

UNIVERSIDADE FEDERAL DE SANTA CATARINA
CAMPUS ARARANGUA
TECNOLOGIAS DA INFORMAÇÃO E COMUNICAÇÃO

AMANDA IRIZAGA LUCRECIO

**COMPARAÇÃO E APLICAÇÃO DE DIFERENTES FERRAMENTAS
PARA ENSINO DE PROGRAMAÇÃO PARA CRIANÇAS**

Araranguá, 07 de dezembro de 2016

AMANDA IRIZAGA LUCRECIO

COMPARAÇÃO E APLICAÇÃO DE DIFERENTES FERRAMENTAS PARA ENSINO DE
PROGRAMAÇÃO PARA CRIANÇAS

Trabalho de Conclusão de Curso submetido à Universidade Federal de Santa Catarina - UFSC como parte dos requisitos necessários para a obtenção do Grau de Bacharel em Tecnologias da Informação e Comunicação. Sob a orientação da Professora Eliane Pozzebon.

Araranguá, 2016

Amanda Irizaga Lurecio

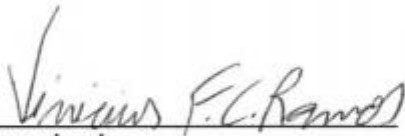
Comparação e aplicação de diferentes ferramentas para ensino de programação para crianças

Trabalho de Curso submetido à Universidade Federal de Santa Catarina - UFSC, como parte dos requisitos necessários para a obtenção do Grau de Bacharel em Tecnologias da Informação e Comunicação.




Orientador

Prof.ª D.ra Eliane Pozzebon



Examinador

Prof. D.r Vinicius Faria Culmant Ramos



Examinador

Prof. D.r Giovanni Lunardi

Araranguá, 7 de Dezembro de 2016

Ficha de identificação da obra elaborada pelo autor,
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Lucrécio, Amanda Irizaga
Comparação e Aplicação de Diferentes Ferramentas para
Ensino de Programação para Crianças / Amanda Irizaga
Lucrécio ; orientadora, Eliane Pozzebon - Araranguá, SC,
2016.
93 p.

Trabalho de Conclusão de Curso (graduação) -
Universidade Federal de Santa Catarina, Campus Araranguá.
Graduação em Tecnologias da Informação e Comunicação.

Inclui referências

1. Tecnologias da Informação e Comunicação. 2.
Programação. 3. Ferramentas. 4. Ensino. 5. Crianças. I.
Pozzebon, Eliane . II. Universidade Federal de Santa
Catarina. Graduação em Tecnologias da Informação e
Comunicação. III. Título.

AGRADECIMENTOS

Agradeço primeiramente à minha família, que sempre me incentivou a estudar e me deu apoio sempre que precisei. Ao meu namorado, Guilherme, que esteve comigo em todos os momentos críticos durante os semestres, por muitas vezes acreditando mais em mim, do que eu mesma. Ao meu amigo Felipe Canever, que me ajudou incontáveis vezes no decorrer do curso, e a todos que de alguma forma foram responsáveis por eu ter chegado até aqui.

RESUMO

No presente trabalho, realizou-se a aplicação de duas ferramentas de ensino de programação, Scratch e VisuAlg, em duas turmas compostas majoritariamente por alunos do 8º e 9º ano do ensino fundamental de uma escola pública na cidade de Araranguá. A programação de computadores é uma disciplina presente em muitos cursos superiores de caráter computacional, é também trabalhada com crianças e adolescentes, por sua característica em incentivar o raciocínio lógico e a resolução de problemas, o que traz benefício ao desempenho escolar, em disciplinas como português e matemática, em razão disto, a problemática deste trabalho é verificar qual ferramenta é a mais eficaz para o ensino de programação. Programar computadores pode ser uma tarefa difícil para diferentes níveis de educação, mesmo no ensino superior, a disciplina possui um alto nível de reprovação, portanto, uma das motivações deste trabalho é encontrar um método que facilite a aprendizagem dos conceitos básicos, desmistificando que programação é algo impossível, bem como analisar e comparar as diferentes ferramentas existentes que servem de auxílio para o ensino. Para corroborar com a proposta do trabalho, um questionário foi desenvolvido contendo questões sobre todo o conteúdo explorado ao longo das atividades, além da observação direta dos alunos no que se refere a resolução de exercícios em sala de aula. Após analisar os resultados obtidos, é possível afirmar que a ferramenta Scratch é a mais adequada para o aprendizado dos conceitos básicos e iniciais de programação, além disto, a opinião geral dos alunos foi positiva, levando em consideração as respostas da questão aberta sobre as atividades como um todo, atingindo então, o objetivo inicial do trabalho.

Palavras-chave: Programação de computadores, Ferramentas de programação, Alunos, Ensino Fundamental, Scratch, VisuAlg.

ABSTRACT

In this study, two programming teaching tools were implemented, Scratch and VisuAlg in two groups composed mainly of students from the 8th and 9th grade of elementary education in a public school in the city of Ararangua. Computer programming is a discipline that is present in many higher courses of a computational nature. It is also worked with children and adolescents, because of its characteristic in encouraging logical reasoning and problem solving, which brings benefits to school performance in disciplines such as Portuguese and mathematics. The problem of this work is to check which tool is the most effective for teaching programming. Program computers can be a difficult task for different levels of education, even in higher education, the discipline has a high failure thus one of the motivations of this work is to find a method that facilitates the learning of the basic concepts, demystifying that programming is impossible, and analyze and compare the different tools that serve as aid for education. To support the proposed work, a questionnaire was developed with questions about all content explored over the activities, in addition to direct observation of students in regard to solving exercises in the classroom. After analyzing the results, we can say that the Scratch tool is the most suitable for learning the basic and initial programming concepts, in addition, the general opinion of the students was positive, taking into account the answers to the open question about the activities as a whole, reaching then the initial goal of the work.

Keywords: Computer Programming, Programming Tools, Students, Basic Education, Scratch, VisuAlg

LISTA DE ILUSTRAÇÕES

Figura 1: Interface da Ferramenta LOGO. Extraída de: (captura de tela)	38
Figura 2: Interface da Ferramenta Code Combat. Extraída de: (captura de tela)	39
Figura 3: Interface da Ferramenta Code Studio. Extraída de: (captura de tela)	40
Figura 4: Interface da Ferramenta Scratch. Extraída de: (captura de tela)	41
Figura 5: Interface da Ferramenta VisuAlg. Extraída de: (captura de tela)	43
Figura 6: Interface da Ferramenta Code Monster. Extraída de: (captura de tela)	44
Figura 7: Interface da Ferramenta The Foos. Extraída de: (captura de tela)	45
Figura 8: Interface da Ferramenta YouCode. Extraída de: (captura de tela)	46
Figura 9: Aplicação da atividade 1. Extraída de: (do autor)	56
Figura 10: Aplicação da atividade 2. Extraída de: (do autor)	57
Figura 11: Gráfico da questão 2. Extraída de: (do autor)	61
Figura 12: Gráfico da questão 4. Extraída de: (do autor)	63
Figura 13: Gráfico da questão 5. Extraída de: (do autor)	64
Figura 14: Gráfico da questão 8. Extraída de: (do autor)	67
Figura 15: Gráfico da questão 10. Extraída de: (do autor)	69
Figura 16: Gráfico da questão 15. Extraída de: (do autor)	73
Figura 17: Gráfico da questão 16. Extraída de: (do autor)	74

LISTA DE TABELAS

Tabela 1: Comparação entre as escolas. Extraída de: (do autor)	26
Tabela 2: Comparação entre sites e plataformas online. Extraída de: (do autor)	29
Tabela 3: Comparação entre projetos. Extraída de: (do autor)	33
Tabela 4: Ferramentas com programação em bloco. Extraída de: (do autor)	46
Tabela 5: Ferramentas com linguagem aplicada a jogos. Extraída de: (do autor)	47
Tabela 6: Ferramentas com interpretadores de código. Extraída de: (do autor)	48
Tabela 7: Comparação entre Scratch e VisuAlg. Extraída de: (do autor)	77

SUMÁRIO

1. INTRODUÇÃO	13
1.1 PROBLEMÁTICA	15
1.2 OBJETIVOS.....	15
1.2.1 OBJETIVOS GERAIS	15
1.2.2 OBJETIVOS ESPECIFICOS	15
1.3 JUSTIFICATIVA	16
1.4 METODOLOGIA	16
1.5 ORGANIZAÇÃO DO TRABALHO	17
2. ENSINO DA PROGRAMAÇÃO.....	19
2.1 HISTÓRICO.....	19
2.2 A IMPORTÂNCIA DE APRENDER A PROGRAMAR	21
2.3 INICIATIVAS NACIONAIS E MUNDIAIS VOLTADAS A PROGRAMAÇÃO	22
2.3.1 ESCOLAS.....	22
2.3.2 SITES E PLATAFORMAS ONLINE.....	27
2.3.3 PROJETOS.....	31
3. FERRAMENTAS PARA O ENSINO DE PROGRAMAÇÃO	36
4. PROPOSTA DE ATIVIDADES	49
4.1 PLANO DE ATIVIDADES	49
4.1.1 ATIVIDADE 1: CONCEITOS BÁSICOS COMPUTACIONAIS	50
4.1.2 ATIVIDADE 2: ESTRUTURAS DE SELEÇÃO E OPERADORES	51
4.1.3 ATIVIDADE 3: ESTRUTURAS DE REPETIÇÃO	53
4.2 APLICAÇÃO DAS ATIVIDADES PROPOSTAS.....	54
5. RESULTADOS	59
5.1 QUESTIONÁRIO	59
6. TRABALHOS RELACIONADOS	79
7. CONSIDERAÇÕES FINAIS	81
8. REFERENCIAS.....	83
9. APÊNDICE.....	91

1. INTRODUÇÃO

A tecnologia educacional dentre várias definições, pode ser entendida como “o conjunto de processos, métodos e técnicas para enfrentar os problemas das práticas educativas e para favorecer a dinâmica da aprendizagem.” (GUÉDEZ,1982, p.19 apud CROCHIK, 1998)

Na escola, os alunos já estão inseridos em um ambiente tecnológico, pois, o quadro negro, assim como o giz, mesmo que ultrapassados, também são tecnologias. Porém, mais do que isto, celulares e computadores fazem parte do dia a dia destes alunos, e muitos deles inclusive, trazem estas ferramentas para dentro das salas, o que pode ser um infortúnio para os professores. Felizmente, alguns docentes buscam transformar esta adversidade em oportunidade, fazendo o uso destas tecnologias aliado ao ensino e aprendizagem. Como a utilização de *tablets* dentro da sala de aula, estes que foram entregues gratuitamente a diversas escolas de ensino público pelo Ministério da Educação através do Programa Nacional de Tecnologia Educacional -ProInfo (PORTAL MEC,2016).

