendizado e permite que se crie conteúdo de uma maneira mais lúdica, facilitando assim a introdução de conceitos de programação.

2.4 DESENVOLVIMENTO

Para o desenvolvimento e aplicação das atividades de programação, escolheu-se o Scratch, por possuir uma interface mais simples e amigável o que facilitou a sua utilização.

Foram desenvolvidas e aplicadas atividades que abordam as estruturas de decisão e repetição, após explicado o funcionamento das estruturas e do ambiente de desenvolvimento do Scratch, aplicou-se os conceitos na explicação e resolução de problemas matemáticos.

Ao término foi aplicado um questionário aos alunos para que eles avaliassem a dificuldade de utilização da linguagem, e se a sua utilização facilitou no aprendizado e na resolução dos problemas matemáticos, o mesmo questionário foi aplicado ao professor para que ele avaliasse a viabilidade de incorporar as técnicas de programação a prática do ensino.

2.4.1 Atividade 1: Introdução a ferramenta.

O Ambiente de programação do Scratch consiste em um palco onde pode ser visualizado as ações que os atores irão desempenhar,

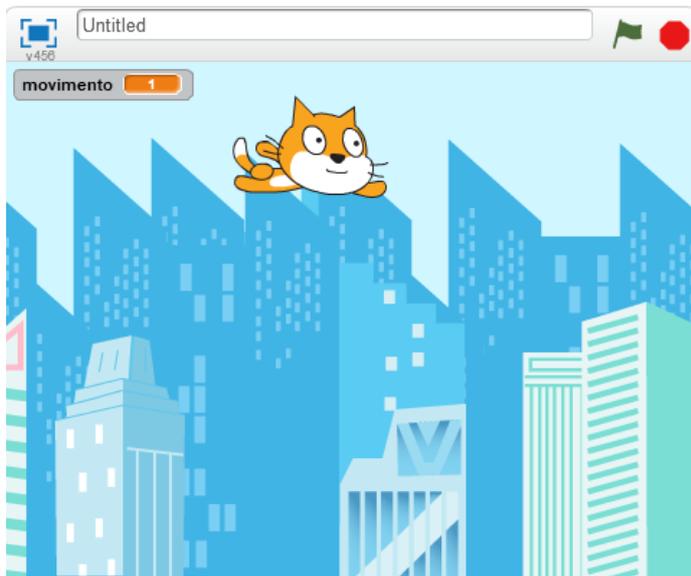


Figura 1: Palco
Fonte: O Autor

Um painel de atores onde é possível visualizar os atores que estão sendo utilizados no projeto,

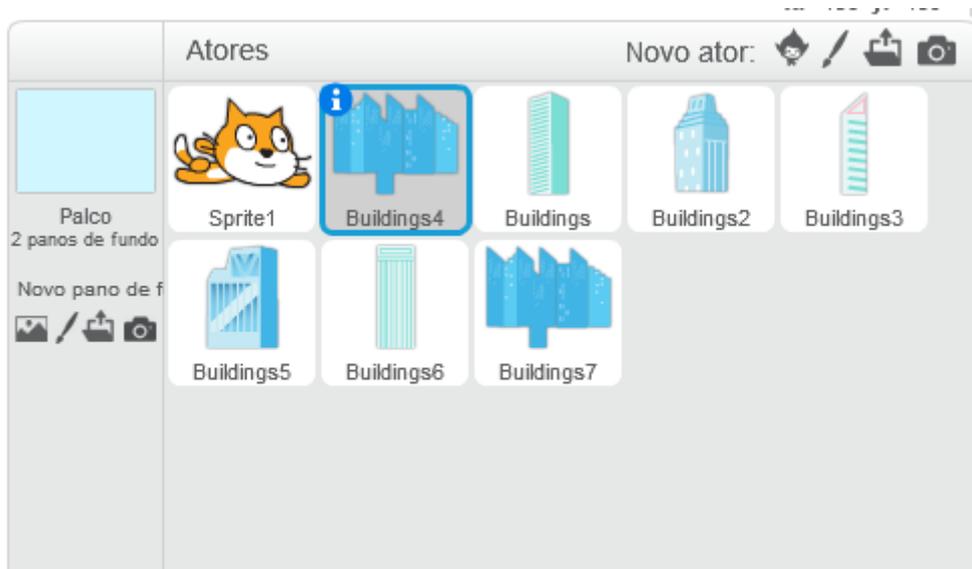


Figura 2: Painel de Atores
Fonte: O Autor

Uma área de ferramentas onde podem ser encontrados os blocos de programação, estruturas de repetições, ferramentas de controle, fantasias e sons que serão utilizados pelos personagens.

E uma área de programação onde os blocos são montados, esta área permite que seja feita a programação do jogo ou definida as ações das animações.