Os dispositivos móveis como os celulares e *tablets*, são poderosas ferramentas para complementar o ensino regular nas escolas, pois são como “computadores de bolso”, isto é, se a escola possui internet, os alunos podem utilizar ferramentas de pesquisa como o *Google* sem ter que sair da sala de aula, ou então, fazer uso de aplicativos educacionais previamente instalados nos dispositivos, para que atividades sem uso de internet possam ser uma opção. Os *tablets* servem como ferramenta de apoio pedagógico, não sendo necessária a substituição dos livros, os alunos leem sobre tecnologia, mas o passo mais importante é justamente utilizar desta tecnologia para incrementar o conhecimento dentro da sala de aula.

E toda esta expansão tecnológica só é possível pois, mais países estão buscando a inclusão digital. Inclusão digital segundo Ribeiro (2016) se refere “ao acesso à informação que está nos meios digitais e, como ponto de chegada à assimilação da informação e sua reelaboração em novo conhecimento, tendo como consequência desejável a melhoria da qualidade de

vida das pessoas”, isto é, não basta fornecer o computador e a internet, é necessário que o usuário aprenda algo, e consiga aplicar este aprendizado no seu dia a dia. Se isto não for possível, então não temos inclusão digital.

A escola tem um papel importante no que diz respeito a inclusão digital, pois muitos alunos só têm acesso à internet por conta dos laboratórios existentes na mesma. Mas existe uma questão eminente no meio acadêmico, que é sobre o que a internet vem proporcionando aos alunos, bem como o conteúdo acessado por eles, e se de fato, está ocorrendo o que chamamos de inclusão digital.

A quantidade de adeptos tecnológicos é diretamente proporcional a quantidade de tecnologias existentes, e só cresce. Em contrapartida, enquanto o número de adeptos aumenta, a faixa etária dos mesmos diminui. Isto ocorre, pois, crianças e adolescentes estão cada vez mais cedo, tendo contato com a tecnologia, e isto pode ser algo bom e proveitoso, pois, desta forma, a criança terá mais facilidade em lidar com as tecnologias apresentadas no decorrer da sua vida, pois, mesmo que a tecnologia mude, o impacto inicial já foi quebrado. Considerando isto, projetos relacionados a programação de computadores vem ganhando força, pois reúnem tecnologia e ensino, além de todo o aparato característico da programação, como raciocínio lógico, incentivo a criatividade e a resolução de problemas.

Para Stroustrup (2014), “A programação é a arte de expressar soluções para problemas de modo que o computador possa executá-las”. Programar nada mais é do que escrever uma sequência de comandos para que o computador interprete. A programação, por se tratar de sequências lógicas de comando, pode facilitar o aprendizado em outras áreas, como a matemática e física, que exigem o raciocínio lógico para a resolução de problemas. Incentivar crianças e adolescentes a programar, é mostrar uma nova perspectiva em relação ao mundo e suas coisas.

Iniciativas como “Hour of Code” no Estados Unidos e “Code Club” que tem a extensão do projeto no Brasil, proporcionam a crianças e adolescentes momentos de lazer aliado ao saber. O fundador do “Happy Code” na Estônia, Rodrigo Santos, acredita que é fundamental que crianças e adolescentes aprendam a programar. Segundo Dantas e Costa (2013) programar explora “a habilidade de raciocínio lógico, conhecimentos matemáticos, trabalho em equipe, capacidade de resolver problemas e o estímulo da criatividade”.

Para participar destas iniciativas, as escolas interessadas devem se cadastrar, e ao fazerem isto, tem um papel muito importante no que diz respeito ao incentivo aos alunos, para que

os mesmos continuem até o fim. Este incentivo pode surgir em forma de competição, resolução de problemas, dinâmicas, entre outros, de qualquer forma, é essencial que os professores se comuniquem com os alunos para buscar a melhor forma de ensino/ aprendizagem.

Mas se há tantos pontos positivos ao ensinar programação, porque ainda não é uma prática mais comum? Muitas podem ser as justificativas, Gomes et al. (2008) acredita que “Os métodos de ensino tradicionalmente utilizados não parecem adequados às necessidades da maioria dos alunos”.

Um dos desafios em ensinar programação é entendê-la. Se tratando de raciocínio lógico, muitas pessoas podem apresentar dificuldade nesta área, transformando-a em um impasse. Além disto, há um certo desconhecimento no conceito de programar, o que dificulta o interesse tanto dos professores, quanto dos alunos, pois os mesmos acreditam que este conhecimento adquirido pode não servir para algo no futuro.

1.1 PROBLEMÁTICA

Se há tantas ferramentas para ensinar programação, porque ainda não é uma prática mais comum? Qual o método mais eficaz para ensinar programação?

1.2 OBJETIVOS

1.2.1 OBJETIVOS GERAIS

O objetivo deste trabalho é avaliar duas ferramentas para o ensino de programação e buscar identificar qual obteve o resultado mais eficaz.

1.2.2 OBJETIVOS ESPECIFICOS

- Pesquisar sobre o ensino e aprendizagem de programação.
- Efetuar pesquisa sobre ferramentas disponíveis para o ensino de programação.

- Escolher as ferramentas e definir planos de atividades para cada uma.
- Escolher uma escola do ensino público para aplicar as ferramentas.
- Aplicar o questionário de avaliação nas turmas.
- Analisar e discutir os dados obtidos.

1.3 JUSTIFICATIVA

Atualmente, muitas crianças continuam tendo dificuldade na resolução de problemas, é o que indica a Avaliação Nacional de Alfabetização feita em 2015, onde os alunos foram divididos em níveis de acordo com seu conhecimento e habilidade. No Brasil, 1 em cada 4 alunos está no nível mais baixo, para o ministro Renato Janine Ribeiro, isso se explica porque “muitas vezes a dificuldade do aluno é também em português e para interpretar os problemas”. (G1, 2015)

Esta dificuldade em interpretar problemas é também um dos obstáculos que o ensino de programação busca superar. Levando em consideração que um programa é uma sequência de comandos para a resolução de um problema, se o aluno encontra dificuldade em matemática, provavelmente encontrará em programação. Portanto, se elas estão relacionadas desta forma, uma pode servir de ajuda para a outra.

Um reflexo desta adversidade, é a falta de mão de obra especializada na área de TI, anualmente, há uma crescente oferta na área, porém, muitas vagas continuam vazias pois falta conhecimento. Segundo o Índice Brasscom de Convergência Digital, “ a evasão nos cursos da área de TI, em 2010 foi de 87% em todo o Brasil” (BRASSCOM,2012).

Por esta razão, este trabalho pretende analisar diferentes ferramentas para ensino de programação e realizar a aplicação de algumas delas para verificar qual destas é a mais adequada para se aprender a programar.

1.4 METODOLOGIA

Este trabalho é uma pesquisa quantitativa descritiva, embasada em tecnologia que visa a aplicação de diferentes ferramentas educacionais para o ensino da programação em turmas de uma escola do ensino público, para avaliar qual ferramenta é a mais eficaz. A metodologia de desenvolvimento deste trabalho foi dividida em sete etapas:

Etapa 1: Análise da literatura com foco na área de programação, bem como o ensino de programação para crianças e adolescentes.

Etapa 2: Identificação das ferramentas utilizadas para o ensino de programação e escolha das ferramentas a serem aplicadas.

Etapa 3: Desenvolvimento de um plano de atividades para as 2 ferramentas escolhidas, com uma descrição detalhada e o objetivo que deseja alcançar com cada atividade.

Etapa 4: Elaboração do questionário de avaliação a ser aplicado nas turmas do 9º ano após a realização das atividades propostas.

Etapa 5: Aplicação das ferramentas de acordo com o plano de atividades estabelecido.

Etapa 6: Aplicação do questionário para os estudantes que participaram das atividades;

Etapa 7: Avaliação dos questionários previamente respondidos, bem como a avaliação durante a aplicação das ferramentas.

1.5 ORGANIZAÇÃO DO TRABALHO

No decorrer deste trabalho, foram desenvolvidos 9 capítulos, sendo os 5 primeiros, essenciais para a conclusão do mesmo. A seguir, um demonstrativo de como a estrutura do trabalho foi desenvolvida:

Capítulo 1: Neste capítulo, o tema do trabalho é descrito de forma introdutória, contendo os principais tópicos como objetivos, justificativa e metodologia utilizada.

Capítulo 2: No segundo capítulo, um breve histórico sobre a programação de computadores é narrado, bem como a importância de se aprender a programar, e os diversos projetos e iniciativas responsáveis pelo ensino de programação no Brasil e no mundo.

Capítulo 3: Neste capítulo, são realizadas algumas comparações referentes as ferramentas existentes para o ensino de programação.

Capítulo 4: Todo plano de atividades realizado é apresentado, assim como um detalhamento das atividades desenvolvidas.

Capítulo 5: Através da análise do questionário, é feita uma discussão sobre os resultados obtidos com as atividades.

Capítulo 6: Alguns trabalhos relacionados são citados, deixando em ênfase suas semelhanças e diferenças quanto ao presente trabalho.

Capítulo 7: Apresentação das considerações finais que sejam relevantes para o trabalho como um todo.

Capítulo 8: Referencial teórico utilizado para desenvolver o trabalho.

Capítulo 9: Os apêndices são apresentados, a carta convite feita a diretoria da escola e o termo de autorização dos alunos para o uso da imagem.