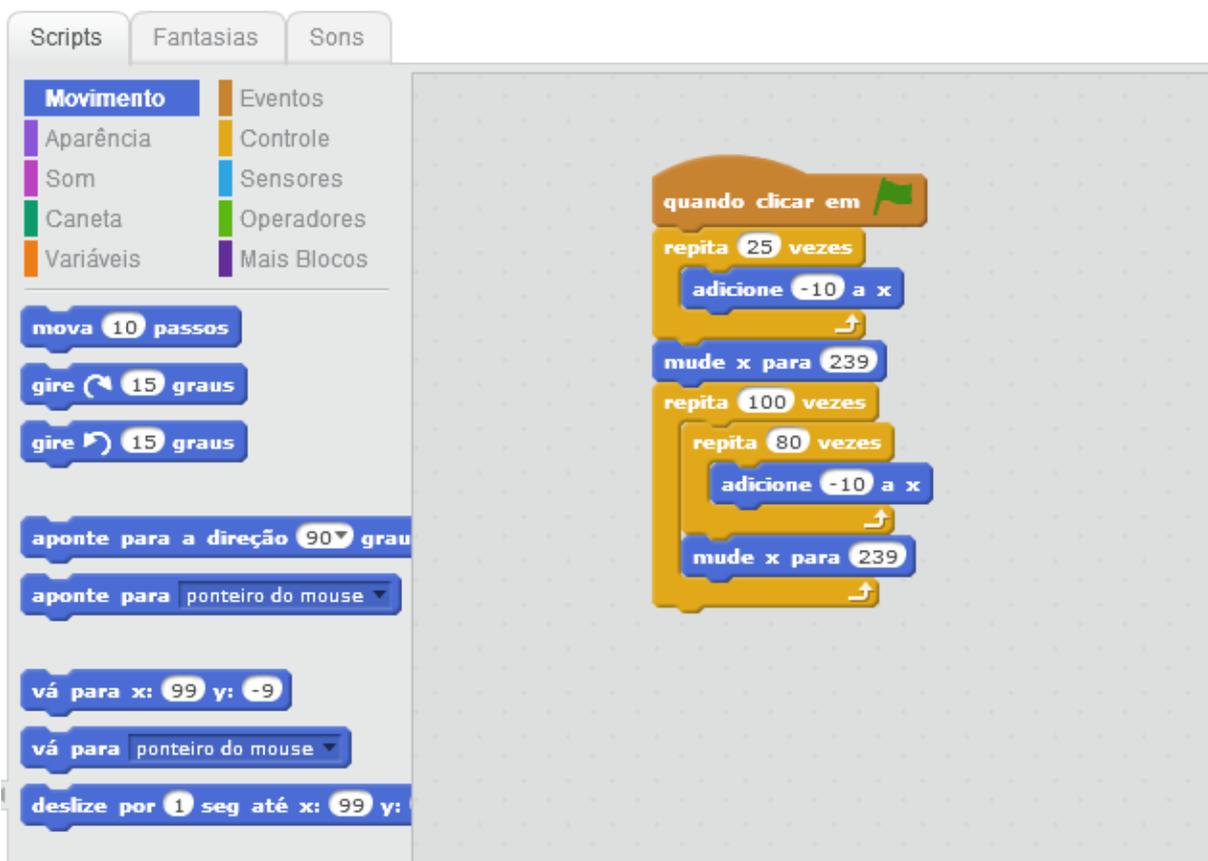


Figura 3: Painel de Scripts e blocos de programação.
Fonte: O Autor

Elaborou-se um exercício simples onde gera-se a ideia de voo, para isso foram utilizados alguns atores e conceitos simples de repetições.

Para o primeiro passo foi definir o plano de fundo do palco, para isso escolhemos o plano de fundo padrão blue sky 2.

Para gerar a ideia de movimento foi inserido alguns atores com formatos de prédios, e adicionado um evento a ele, para isso, é necessário selecionar a opção eventos no painel de scripts e selecionar o evento, **quando a tecla # for pressionada**, seleciona-se então este evento e o arrasta para o painel de programação, em seguida adicionar um **bloco adicione # a x** que fará com que o ator se mova para a direção desejada,