2 ENSINO DA PROGRAMAÇÃO

Aprender a programar é fundamental para se obter um pensamento computacional das tecnologias ao seu redor, bem como aprimorar o raciocínio lógico. No entanto, muitos fatores podem contribuir negativamente na aprendizagem de programação, portanto, os professores devem estimular os alunos, através de atividades diferenciadas, pois deste modo, “muito dos problemas alegados como indisciplina ou desinteresse deixariam de existir, em virtude do prazer proporcionado aos alunos pelo entendimento do assunto trabalhado” afirmam Pezzini e Szymanski (2016).

Diversos projetos tem garantido que muitas pessoas considerem a programação como uma opção de lazer ou como uma alternativa futura para trabalhar. Neste capítulo, será descrito um breve histórico do ensino da programação no Brasil e no mundo, a importância de aprender a programar e projetos que incentivam crianças e adolescentes a se interessarem pela área computacional.

2.1 HISTÓRICO

Nos últimos anos, o ensino da programação nas escolas vem sendo debatido constantemente, por conta disto, muitas pessoas leigas no assunto acreditam se tratar de algo recente, porém, a prática em alguns lugares já ocorre há muitos anos.

Um dos pioneiros no ensino da programação é o matemático Seymour Papert, que juntamente com uma equipe de pesquisadores do Instituto de Tecnologia de Massachusetts - MIT desenvolveram a linguagem LOGO. (POCRIFKA; SANTOS, 2009)

Esta linguagem foi criada na década de 60, com o objetivo de ensinar programação a crianças de forma simples e inteligente, através de uma interface intuitiva que contava com uma tartaruga. O propósito era fazer com que a tartaruga se movimentasse, e para isto, a cri-

ança deveria escrever comandos chave existentes na linguagem, como por exemplo “para frente e para trás”.

Seymour Papert afirmava que “o objetivo da linguagem LOGO era o de oportunizar as crianças a aprender com prazer a programar e assim potencializar a aprendizagem”. (PAPERT, 1997; SOUZA, 2007 apud POCRIFKA, 2009; SANTOS, 2009).

O ensino de programação vem acontecendo em algumas escolas brasileiras, no estado de São Paulo, por exemplo, a coordenadora do Núcleo de Tecnologia do Colégio Santa Maria, Muriel Alves, relata que há 20 anos o colégio está envolvido com projetos que tem como base a programação de computadores, no começo, era utilizada a linguagem de programação LOGO, e desde 2007 é utilizada a linguagem Scratch. (ALVES apud ALVAREZ, 2014)

No Rio de Janeiro não é diferente, há seis anos, a professora da PUC/Rio Clarisse de Souza, ensina aos alunos a programar fazendo uso da linguagem LOGO, desenvolvida especificamente para o ensino aprendizagem de programação. A prática acontece através de algoritmos e rotinas de exercícios. Um dos jogos propostos pela linguagem tem como objetivo a travessia de um sapo para o outro lado do rio, esta travessia conta com obstáculos e é responsabilidade do aluno pensar em algum modo para que o sapo consiga chegar ao outro lado sem dificuldade. Esta iniciativa teve como inspiração o projeto *Scalable Game Design*, criado pelo professor Alexander Repenning, da Universidade do Colorado, nos Estados Unidos. (SOUZA, 2016)

Outras escolas brasileiras também se interessaram pela programação visando o aprendizado dos alunos, é o caso do colégio Visconde de Porto Seguro, em São Paulo. A professora Maria Fernanda Balugani, responsável pelo projeto, surgiu com a proposta, depois de conhecer ferramentas de ensino computacional na internet.

Segundo a professora,

“O aprendizado foi muito mais dinâmico do que nas turmas anteriores sem o uso da tecnologia. Aos cinco anos de idade, muitas dessas crianças já haviam tido seu primeiro contato com dispositivos digitais como *tablets* e smartphones, mas a experiência de desenvolver era inédita”. (OLHAR DIGITAL, 2013)

O Instituto de Tecnologia de Massachusetts - MIT tem como missão, o avanço do conhecimento e educar os alunos em ciência, tecnologia e outras áreas de estudo que melhor servem a nação e o mundo no século 21 (MIT, 2016). E com o passar dos anos, o instituto tem conseguido alcançar seus objetivos, através de diversos projetos relacionados ao ensino de

programação e robótica. Seus pesquisadores acreditam que ensinar a criança a desenvolver tecnologia, só traz benefícios, segundo Mitchel Resnick, criador da linguagem Scratch, “é importante aprender a codificar, não só pelas oportunidades de trabalho, mas pela possibilidade de ver o mundo de novas maneiras”. (RESNICK apud CARVALHO,2015)

O ensino de programação, por mais que não seja algo novo, vem ganhando força com diversas iniciativas que contam com o apoio de celebridades, como o então presidente dos Estados Unidos Barack Obama, ou ícones da tecnologia como Bill Gates (Google) e Mark Zuckerberg (Facebook). Estas iniciativas são de grande importância para o incentivo a tecnologia, ao dar a oportunidade ao aluno a desenvolver programas que podem ser utilizados em tarefas do dia a dia, do mesmo modo que incentiva ao aluno a estudar e buscar conhecimento através da programação de computadores.

2.2 A IMPORTÂNCIA DE APRENDER A PROGRAMAR

É cada vez mais comum que as disciplinas menos adoradas pelos alunos da área computacional sejam as de programação, segundo Deters et al. (2016) “tais disciplinas são consideradas desafiadoras pelos alunos, a consequência disto, é o elevado número de problemas de aprendizagem, acarretando reprovações e desistências”. Estes problemas podem surgir na forma de uma dificuldade na matemática, ou na dificuldade em interpretar problemas, por exemplo.

Tudo isto poderia ser amenizado se os alunos tivessem contato com a programação desde as séries iniciais, pois no futuro, caso optassem por cursos na área, não teriam o impasse inicial na disciplina, pois já teria um conhecimento prévio. Mas é válido lembrar que a programação não serve somente para o ramo computacional, ela pode ser aplicada em diferentes áreas como a matemática, geografia e física. Foi o caso do aluno de Mestrado da Universidade do Norte do Paraná que utilizou a linguagem de programação Scratch para o ensino de geometria em uma escola da região, segundo Cabral (2015) “na sociedade do conhecimento, as tecnologias ganham contornos mais definidos, principalmente para a educação, visto que são importantes no processo educativo e a tecnologia pode apresentar vantagens e desvantagens para o ensino e aprendizagem”.

Para Almeida (2006)

“Não existe uma causa única que justifique as bases das dificuldades com a linguagem matemática, que podem ocorrer por falta de aptidão para a razão matemática ou pela dificuldade em elaboração do cálculo. Essa di-

ficuldade não se relaciona com a ausência das habilidades básicas de contagem, mas sim com a capacidade de relacioná-las com o mundo”.

Crianças e adolescentes estão cada vez mais inseridos no ambiente tecnológico, através de computadores, *tablets* e principalmente os celulares, diante disto, se vê a necessidade de incentivá-las a buscar conhecimento na área de implementação, pois além de utilizar da tecnologia, melhor ainda é saber como implementá-la.

2.3 INICIATIVAS NACIONAIS E MUNDIAIS VOLTADAS A PROGRAMAÇÃO

A internet possui um grande poder de alcance, isto é, ao colocar uma informação na *web*, ela será espalhada rapidamente. Além disto, os adolescentes no Brasil configuram o grupo de pessoas que mais acessa a internet. Conforme o estudo Acesso à Internet e a Televisão e Posse de Telefone Móvel Celular para Uso Pessoal, feito em 2015 pelo IBGE, os adolescentes com idade entre 15 e 17 anos somam 75,7% dos conectados, em contrapartida, adultos que possuem idade entre 30 e 34 anos somam 60,3%, isto é, à medida que a idade aumenta, o acesso à internet cai. (BELLONI,2015)

Os projetos voltados a programação utilizam a internet como um dos principais meios de divulgação, direcionando seu conteúdo a quem interessa. No Brasil e no mundo, há diversas iniciativas que buscam o interesse de crianças e adolescentes no âmbito computacional, a grande maioria são instituições sem fins lucrativos, que conseguem apoio através de doações.

2.3.1 ESCOLAS

Atualmente no Brasil, existem diversas escolas voltadas ao ensino de programação para crianças e adolescentes. Estas escolas proporcionam momentos de lazer, através de jogos e aplicações diversas, ao mesmo tempo em que utilizam ferramentas para o ensino de programação. Deste modo, os alunos aprendem se divertindo. A seguir, algumas escolas de programação no Brasil.

Super Geeks

A escola possui diversas franquias espalhadas no Brasil inteiro, e oferece cursos sobre programação, desenvolvimento de jogos e robótica. A faixa etária varia de acordo com o curso e módulo, e cada módulo possui uma particularidade. São eles:

Curso QuickCode : É um curso de curta duração voltado para crianças acima dos 7 anos, possui 4 módulos de 4 aulas cada: Criação de jogos com a ferramenta Scratch; Robótica, onde o aluno aprende os conceitos básicos utilizando o arduíno; Programação, onde a criança aprende os comandos essenciais de um algoritmo utilizando o jogo Minecraft; Criação de modificações, também utilizando o jogo Minecraft, porém com comandos mais complexos. Cada módulo custa R\$320,00 reais.

Curso Regular: É um curso de longa duração voltado para crianças e adolescentes, possui 16 aulas por fase. São elas:

Fase 0: Voltada para crianças e adolescentes que não possuem experiência tecnológica, visa ensinar conceitos básicos de programação e robótica.

Fase 1: Voltada para crianças e adolescentes com algum conhecimento em tecnologia, que desejam aprender programação através do desenvolvimento de jogos em 2D, utilizando ferramentas como Scratch, Minecraft e Kit de Robótica.

Fase 2: Esta fase é sugerida para aqueles alunos que já possuam um bom conhecimento na área tecnológico e que desejam aprender linguagem de programação e desenvolvimento de jogos, utilizando linguagens de marcação como o HTML, Unity e Unreal Engine.

Fase 3 a 9: Voltada para adolescentes com conhecimento intermediário em programação e robótica, desenvolvendo atividades mais complexas nas ferramentas Unity e Unreal Engine. Cada fase custa R\$1360,00.