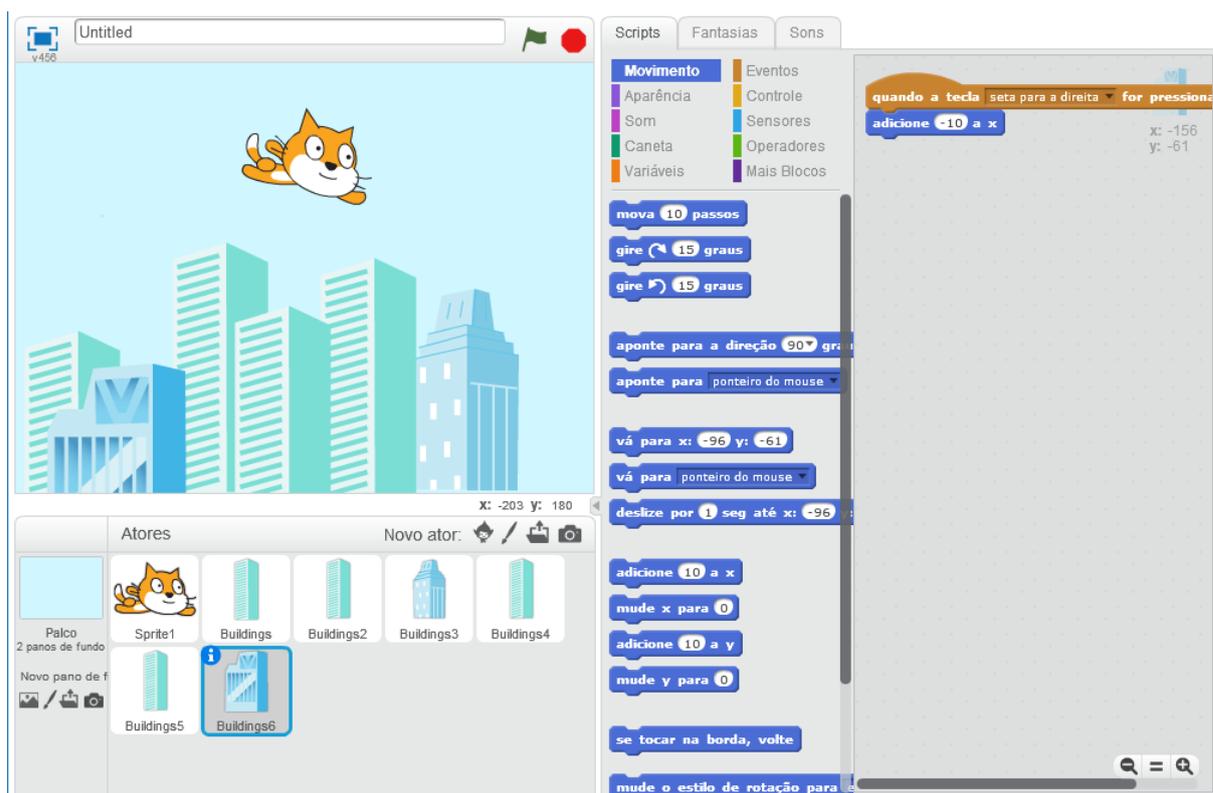


Figura 4: Exemplo de movimento
Fonte: O Autor.

Quando se adiciona um valor positivo o ator se move para a direita, ao adicionar um valor negativo, o ator se moverá para a esquerda.

2.4.1.1 Atividade 1: Estruturas de seleção

Ao deslocar o cenário um problema verifica-se o problema com a movimentação dos edifícios, eles não voltam para a o início do palco ao chegarem na extremidade, fazendo com que eles fiquem acumulados no canto do cenário, para corrigir este problema usaremos de estruturas de decisão.

As estruturas de decisão são trechos de códigos que nos permitem alterar o fluxo do programa dado uma determinada condição, elas normalmente se apresentam na forma de if, if/else, if/else if/else, se, se / senão e se/ senão se/ senão.

Para a correção de nosso problema utilizamos a seguinte condição, quando o ator chega ao fim do palco (eixo x = -240) enviamos o ator novamente para o início do palco,

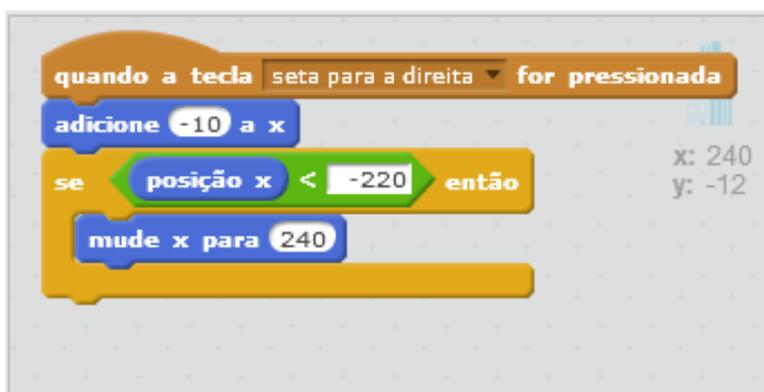


Figura 5: Estrutura de repetição "se"
Fonte: O Autor.

2.4.1.2 Atividade 1: Estrutura de repetição I

Para gerar saber a quanto tempo está rodando a atividade, podemos utilizar um contador para isso, utilizando-se de uma estrutura de repetição pode-se determinar há quanto tempo está rodando.

Estruturas de repetição, são blocos de códigos que executam um determinado trecho de programa de maneira repetitiva variando de nenhuma repetição até o infinito, neste último caso ocorre o que chamamos de loop infinito.

No geral há três estruturas de repetições, for, while e do... while (repita, enquanto, e faça...).

- For: A instrução for normalmente é acompanhada de uma variável de controle, que determina quando o laço de repetição será encerrado, um exemplo é: `for(int x =0; x =10; x++)`, neste caso a primeira parte define-

se que a variável x iniciará com valor 0, após tem-se a declaração $x=10$, nesta parte temos a condição de parada que é quando o valor da variável x for 10 e por fim temos a incrementação do valor $x++$, aqui a variável x é incrementada em mais um, nesta última parte do laço for podemos executar qualquer tipo de trecho de código, desde a incrementação da variável x até uma chamada de método.

- While(#): o laço while opera de maneira semelhante, porém ele conta apenas com a condição de parada, neste caso ele é executado enquanto a condição for verdadeira, o laço while é o mais propício a causar loopings infinitos, pois a sua condição pode ser configurada como verdadeira e não ser mais alterada. Um exemplo de laço while em Java:

```
Int x =1;
While(x <3){
    x++;
}
```

- Do ... While: o laço do...while funciona de maneira semelhante ao while, porém ele irá executar o seu bloco de código antes de testar a condição de repetição, neste caso garantimos que o código será executado ao menos uma vez: um exemplo de laço do...while em Java:

```
Int x = 1;
Do{
    X =X+5;
}while(x<6);
```

Neste exemplo podemos ver que o laço será executado uma única vez.

Utilizando-se de um laço while podemos incrementar o valor do cronometro.



Figura 6: Laço de repetição while
Fonte: O Autor

2.4.2 Atividade 2: Resolução de cálculos

O propósito desta atividade é criar uma calculadora que efetue as operações aritméticas básicas, soma, subtração, divisão e multiplicação.

Para isso utilizou-se os recursos que foram previamente explanados na atividade 1.

Com esta atividade buscou-se aprofundar o aprendizado em eventos e exemplificar como uma ação em um componente do sistema pode influenciar o todo.

Para início definiu-se o painel mostrador, onde os dígitos serão exibidos para o usuário, nesta etapa definiu-se com quantas casas a calculadora é capaz de operar, para este exemplo utilizou-se 9 casas.



Figura 7: Painel calculadora contendo 9 casas
Fonte: O autor.

Para exibição dos dígitos, utilizou-se de nove atores com fantasias que podem variar de zero a nove, que variam de acordo com o número selecionado.

Para armazenar os valores inseridos fez-se o uso de um vetor com nove posições onde cada uma de suas posições é exibida por uma casa no painel, desta forma trabalha-se com números individuais que são pré-processados concatenando os valores do vetor em um número que é armazenado em uma variável para então ser utilizado no cálculo.

O painel numérico utiliza um ator para cada número, quando o ator é clicado, ele adiciona um na primeira posição do vetor, o que garante a correta exibição dos números, ao término uma mensagem de refresh é enviada para os atores que compõem o painel mostrador, isso faz com que eles se sincronizem com os valores presentes no vetor.



Figura 8: Código de refresh do painel.
Fonte: O Autor



Figura 9: Painel Numérico
Fonte: O Autor



Figura 10: Código fonte utilizado no botão 1 do painel.
Fonte: O Autor

Os botões de operações somente adicionam um valor a variável de operação, este valor indicara qual operação o sistema deverá executar quando o botão de igual for clicado.



Figura 11: Código fonte operação de soma.
Fonte: O Autor

Os botões de operações também promovem o pré-processamento do vetor, onde concatena-se as suas posições e salvando-se o número resultante em uma variável, após o pré-processamento, limpa-se o vetor para receber o segundo número da operação.

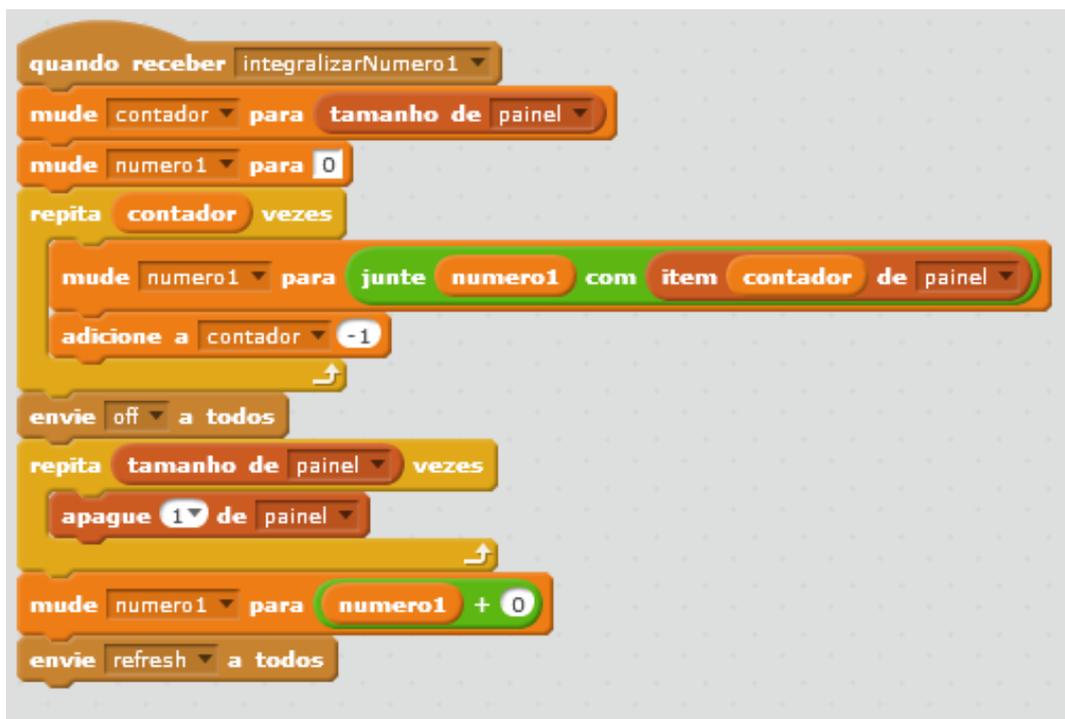


Figura 12: Concatenação do vetor
Fonte: O Autor

Pressionando-se o botão de igual é efetuada a operação desejada, o valor resultante da operação é fragmentado em algarismos independentes que são armazenados no vetor para então serem exibidos no painel da calculadora.