As aulas são presenciais e acontecem 1 vez por semana. (SUPERGEEKS, 2015)

MadCode

A MadCode (ou em português, Código de louco) é um centro de Tecnologia Criativa para Crianças e Adolescentes, de 05 a 17 anos de idade. A escola oferece 4 cursos pagos, cada um acontece 1 vez por semana, e são voltados a tecnologia, são eles: Pre-Tech, Programação, Aplicativos e Jogos.

Pre -Tech: Tem como público alvo, crianças de 5 a 7 anos que tenham interesse em aprender conceitos de lógica utilizando ferramentas como o Scratch e Blockly.

Programação: Voltado para crianças e adolescentes de 8 a 17 anos, utiliza linguagens de programação como Java e PHP para desenvolvimento de algoritmos.

Aplicativos: O curso ensina a aplicação do pensamento computacional na criação de aplicativos para dispositivos móveis. Utiliza linguagens de programação como C++ e Python, e tem como público alvo crianças e adolescentes que possuam idade entre 10 e 17 anos.

Jogos: É um módulo avançado para desenvolvimento de jogos, indicado para alunos que tenham idade entre 10 e 17 anos, e que se interessem por ferramentas profissionais como a Engine. (MADCODE, 2015)

Ctrl + Kids

A escola também possui diversas franquias em todo o Brasil, e oferece vários cursos com teor tecnológico para crianças e adolescentes. Todos os cursos são pagos, e cada um é dividido por módulo, são eles:

Code Kids: É dividido em dois módulos, sendo que cada um, possui 18 horas aula. O curso acontece uma vez por semana, e é voltado para crianças dos 7 aos 9 anos, neste curso as crianças são estimuladas na criatividade, raciocínio lógico e no inglês. Através da programação, desenvolvimento de jogos, lógica e robótica. As ferramentas utilizadas são o Scratch, Minecraft e o Kit de robótica.

Curso Regular: Voltado para crianças e adolescentes dos 10 aos 16 anos, possui 8 módulos, com frequência de 1 vez por semana, e 18 horas aula cada módulo. Os módulos oferecidos são:

Módulo 1: Introdução a lógica de programação e à criação de jogos em 2D, utilizando o Scratch.

Módulo 2: Criação de jogos em 2D e introdução de jogos em 3D, utilizando o Minecraft.

Módulo 3: Criação de jogos em 3D e robótica, utilizando o kit de robótica LEGO.

Módulo 4: Criação de aplicativos e robótica, utilizando a linguagem PHP.

Módulo 5: Criação de jogos em 3D intermediário.

Módulo 6: Desenvolvimento Web, utilizando HTML e PHP.

Módulo 7: Administração de servidores e criação de servidor WEB.

Módulo 8: Banco de Dados.

(CTRLKIDS,2016)

Happy Code

A escola Happy Code (ou em português, código feliz) oferece 3 cursos pagos para crianças e adolescentes, também são divididos por módulos, e a frequência das aulas é de 1 vez por semana.

Curso Code School: Módulos diferenciados por conteúdo, são eles:

Fase 1: Programação, indicado para crianças que se interessem por lógica, utiliza Scratch.

Fase 2 e 3: Desenvolvimento de Jogos em 2D e 3D, indicado para crianças que gostem de jogos e conceitos de inteligência artificial, utiliza Scratch e Unity.

Fase 4: Aplicativos, voltado para o desenvolvimento e criação de aplicativos nas plataformas iOS e Android.

Curso de Robótica: Este curso possui 4 fases, onde os alunos aprendem os conceitos do tema utilizando drones com arduíno.

A escola não especifica faixa etária para cada atividade. (HAPPYCODE,2015)

Gênio Azul

A escola de Fortaleza funciona há alguns anos, e tem como missão “Desenvolver competências através da educação tecnológica (programação, games e robótica), estimulando os alunos a pensar e criar de forma responsável, crítica e consciente.” (GENIO AZUL,2016)

O curso dado pela escola, é dividido em quatro módulos, sendo eles: programação, games, robótica e aplicativos. Cada módulo tem como objetivo o ensino aprendizagem da criança e adolescente na área tecnológica. A escola é privada, e o aluno interessado deve se inscrever e pagar cada módulo.

Como dito anteriormente, estas são algumas escolas voltadas ao ensino de programação. De um modo geral, existem muitas semelhanças entre elas, portanto, a tabela abaixo mostra algumas características das escolas.

ESCOLA	MÉTODOS	CONTEÚDO	TECNOLOGIAS E FERRAMENTAS	PÚBLICO ALVO
SuperGeeks	Cursos presenciais	Jogos, robótica e programação	Minecraft, HTML, scratch, unity, unreal engine, JavaScript, computadores, Kit de robótica	Crianças e adolescentes, dos 7 aos 17 anos.
Mad Code	Cursos presenciais	Programação, robótica, aplicativos e jogos	Scratch, Igeom, blockly, Java, PHP, python, C++, computadores, Kit de robótica	Crianças e adolescentes, dos 5 aos 17 anos.
Ctrl + Kids	Cursos presenciais	Jogos, programação, aplicativos e robótica	Minecraft, PHP, scratch, computadores, Kit de robótica	Crianças e adolescentes, dos 7 aos 16 anos.
Happy Code	Cursos presenciais	Programação, jogos, robótica e aplicativos	Unity, scractch, Ipads, computadores, Kit de robótica	Crianças e adolescentes.
Gênio Azul	Cursos presenciais	Aplicativos, jogos, programação e robótica	Kodu, scratch e Java, computadores, Kit de robótica	Crianças e adolescentes.

Tabela 1: Tabela de comparação entre escolas voltadas ao ensino de programação (do autor)

De acordo com a tabela, é possível afirmar que todas as escolas possuem o mesmo método de ensino, isto é, o conteúdo é dado através de cursos exclusivamente presenciais. Além do ensino de programação, as escolas também buscam o aprendizado tendo em seu plano de atividades a robótica, o desenvolvimento de jogos e aplicativos. Uma das ferramentas utilizadas que as escolas têm em comum é o software Scratch. Esta ferramenta possui uma

linguagem visual baseada em blocos, o que facilita a compressão das crianças e adolescentes no que se refere a programação.

2.3.2 SITES E PLATAFORMAS ONLINE

As iniciativas e projetos que apoiam o ensino de programação veem na internet, uma oportunidade para alcançar um número ainda maior de crianças e adolescentes que se interessem pela área. Abaixo, algumas iniciativas mundiais que ensinam programação através de plataformas online e sites.

Hour of code

A hora do código é uma introdução de uma hora à ciência da computação vinculada ao code.org, que disponibiliza diversos tutoriais para quem quer aprender a programar. Estes tutoriais são em forma de jogos, e cada jogo tem um objetivo específico. O interessado deve criar uma conta, e diariamente acessar o site, para desenvolver por uma hora, o jogo proposto. No fim, o usuário ganha um certificado. (DU; WIMMER; RADA, 2016)

Code Cademy

O Codecademy é um sistema de ensino que utiliza uma plataforma gratuita inteiramente online. Fundada em agosto de 2011, por dois estudantes da Universidade de Columbia, Ryan Bubinski e Zach Sims, a missão da escola é ser o maior centro de estudos de códigos gratuitos do mundo e para tanto, logo no início de seu desenvolvimento já conseguiu obter mais de US\$ 12 milhões a serem investidos na expansão dos cursos. (ADAMI,2016)

A plataforma disponibiliza tutoriais baseados em diversas linguagens como JavaScript e CSS.

Khan Academy

A Khan Academy é uma organização sem fins lucrativos totalmente a distância, tem como objetivo ensinar crianças, adolescentes e adultos, sobre diversos assuntos por meio de tutoriais na internet. A plataforma dispõe dos mais variados conteúdos, como a matemática, português, história da arte, economia e programação. “A forma de a Khan Academy disponibilizar o conhecimento para as pessoas se baseia em exercícios (variando de acordo com a matéria selecionada) práticos e videoaulas.” (MEDEIROS FILHO; MOURA, 2016)

Tynker

É uma plataforma de computação criativa onde milhões de crianças aprendem a programar e construir jogos, aplicativos e mais. Tynker oferece cursos on-line em ritmo individual para as crianças aprenderem a codificação em casa, bem como um currículo de programação envolvente para as escolas. (TYNKER,2015)

A plataforma proporciona diversas maneiras de aprendizado, e a criança pode escolher qual prefere, por exemplo, há tutoriais e jogos para a criança aprender brincando; na sala de aula, os professores podem utilizar o tempo em aula para ensinar programação com o conteúdo disponibilizado no site; nos *tablets* e *Ipads*, o aluno pode usufruir de aplicações *mobile*, e além disto, há acampamentos anuais e cursos oferecidos para o aluno quando o mesmo não estiver em horário de aula.

We Can Code It

We Can Code It (ou em português, Nós podemos programar ISTO), investe na diversidade da Engenharia e Tecnologia. Fundada por Mel McGee em 2013, tem como objetivo a criação e implementação de currículos inovadores, que mantenham o ritmo com os tempos modernos. Usando programas contemporâneos e tecnologia de ponta, oferecendo educação, orientação e motivação em STEM.

We Can Code It oferece educação e orientação em Tecnologia e Engenharia para aqueles que estão sub-representados na área. Com o foco no empoderamento das mulheres, afro-americanos e hispânicos americanos. (WECANCODEIT,2016)

Programaê

A organização sem fins lucrativos Code Club Brasil lançou, em 2014, a plataforma online de programação para crianças, denominada “Programaê”.

Criada em parceria com a Fundação Lemman, a ferramenta é gratuita e funciona como um incentivo ao aprendizado de programação. De acordo com o Cobe Club, a organização capta programadores e entusiastas que queiram compartilhar seu conhecimento, além de oferecer conteúdo de outras organizações como Codeacademy, Khan Academy, Scratch e Code.org. (OLHAR DIGITAL,2014)

A Plataforma disponibiliza material diferenciado para professores e alunos, isto é, para o professor que deseja utilizar esta ferramenta, é necessário que o mesmo acesse as lições e as modifique de acordo com a necessidade, enquanto o aluno, pode acessar os tutoriais disponíveis para aprender programação.

Made with Code

Made with Code (ou em português, feito com o Código) é uma iniciativa para patrocinar a criatividade, as meninas, e código, de uma só vez. O movimento é projetado para fazer três coisas: para inspirar meninas, celebrando as mulheres e meninas que estão usando o código para fazer grandes coisas; para envolvê-las para tentar codificação por meio de projetos e recursos introdutórios; e para sustentar o seu interesse através da criação de alianças e comunidade em torno de meninas e codificação. (MADEWITHCODE,2015)

O principal objetivo, é trazer mais meninas para o âmbito computacional, e incentivá-las a programar, optando por um método não tradicional de ensino, pois utilizam o computador como mediador entre o aluno e o conteúdo, que pode ser apresentado por um professor, através de tutoriais em rede, ou através de jogos e exercícios disponibilizados na internet. Na tabela a seguir, a comparação destas iniciativas, bem como as tecnologias e ferramentas utilizadas.

SITE/PLATAFORMA	MÉTODO	TECNOLOGIAS E FERRAMENTAS	PÚBLICO ALVO
Hour of Code	Vídeos, tutoriais, jogos, puzzle	Minecraft, Blockly	Crianças e adultos
Codecademy	Exercícios, algoritmos, leitura, compiladores	HTML, CSS, JavaScript, jQuery, Ruby, python, PHP, Java	Estudantes

Khan Academy	Jogos, animação, exercícios, vídeos	JavaScript, ProcessingJS, HTML, CSS, banco de dados	Estudantes
Tynker	Jogos, tutoriais, exercícios, vídeos, acampamentos anuais	Game Design Studio, programação em blocos	Crianças e adolescentes
We Can Code It	Vídeos, tutoriais, exercícios, acampamentos anuais, jogos	Minecraft, JavaScript, programação em blocos	Adolescentes e adultos, em especial mulheres, negros, estrangeiros
Programaê	Vídeos, jogos, exercícios, tutoriais,	JavaScript, CSS, HTML, scratch,	Professores e estudantes
Made with Code	Jogos, animação, exercícios, tutoriais, vídeos	Programação em blocos, Canvas, blockly,	Mulheres

Tabela 2: Tabela de comparação entre sites e plataformas online voltadas ao ensino de programação (do autor)

De acordo com a tabela 2, os recursos utilizados pelos sites e plataformas são semelhantes, é o que indica a coluna MÉTODO, tutoriais e jogos são os principais utilizados para o ensino de programação. Pode-se afirmar que há claramente uma preferência em utilizar os jogos para o ensino e aprendizagem de programação.

Para Tarouco et al. (2004), “Os jogos podem ser ferramentas instrucionais eficientes, pois eles divertem enquanto motivam, facilitam o aprendizado e aumentam a capacidade de retenção do que foi ensinado, exercitando as funções mentais e intelectuais do jogador. ”

Já a iniciativa Made with Code tem seu público alvo bem específico: as meninas. Isto porque na área computacional, há uma evasão feminina muito grande, pois, a falta de incenti-

vo é proporcional. Portanto, esta iniciativa tem como objetivo buscar o interesse das meninas em aprender a programar, bem como incentivar a persistência das mulheres na área.

2.3.3 PROJETOS

Diante de tantos métodos inovadores, alguns ainda recorrem a sala de aula. É o caso dos projetos abaixo, que em parceria com as escolas, proporcionam aulas de programação sem sair do ambiente escolar.

Code.org

É uma organização sem fins lucrativos com sede em Seattle, Washington. Seu objetivo é tornar a educação em ciência da computação disponível para todos os estudantes de todas as escolas públicas até 2020. A organização tem feito muito sucesso no mundo todo, por conta dos vídeos de divulgação que contam com a participação de Bill Gates, Mark Zuckerberg e outros nomes da área ou celebridades conhecidas. (DU; WIMMER; RADA, 2016)

Um dos nomes mais conhecidos que apoia do projeto, é o ex presidente dos Estados Unidos, Barack Obama. Segundo ele, a solução para garantir que todos os trabalhadores tenham uma chance justa de sucesso, na atual economia, cercada por tecnologias, é através da “codificação e a educação em ciência da computação”. (SHAPIRO, 2016)

Code Club Brasil

Code Club, ou em português “Clube do código” foi criado em 2012 pela britânica Clare Sutcliffe em parceria com a desenvolvedora Linda Sandvik, com um único objetivo: Ensinar programação para crianças e adolescentes do mundo todo. Esta iniciativa acontece em diversos países, e é totalmente gerida por voluntários, além de ser gratuita. Todo o material está disponível no site oficial.

O clube do código brasileiro, surgiu em 2013, e possui a mesma metodologia “Os cursos 1 e 2, utilizam o Scratch como ferramenta para ensinar programação. O curso 3 é uma introdução ao desenvolvimento web e utiliza HTML e CSS, e o curso 4 utiliza Python.” (GERALDES, 2014)

Jovem Programador

É um projeto social, sem fins lucrativos, criado pela Designa em parceria com a iMasters com objetivo de ensinar programação para crianças e jovens a partir dos 7 anos de idade. (JOVEMPROGRAMADOR,2015)

O projeto conta com voluntários que ministram as aulas pessoalmente ou através de tutoriais em vídeo. As escolas também podem participar, desde que possuam a devida estrutura.

Scalable Game Design

O objetivo deste projeto é trazer ciência da computação para o ensino médio. A estratégia é reformar a educação do ensino médio baseado em TI em nível sistêmico, explorar noções de jogos com uma abordagem e aspectos educacionais e aspectos motivacionais na área de TI. (REPENNING et al., 2014)

Foi desenvolvido pelo professor Alexander Repenning, na Universidade de Colorado em Boulder, nos Estados Unidos. Aqui no Brasil, há o projeto associado que também incentiva crianças e adolescentes a aprender programação, este projeto é realizado na PUC do Rio de Janeiro, e conta com o apoio de algumas agências de fomento. Atualmente o projeto no Brasil é aplicado em três escolas, os participantes são do ensino médio e fundamental, e as escolas contempladas se situam na cidade do Rio de Janeiro e Niterói.

Girls who Code

Tem como objetivo, alcançar a paridade de gênero nos campos de computação, para garantir a prosperidade econômica das mulheres, famílias e comunidades em todo o globo, e equipar os cidadãos com as ferramentas do século 21 para a inovação e mudança social. Quanto mais meninas estiverem envolvidas com ciência da computação em uma idade jovem mais mulheres trabalharão nas áreas de tecnologia e engenharia. (GIRLSWHOCODE,2016)

Girls who Code (ou em português, garotas que programam) é uma iniciativa patrocinada pelo Google e funciona nas escolas dos Estados Unidos. Turmas de meninas do ensino fundamental a médio são formadas, e os monitores voluntários oferecem palestras e aulas relacionadas a programação.

Real Programming for Kids

Real Programming for Kids (ou em português, Programação real para crianças) oferece um ambiente rico e desafiador de aprendizado para todos os níveis, com toda a diversão

dos videogames. Ensinando programação de computador para milhares de crianças e adolescentes em Ontário desde 2000. (REALPROGRAMMING4KIDS,2016)

Desenvolvida no Canadá há vários anos, além de cursos regulares com professores dentro de uma sala de aula, há também o acampamento de férias anual, que consiste em uma experiência diferente de aprendizado em programação.

Kids Can Code

Com o objetivo de simplesmente ensinar crianças a programar. O projeto acredita que um conhecimento básico de engenharia de software fornece um conjunto de competências fundamentais que é vital, tanto para o futuro da criança quanto para o futuro da economia global. Não é apenas sobre STEM (ciência, tecnologia, engenharia e matemática), mas também sobre artes liberais: ser capaz de pensar criticamente sobre problemas é o primeiro passo para encontrar soluções. Uma rápida introdução à ciência da computação é um investimento no futuro. (KIDSCANCODE,2015)

Kids Can Code (ou em português, Crianças podem programar) é um projeto que oferece diferentes modalidades de ensino, são elas: programa “depois da escola”, isto é, o aluno aprende programação no horário inverso ao que estuda; “sessão na escola”, o aluno aprende programação dentro da escola, com professores voluntários; e “workshops nos finais de semana”, onde o aluno participa de estandes e palestras sobre programação aos finais de semana.

A tabela a seguir faz uma comparação entre os projetos citados.

PROJETO	MÉTODO	TECNOLOGIAS E FERRAMENTAS	PÚBLICO ALVO
Code.org	Aulas semanais com monitores/professores	Blockly, JavaScript, computadores	Estudantes
Code Club Brasil	Aulas semanais com monitores/professores	Scratch, Python, HTML, CSS, computadores	Crianças e adolescentes
Jovem Programador	Aulas semanais com monitores/professores	Java, NetBeans, Eclipse, banco de dados, compu-	Crianças e adolescentes, a partir dos

		tadores	7 anos.
Scalable Game Design	Aulas semanais com monitores/professores, jogos, vídeos, tutoriais	Programação em blocos, JavaScript, animação	Crianças e adolescentes
Girls who Code	Aulas semanais com monitores/professores, palestras	Scratch, JavaScript	Mulheres
Real Programming for Kids	Aulas semanais com monitores/professores, acampamentos de férias, cursos online	Alice, Visual Basic, Microsoft Visual Studio, Java, C#, C++, Minecraft, Unity	Crianças e adolescentes
Kids Can Code	Aulas semanais com monitores/professores, acampamentos de férias	Python, Scratch, Pygame framework	Crianças e adolescentes

Tabela 3: Tabela de comparação entre projetos voltados ao ensino de programação (do autor)

De acordo com a tabela 3, os métodos aplicados em todos os projetos possuem grande semelhança, isto é, os alunos participantes do projeto, aprendem programação em uma sala de aula ou laboratório juntamente com o professor ou monitor responsável pelo curso. Este método permite que o professor/monitor consiga ter um acompanhamento mais específico sobre o rendimento do aluno. Além do método “tradicional”, os projetos Kids Can Code e Real Programming for Kids oferecem também acampamentos de férias aos seus alunos, como uma outra alternativa para se aprender programação em um ambiente diferente.