2.4.3 Questionário

Para análise dos resultados obtidos com as atividades ministradas, aplicou-se um questionário onde alunos e professores responderam algumas questões que fora organizada em dois formulários, o primeiro contendo questões gerais para nivelamento de conhecimento, buscou-se saber do contato dos jovens com o computador e atividades de programação, após o preenchimento do primeiro formulário, aplicou-se as atividades propostas e em seguida o segundo questionário foi aplicado.

No segundo formulário buscou-se identificar a experiência obtida com as atividades, se elas foram positivas e o quanto estas atividades incentivaram ou não o interesse dos jovens pela programação.

Questionário de Avaliação

questionário para avaliação das atividades desenvolvidas em sala de aula.

*Obrigatório

1. **Nome:** *

2. **Idade:** *

3. **Escola:** *

4. **Turma:** *

5. **Qual o seu interesse por informática ***

Marcar apenas uma oval.

- Nenhum, não me interessa.
- Pouco, sei que ela é importante para fazer o youtube funcionar, mas não me interessa como ele funciona.
- Normal, gostaria de saber como algumas coisas funcionam, tipo como que a foto fica salva no computador.
- Tenho interesse, inclusive fiz ou quero fazer cursos técnicos para aprender a mexer com computadores
- Tenho muito interesse, quero aprender como tudo funciona, quero ser um profissional de Tecnologia da Informação

6. **Você já teve contato com o computador? possui um em casa? ***

Marcar apenas uma oval.

- Não, não sei o que é
- Não, só vi de longe.
- Sim tenho contato com o computador porem não tenho em casa
- Sim tenho contato com o computador e possuo um.

Figura 13: Primeiro questionário pg 1.
Fonte: O Autor.

7. **Você já ouviu falar em programação, sabe o que é? ***

Marcar apenas uma oval.

- Não nunca ouvi falar
- Não, mas faço ideia do que seja.
- Mais ou menos
- Sim, já ouvi falar.
- Sim sei o que é e já programei

8. **Você já ouviu falar em estrutura de dados? ***

Marcar apenas uma oval.

- Não, é de comer com o que?
- Sim, mas não faço ideia do que seja.
- Sim, tenho noção do que é, porem não sei como usa.
- Sim sei o que é e como utilizar.

9. **Você gosta de jogos eletrônicos? ***

Marcar apenas uma oval.

- Não, odeio jogos.
- Não, acho sem graça
- Não, mas não tenho nada contra
- Sim eu gosto
- Adoro jogos.

10. **Com que frequência você na semana ***

Marcar apenas uma oval.

- De uma a duas vezes
- De três a quatro vezes
- De cinco a seis vezes
- Todos os dias

11. **Você gosta de saber como as coisas funcionam ***

Marcar apenas uma oval.

- Sim
- Não

12. **Você gostaria de aprender a fazer o seu próprio Jogo ***

Marcar apenas uma oval.

- Sim
- Não

Figura 14: Primeiro questionário pg 2.
Fonte: O Autor.

Questionário de Avaliação

questionário para avaliação das atividades desenvolvidas em sala de aula.

*Obrigatório

1. **Nome:** *

2. **Idade:** *

3. **Escola:** *

4. **Turma:** *

5. **Após as atividades desenvolvidas como você avalia as atividades de programação? ***

Marcar apenas uma oval.

- Não gostei, as atividades foram inúteis e não acrescentaram em nada nada para o desenvolvimento escolar
- Não gostei, as atividades forma boas porem não há interesse de minha parte.
- Gostei, as atividades foram legais, porém não acrescentaram em nada para o desenvolvimento escolar
- Gostei, as atividades foram boas e vejo que é possível utilizar os conceitos aprendidos no dia-a-dia, as atividades com certeza ajudaram no desenvolvimento escolar.

6. **Com relação a esta atividade (Programação) qual o seu interesse em desempenha-la profissionalmente ***

Marcar apenas uma oval.

- Não há interesse.
- Talvez como uma segunda opção.
- Pode ser, mas não por mais de 6 anos
- Possivelmente, não me incomodaria de programar.
- Com certeza, é isso que eu desejo para minha vida profissional.

Figura 15: Segundo questionário pg 1.

Fonte: O Autor.

7. **Com qual matéria você acredita que seria melhor de trabalhar utilizando o Scratch e a programação como ferramenta de aprendizagem? ***

Marcar apenas uma oval.

- Matemática
- Português
- Geografia
- História
- Artes
- Química
- Física
- Outro: _____

8. **Considerando as atividades desenvolvidas, você voltaria a utilizar o Scratch ou outro meio para programar? ***

Marcar apenas uma oval.

- Sim
- Não

9. **Considerando as atividades de programação, você gostaria de ter uma matéria que abordasse somente este assunto? ***

Marcar apenas uma oval.

- Sim
- Não

Figura 16: Segundo questionário pg 2.
Fonte: O Autor.