Fazendo uma análise geral em relação ao capítulo 2, é válido afirmar que, a programação para crianças e adolescente é uma prática incentivada pelo mundo todo. Mesmo no Brasil, onde o ensino é relativamente novo, há diversas opções para que as crianças se interessem por programação, e os demais assuntos que rodeiam o tema, como robótica e desenvolvimento de jogos. As iniciativas privadas e gratuitas tem grande impacto no que se refere a aderência dos jovens no mercado computacional, pois é um nicho em ascensão onde os ado-

lescentes principalmente, veem a oportunidade do aprendizado como lazer a curto prazo, e como utilidade futura.

3. FERRAMENTAS PARA O ENSINO DE PROGRAMAÇÃO

Diferentes ferramentas existem para auxiliar alunos e professores no ensino de programação. Os professores podem utilizar estas ferramentas como complemento na sala de aula, e os alunos para aprenderem sozinhos em casa, ou em grupo de colegas e amigos. O interesse do aluno em aprender programação é crucial para um bom rendimento. Jenkins (2002) acredita que, as motivações que levam o aluno a programar, impactam diretamente em seu aprendizado:

“Os alunos se aproximam de graus da computação por uma variedade de motivos. Alguns tem um interesse genuíno no assunto (motivação intrínseca), outros pela possibilidade de carreira lucrativa (motivação extrínseca) e outros para agradar pais e familiares (motivação social). A forma de motivação parece ser um fator. Talvez, não surpreendentemente, os estudantes que lutam com a programação, são mais propensos a terem uma motivação extrínseca do que aqueles colegas que se destacam” (JENKINS,2002).

Ainda que parta de uma motivação pessoal, programar pode ser um desafio para muitos alunos, pois são necessárias, habilidades específicas como o raciocínio lógico, pensamento analítico e a contínua resolução de problemas. Gomes (2000) afirma que “uma das grandes dificuldades reside principalmente na compreensão, e em particular, na aplicação de noções básicas, e na criação de algoritmos que resolvam problemas concretos”. Já Alves e Costa (2004) declaram que “Alunos que apresentam deficiências advindas do ensino fundamental e médio, encontram dificuldades em aprender conceitos simples de programação de computadores”.

A programação de computadores está diretamente ligada aos princípios básicos da matemática, é o que afirma Oro et al. (2015), “O estudante, ao programar, aplica seus conhecimentos prévios de matemática, aprende e desenvolve outros conhecimentos necessários à programação, sem saber necessariamente que o está fazendo”. Assim como a linguagem de programação tradicional, existem as chamadas “linguagens visuais” que são softwares educacionais produzidos para o ensino de programação mais lúdico, isto é, o aluno aprende de for-

ma flexível, o que estimula sua criatividade. É válido salientar que, esta não é uma forma menos importante de ensinar programação. Slany (2012) reforça “Não é sobre programação para leigos, mas sobre motivar e evitar frustração devido a, por exemplo, fluxos de trabalho desnecessariamente complicados ou a dificuldade de sinalizar erros de sintaxe nas linguagens tradicionais”.

Jogos educacionais estão cada vez mais presentes quando se fala em ensino de programação, atualmente existem diversas plataformas que utilizam os jogos para que as crianças se familiarizem com conceitos de lógica, sem grandes dificuldades. Os jogos educacionais se mostram atrativos pois, é a partir deles que a criança e o adolescente começam a interagir com o mundo tecnológico, é o que afirma Gros (2003) “Videogames estão entre os principais meios de acesso que crianças e adolescentes possuem ao mundo tecnológico, seu primeiro contato se dá, através de jogos de computador”.

Sintetizando o assunto, algumas ferramentas utilizadas para o ensino de programação serão apresentadas a seguir:

Linguagem LOGO

Criada na década de 60 por um grupo de pesquisadores do Instituto de Tecnologia de Massachusetts (MIT) liderado pelo matemático Seymour Papert, tornou-se mundialmente conhecida por conta do personagem principal da ferramenta. Esta linguagem foi desenvolvida para o ensino de lógica através de uma interface simples que contava com um objetivo: manipular uma tartaruga. Os comandos utilizados eram intuitivos, se a criança quisesse mover a tartaruga 10 vezes para o lado esquerdo, bastava escrever na ferramenta “paraesquerda 10”.

Segundo Prado (1999), a linguagem é sofisticada, pois possui “características pertencentes a três paradigmas computacionais distintos: procedural, orientado a objetos e funcional”.

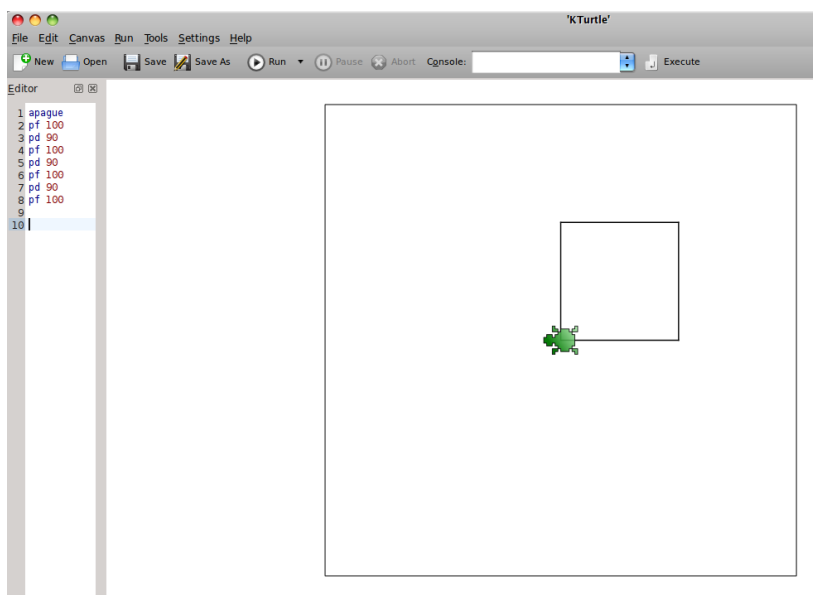


Figura 1: Interface da Ferramenta LOGO (captura de tela)

Blockly

Esta ferramenta foi desenvolvida pela Google em 2012 e, como o nome já diz, é baseada em blocos. A linguagem em blocos é frequentemente usada para o ensino de programação para crianças, por possuir interface lúdica e de fácil manipulação. Inclusive, esta é muito semelhante a outra ferramenta, Scratch. De acordo com Marron, Weiss e Wiener (2012), a popularidade da ferramenta Scratch sugere que essa tenha seu estilo de programação mais acessível a crianças, no entanto, a combinação de visual X programação faz da ferramenta Blockly uma excelente opção para demonstrar técnicas de programação sem que seja necessário exigir pseudo código, como em outros casos, pois afinal, o código criado através da linguagem de blocos, também pode ser transformado em um código funcional em Javascript, tornando este, o diferencial da ferramenta.

Alice

Criada em 1999, esta ferramenta de autoria possui um ambiente 3D para modelagem de objetos e foi desenvolvida por um grupo de pesquisa da Universidade Carnegie Mellon liderado por Randy Pausch. (COOPER; DANN; PAUSCH, 2016)

Estes objetos podem incorporar diferentes estados, a depender do momento, e estes estados são manipulados através dos métodos, e por conta disto, Alice utiliza os conceitos básicos de programação orientada a objetos. No início, a modelagem só podia ser feita com a linguagem Alice, similar a Python, mas as versões atuais permitem que o usuário construa

mundos virtuais através da própria interface gráfica da ferramenta. (LIMA; TEICHRIEB; KELNER, 2004).

Code Combat

Desenvolvido em 2013 por uma equipe de profissionais das mais diversas áreas (engenharia, gestão de negócios, designer), é uma ferramenta que combina jogo e linguagem de programação. (CODECOMBAT, 2016)

Funciona da seguinte forma: O usuário tem acesso a diferentes mundos (níveis) sendo que apenas o primeiro nível está liberado, cada nível possui algumas fases, que vão aumentando a dificuldade de acordo com o progresso do usuário. De um lado, existe um ambiente com objetos e um avatar que representa quem está jogando, e que precisa realizar diferentes tarefas como, apanhar objetos, movimentar-se, lutar, etc. Do outro, é a interface de comandos, que o usuário precisa escrever para que o avatar realize estas atividades, desta forma, o usuário consegue visualizar de uma maneira mais clara, o que acontece e quando acontece cada tarefa. A cada nível, a linguagem vai mudando, mas isto é personalizável. Apenas o primeiro nível (e algumas fases deste) é gratuito, portanto, para continuar jogando, é necessário comprar os demais módulos bloqueados.

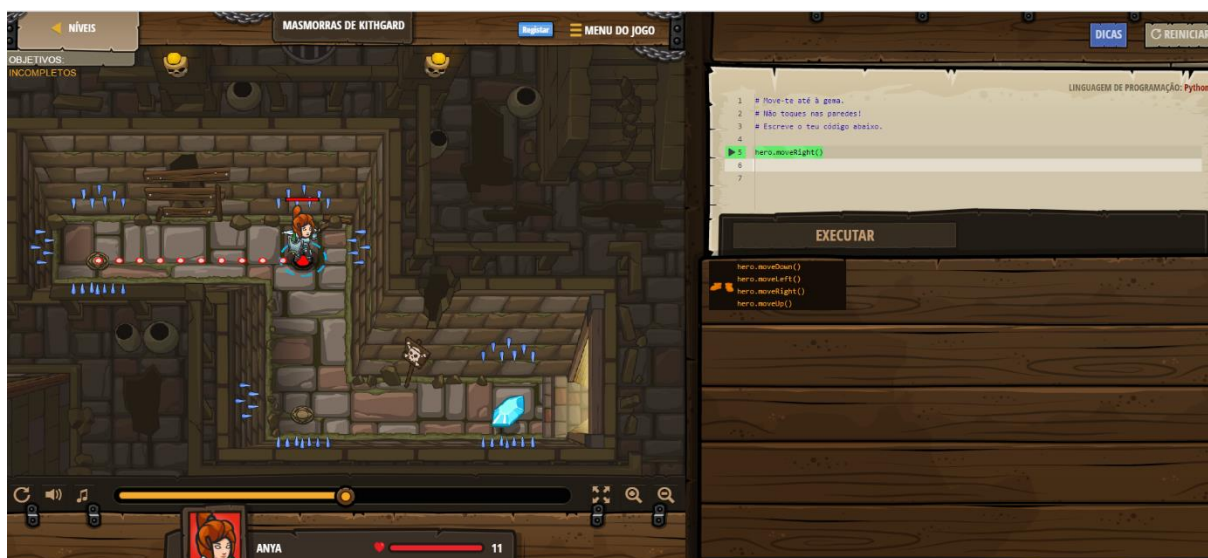


Figura 2: Interface do Code Combat (captura de tela)

Code Studio

É uma ferramenta online para ensino de programação voltada a crianças das mais diversas idades. Esta ferramenta foi criada em 2013 em parceria com desenvolvedores da Google, Microsoft e Twitter. (SENA et al., 2015)

Ela é mundialmente utilizada através do projeto “Hora do código” e possui diversos jogos separados por temas e dificuldade. Por conta do seu público alvo, os jogos possuem temas populares, como Angry Birds, Frozen, etc. A linguagem utilizada também é em blocos e seu diferencial são as frases motivacionais a cada tarefa completada corretamente, o que resulta em um bom entretenimento.

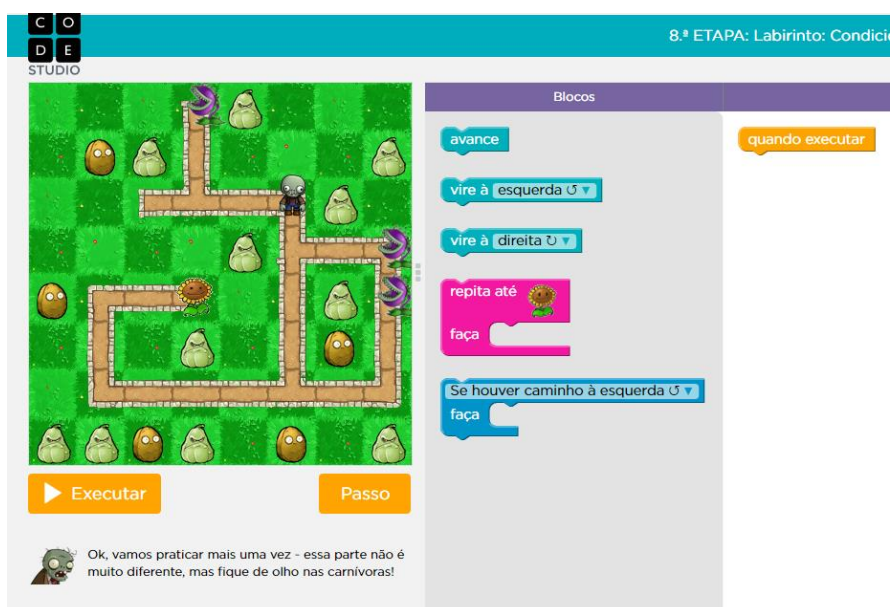


Figura 3: Interface do Code Studio (captura de tela)

Lightbot

É um jogo educacional, desenvolvido pela Armor Games. O objetivo é programar um pequeno robô para iluminar todos os blocos azuis em uma placa. Para alcançá-lo, é preciso codificar uma série de instruções utilizando um conjunto específico de comandos, em um espaço que é limitado. (GOUWS; BRADSHAW; WENTWORTH, 2013)

O jogo possui vários níveis, mas todos com o mesmo ambiente: um personagem (robô) com alguns quadrados pelo caminho, esperando por uma sequência de passos que podem variar entre pular e acender uma lâmpada. Como a sequência será feita, depende do nível em

que o usuário se encontra. A cada fase, um conceito é incorporado, ao final, o usuário terá visto recursividade, laços de repetição, laços condicionais e funções.

Scratch

É uma ferramenta em 2D criada pelo grupo de pesquisa Lifelong Kindergarten, liderado pelo professor Mitchel Resnick, no MIT, com o objetivo de ensinar programação para crianças na faixa etária de 8 a 12 anos. (SCRATCH,2016)

Mitchel Resnick é um grande defensor do ensino de programação para crianças e adolescentes, em sua apresentação para o TED, Resnick(2012) disse “Se você aprende a escrever códigos, você pode escrever códigos para aprender”. Isto é, a lógica é um âmbito vasto, que pode contribuir para a aprendizagem do aluno em diversas áreas.

O Scratch se popularizou, e hoje é utilizado por pessoas de todas as idades que queiram aprender a programar. A linguagem utilizada é em blocos, e sua interface é simples e intuitiva. A ferramenta é gratuita e foi traduzida em mais de 40 idiomas. Com ela, é possível criar jogos e animações dos mais diversos tipos, utilizando os principais conceitos de lógica: laço de repetição, laço condicional, variáveis e vetores.

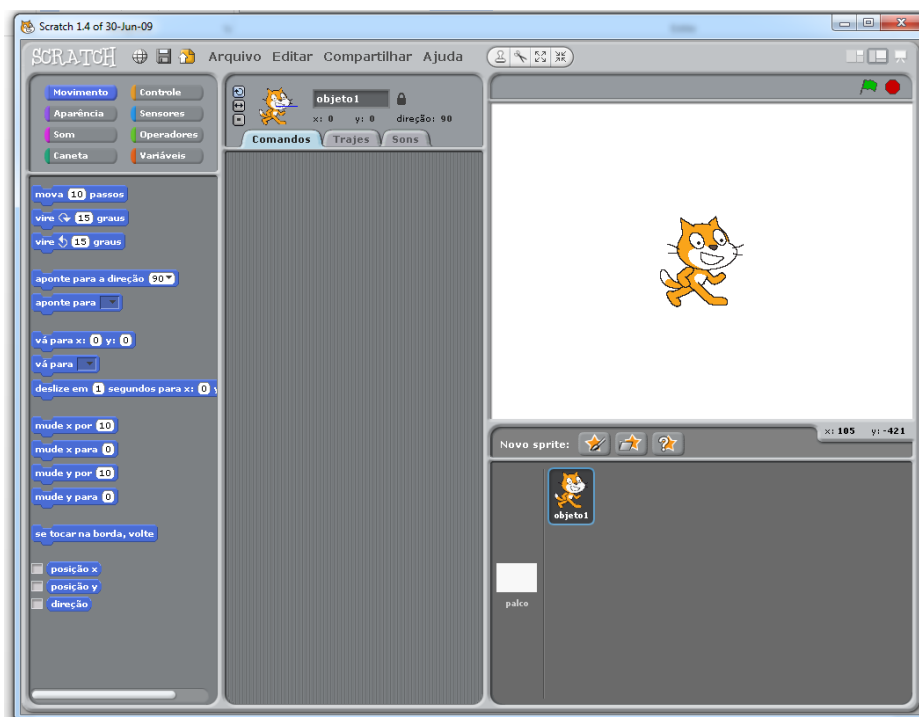


Figura 4: Interface do Scratch (captura de tela)

Code Hunt

É uma plataforma online de ensino de programação voltada para linguagens específicas. Ela foi criada pela Microsoft Research, para o desenvolvimento de códigos utilizando as linguagens Java ou C#, o usuário pode escolher em qual linguagem quer programar. (SENA et al., 2015)

O ambiente é bem interativo, com música, vídeos e pontuação dos usuários, o que pode ser um fator motivador para o aluno. Como a plataforma especificamente trata de duas linguagens, é recomendado que o aluno já tenha um conhecimento prévio sobre programação, se não, as tarefas podem se tornar difíceis para que o usuário consiga prosseguir. A plataforma é gratuita, mas seu único idioma é o inglês. A interface é dividida, de um lado, o aluno deve escrever o código corretamente, do outro, está o código incompleto. O conhecimento adquirido ao final de todos os módulos deverá ser: vetores, funções, variáveis, laços de repetição e lógica básica.

VisuAlg

Esta ferramenta é um editor e interpretador de códigos que utiliza uma linguagem próxima ao português de forma estruturada. Criada em 2009 pelo professor Cláudio Morgado, possui diversas versões, todas em português e gratuitas. Os códigos nesta ferramenta são chamados de “pseudocódigos” pois não possuem uma linguagem de programação propriamente dita. Souza (2009) complementa “o VisuAlg é um aplicativo que fornece aos estudantes que se iniciam nas disciplinas de programação ferramentas para digitar, executar e depurar o pseudocódigo para resolver problemas propostos nas aulas e em exercícios.”

Diferentemente de outras ferramentas, esta não necessita de conhecimento prévio para resolver os problemas, mas é recomendado que o aluno conheça o mínimo de lógica. A interface desta ferramenta é semelhante a um compilador, onde o aluno deve escrever suas linhas de código para executar determinada tarefa.