2.4.4 Resultados

As atividades foram realizadas com alunos de 7º e 6º ano totalizando 38 alunos com uma média de idade de 12 anos, após a aplicação das atividades e questionários, obteve-se os seguintes resultados.

Em relação a distribuição das Idades verificou-se que 74% dos alunos participantes possuem idade entre 11 e 12 anos. Figura17.

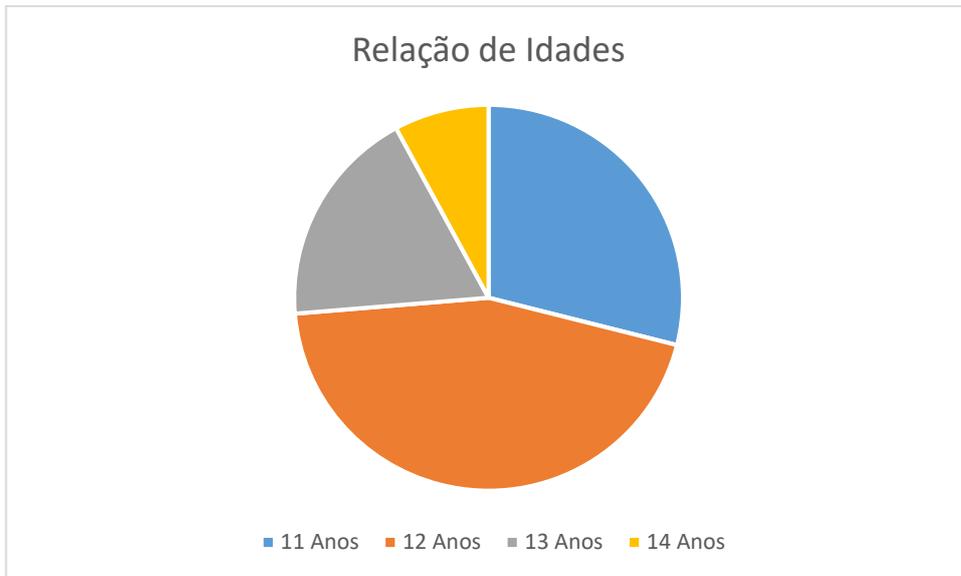


Figura 17: Distribuição de idades.
Fonte: O Autor.

Distribuição dos Alunos em relação as turmas, verificou-se que a maioria dos alunos estão cursando o sexto ano do ensino fundamental como observa-se na figura 18.

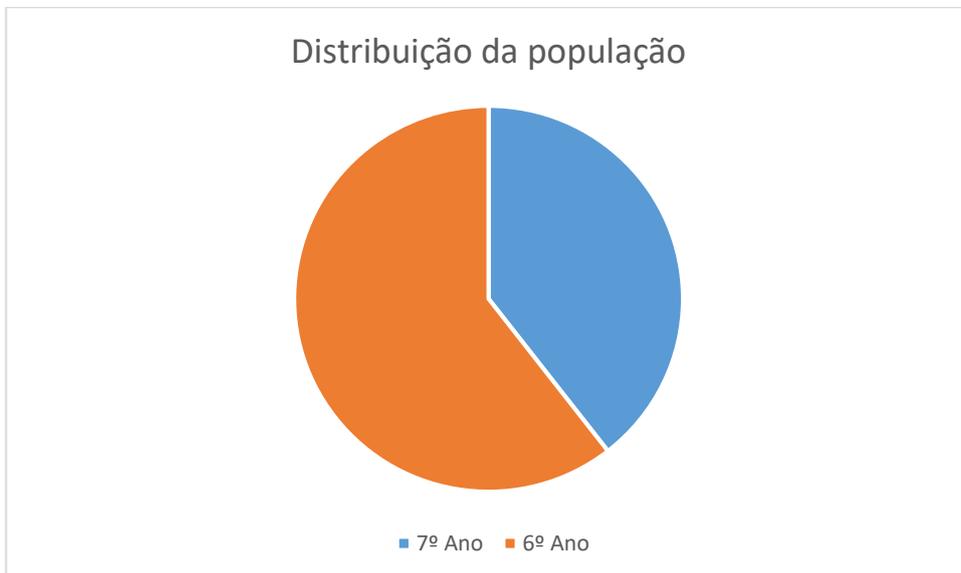


Figura 18: Distribuição de Turmas.
Fonte: O Autor.

Com base nas respostas dos questionários pode-se afirmar que há um grande interesse no aprendizado de computação e programação (Figura 19) e que há uma

alta inserção dos computadores na vida dos alunos de ensino fundamental (Figura 20).

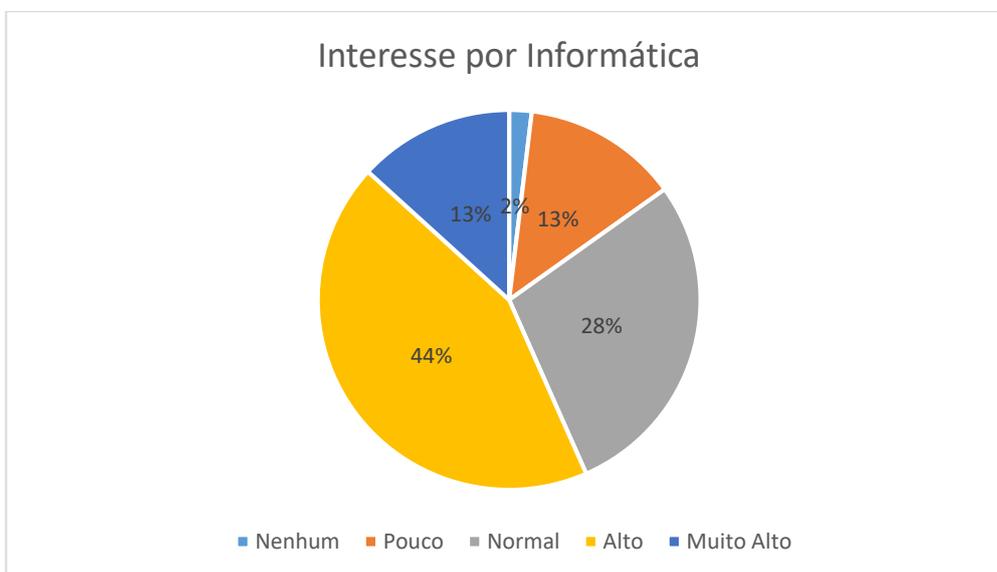


Figura 19: Totalização do interesse dos alunos por informática.
Fonte: O Autor.

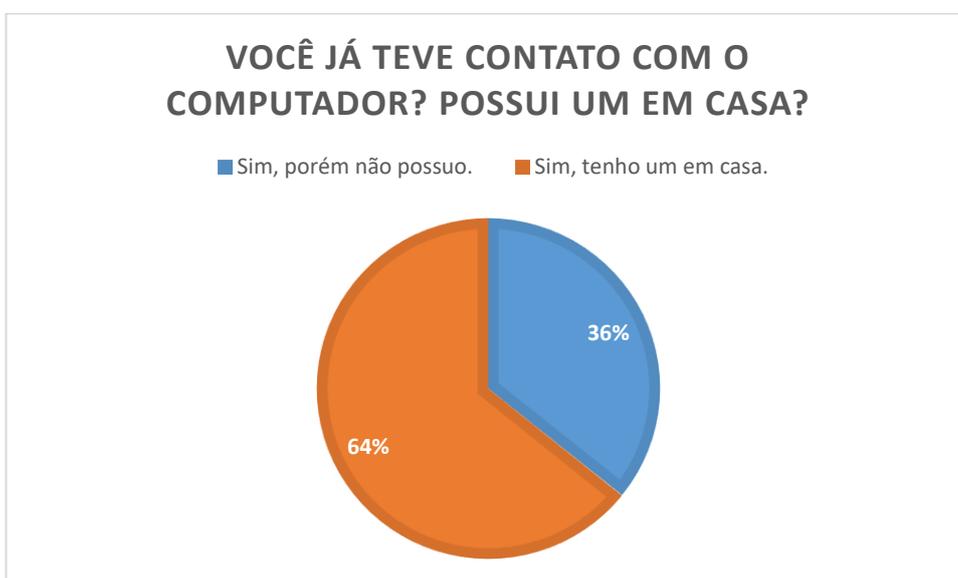
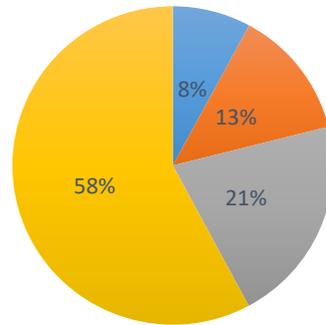


Figura 20: Contato com computadores.
Fonte: O Autor.

Levando-se em conta os resultados das questões aplicadas, verificou-se que há um grande interesse por parte dos alunos em aprender a programar e possivelmente desempenhar esta atividade profissionalmente, como pode ser observado nas figuras 21, 22 e 23.

Após as atividades desenvolvidas como você avalia as atividades de programação?



- Não gostei
- Não gostei, não tenho interesse
- Gostei, porém não vejo como isso pode me ajudar na escola.
- Gostei, as atividades foram boas, e me ajudaram na escola

Figura 21: Avaliação das atividades.
Fonte: O Autor.

COM RELAÇÃO A ESTA ATIVIDADE (PROGRAMAÇÃO) QUAL O SEU INTERESSE EM DESEMPENHA-LA PROFISSIONALMENTE

- Não há Interesse
- Talvez
- Pode ser
- Possivelmente
- Com certeza

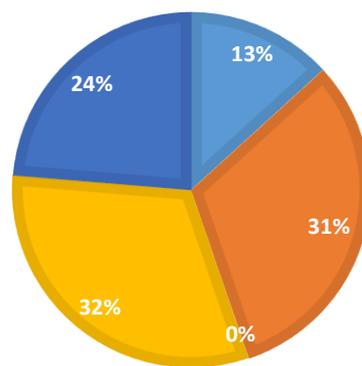


Figura 22: Interesse profissional em programação.
Fonte: O Autor.

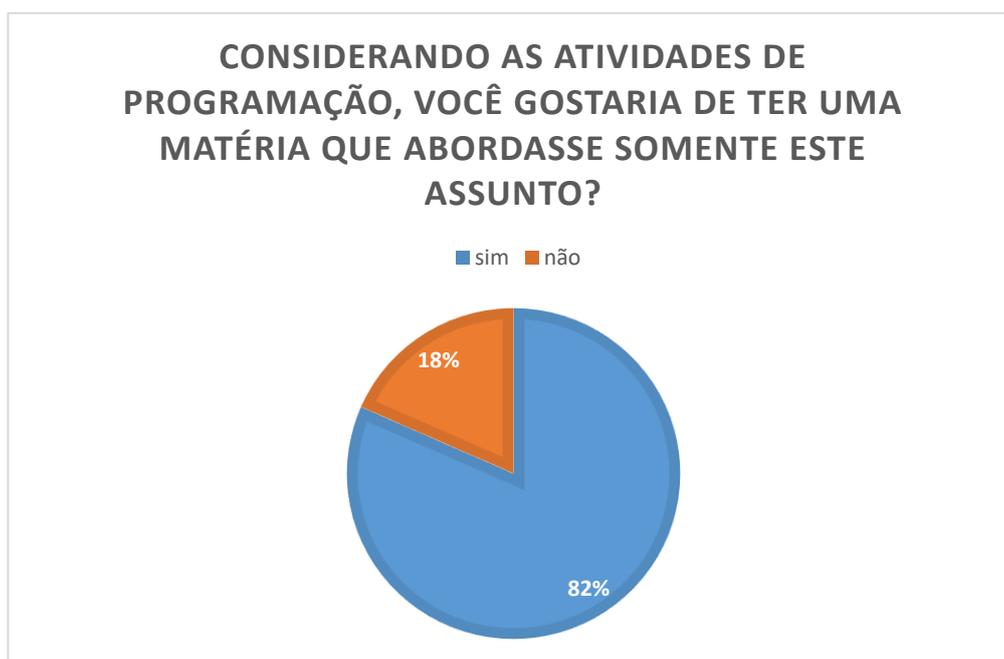


Figura 23: Interesse por aprender programação.
Fonte: O Autor.

3 CONCLUSÃO

Com base na realização deste estudo observou-se que durante a aplicação das atividades todos os alunos demonstraram interesse embora nem todos tenham demonstrado a mesma dedicação e concentração durante as atividades.

Os resultados dos questionários apontaram que das crianças entrevistadas, todas possuem um computador em casa e toda tem interesse em informática, destas algumas sabiam o que é programação e nenhuma conhecia previamente o conceito de estruturas de dados.

Ao término foi percebido um grande interesse pela possibilidade de desenvolver seus próprios jogos e a grande maioria gostaria de ter mais contato com a programação mesmo que em outras matérias na forma de atividades.

REFERÊNCIAS

DEITEL, Paul; DEITEL, Harvey. **Java como programar**. 8. ed. São Paulo: Pearson, 2009. 114p.

FERNANDES, Cláudia Santos. **Ciência da Computação para Crianças**. 2002. 165 f. Dissertação (Mestrado em Ciência da Computação) -Universidade Federal do Rio Grande do Sul, Porto Alegre, 2002.

FROTA, Handerson. **Introdução a JVM – Java Virtual Machine**. Handerson Frota, 2013. Disponível em: <<http://www.handersonfrota.com.br/introducao-a-jvm-java-virtual-machine/>>. Acessado em: 30 mai. 2016.

LANGA, Sara Alvarez. **Tipos de linguagens de programação**. Criarweb, 2006. Disponível em: <<http://www.criarweb.com/artigos/685.php>>. Acessado em: 30 mai. 2016.

LAMB, Annette; JOHNSON, Larry. **Scratch: Computer Programming for 21st Century Learners**. 2011. Disponível em:< <https://www.highbeam.com/doc/1G1-259378433.html>>. Acesso em 30 mai. 2016.

MANSSOUR, Isabel Harb. **Paradigmas de Linguagens I**. Scrib 200?. Disponível em: <<http://www.scribd.com/doc/267024067/paradigmas-aula01#scribd>>. Acessado em: 30 mai. 2016.

MICROSOFT, **Microsoft Small Basic, Uma Introdução à programação**, Microsoft, 200?, Disponível em:< http://download.microsoft.com/download/A/9/8/A9821DB3-A795-4059-A248-0705BA748C97/Introducao_ao_Small_Basic.pdf>. Acesso em: 30 mai. 2016.

RAJI, Vijaye. **Hello World**. Vijaye Raji, 2008. Disponível em: < <https://blogs.msdn.microsoft.com/smallbasic/2008/10/23/hello-world/>>. Acessado em: 30 mai. 2016.

ROMANATO, Allan. **Entenda como funciona a Java Virtual Machine (JVM)**. Devmedia 200?. Disponível em <<http://www.devmedia.com.br/entenda-como-funciona-a-java-virtual-machine-jvm/27624>>. Acessado em: 30 mai. 2016.

VENTORINI, André Eduardo; FIOREZE, Leandra Anversa. **O Software Scratch: Uma Contribuição Para O Ensino E A Aprendizagem Da Matemática**. 2014. 14f. Artigo - Universidade Federal de Santa Maria, Rio Grande do Sul, Santa Maria, 2014.