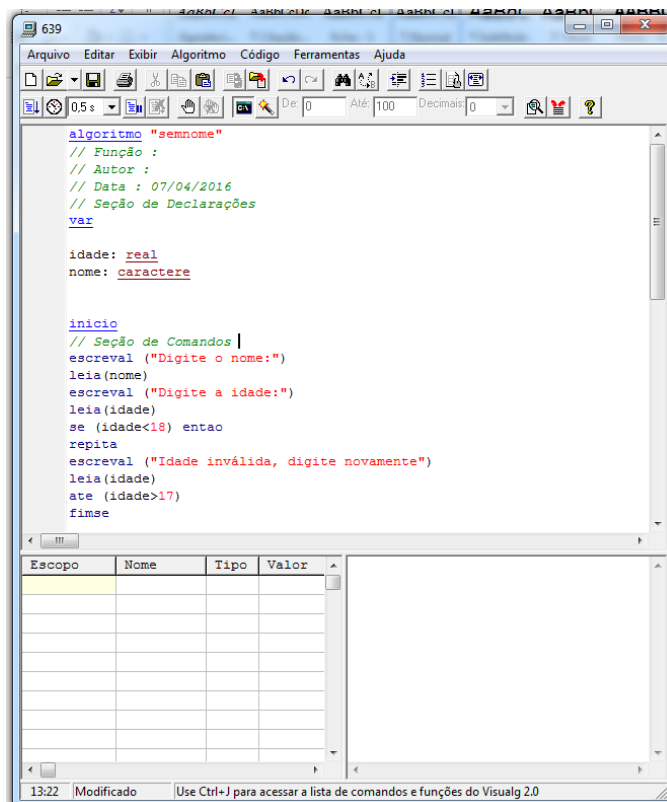


Figura 5: Interface do VisuAlg (captura de tela)

Crunchzilla

Desenvolvido em 2015 pelo grupo Geeky Ventures para a criação de jogos educacionais, é também um ambiente online, que inclui diferentes tutoriais interativos para crianças e adultos. Os interessados podem programar, construir e aprender, através do passo a passo, criando gráficos e animações. Os jogadores aprendem conceitos importantes de programação como variáveis, loops, estruturas de condição e funções. (QUEIRÓS, 2014)

A interface da plataforma é bem simples, de um lado, existe a tela que servirá para escrever os códigos, do outro, é o resultado da programação da tela anterior. As linhas de código servem para manipular formas geométricas, que variam de tamanho, cor e quantidade, de acordo com o que for programado. A ferramenta é gratuita, mas não possui versão em português.

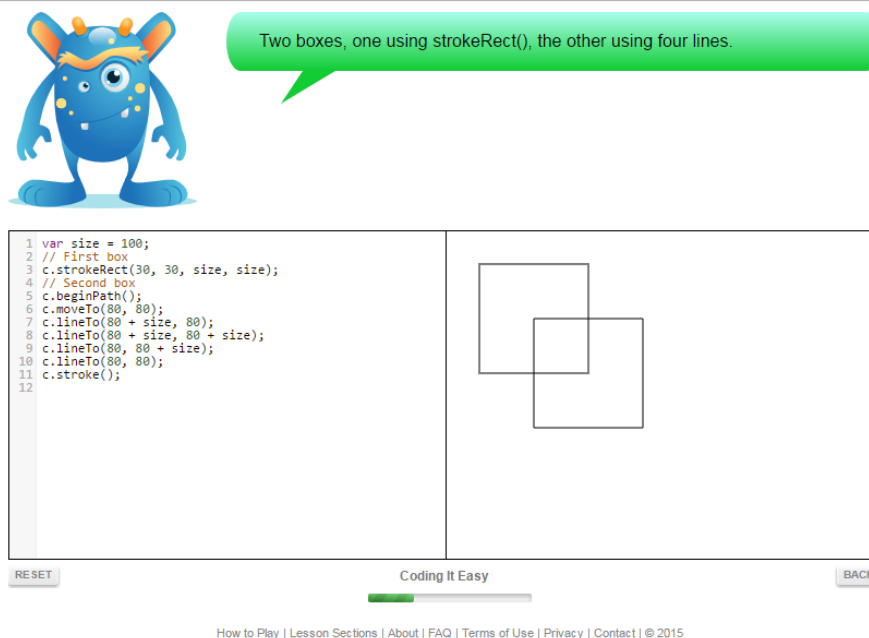


Figura 6: Interface do Code Monster (captura de tela)

The Foos

É um jogo criado para crianças a partir dos cinco anos de idade, com o objetivo de ensinar conceitos computacionais através de uma interface simples e divertida. O jogador deve guiar os personagens para realizar pequenas tarefas, e à medida que o jogador avança de fase, novos recursos e botões de comando são introduzidos. (GOMES, 2015)

O aluno manipula personagens utilizando ícones disponibilizados na tela, que precisam seguir uma sequência específica para que o personagem finalize a tarefa, a cada tarefa, o aluno vai arrecadando pontos, representados por estrelas, e cada módulo precisa de um número determinado de estrelas para ser desbloqueado. No fim, o aluno deverá ter aprendido conceitos como laços de repetição e condição.

Este jogo é utilizado mundialmente na Hora do código, e por conta disto, há apostilas traduzidas em alguns idiomas (incluindo português) para que o aluno a utilize como auxílio, para tirar dúvidas, etc.



Figura 7: Interface do The Foos (captura de tela)

YouCode

É uma plataforma online e gratuita que ensina conceitos de programação através de jogos, utilizando uma linguagem orientada a objetos, ActionScript. A interface do jogo possui telas divididas, em uma o usuário deverá escrever seu código, outra possui uma lista de funções/variáveis, outra, é a tela onde os comandos são representados, através de cenários e personagens diversos como gatos, robôs, morcegos, e outra é uma introdução da tarefa a ser realizada. O jogo possui níveis de dificuldade, a cada tarefa finalizada, o jogador vai aumentando sua grade de conceitos, como variáveis, funções, passagem por parâmetro e valor, etc.

Segundo Sena et al. (2015), um diferencial do jogo é “ele apresenta erros de sintaxe e comandos trocados, ao longo do tutorial. Isso faz com que o aluno já fique mais preparado para encontrar comandos inválidos ou que estejam fora da posição correta”.

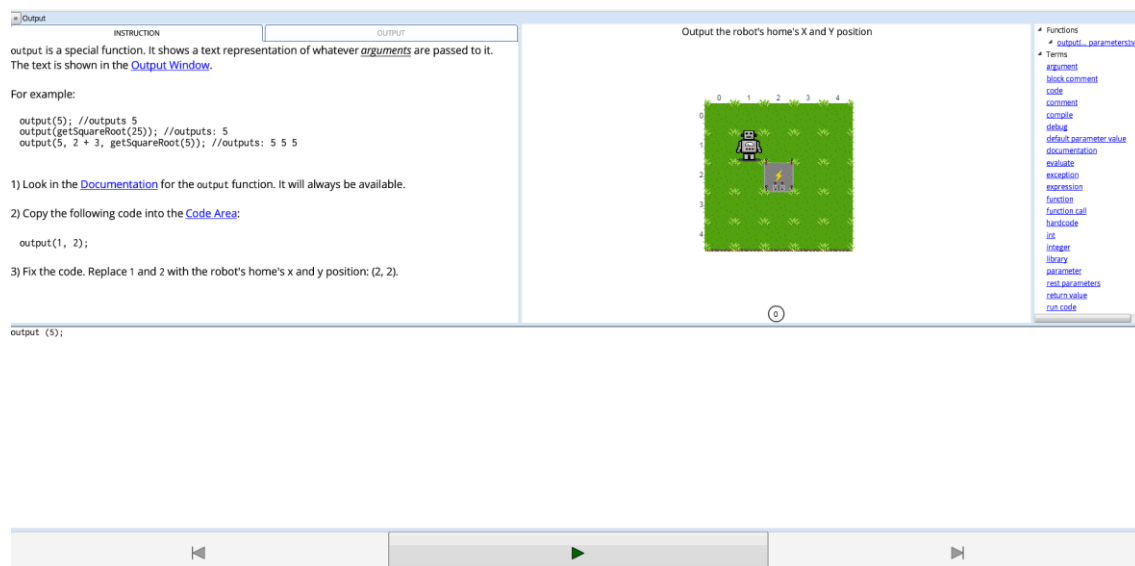


Figura 8: Interface do YouCode (captura de tela)

A partir das plataformas pesquisadas, foram elaboradas algumas tabelas a fim de caracterizar e diferenciar estas ferramentas.

FERRAMENTA	IDIOMA	MÉTODO	CONTEÚDO	DISPONIBILIDADE
Blockly	Diversos idiomas, incluindo português	Programação em blocos, jogo	Variáveis, laços de repetição/condição, funções	Gratuito, disponível para computador
Code Studio	Diversos idiomas, incluindo português	Programação em blocos, jogo	Variáveis, laços de repetição/condição, funções	Gratuito, disponível para computador
LighBot	Inglês	Comandos em sequência, jogo	Recursividade, funções	Gratuito, disponível para computador e dispositivos móveis
Scratch	Diversos idiomas,	Programação em blocos,	Variáveis, laços de repetição/condição,	Gratuito, disponível para computador

	incluindo português	jogo	vetores	
The Foos	Inglês	Comandos em sequência, jogo	Laços de repetição/condição, recursividade	Gratuito, disponível para computador e dispositivos móveis

Tabela 4: Tabela referente as ferramentas que possuem como método, a programação em blocos aplicadas aos jogos.

FERRAMENTA	IDIOMA	MÉTODO	CONTEÚDO	DISPONIBILIDADE
LOGO	Diversos idiomas, incluindo o português	Linguagem LOGO, jogo	Geometria, variáveis, laços de repetição/condição, funções	Gratuito, disponível para o computador
Alice	Inglês	Linguagem (Java, C++, C#), jogo	Variáveis, laço de repetição/condição, funções	Gratuito, disponível para o computador
Code Combat	Diversos idiomas, incluindo o português	Linguagem (JavaScript, Python, Lua), jogo	Variáveis, laços de repetição/condição, funções, recursividade, vetores, programação orientada a objetos, aritmética	Pago, com níveis iniciais gratuitos, disponível
Crunchzilla	Inglês	Linguagem (JavaScript), jogo	Variáveis, laços de repetição/condição, recursividade, funções, aritmética.	Gratuito, disponível para o computador
YouCode	Inglês	Linguagem orientada a	Variáveis, laços de repetição/condição,	Gratuito, disponível

		objetos, jogo	funções, programação orientada a objetos	para computador
--	--	---------------	--	-----------------

Tabela 5: Tabela referente as ferramentas que possuem como método, linguagens de programação aplicadas aos jogos.

FERRAMENTA	IDIOMA	MÉTODO	CONTEÚDO	DISPONIBILIDADE
CodeHunt	Inglês	Interpretador de algoritmos, utiliza linguagem Java e C#	Aritmética, variáveis, laços de repetição/condição, funções, vetores	Gratuito, disponível para o computador
VisuAlg	Português	Interpretador de algoritmos, utiliza pseudocódigo	Variáveis, laços de repetição/condição, função, vetores	Gratuito, disponível para o computador

Tabela 6: Tabela referente as ferramentas que possuem como método, linguagens de programação através dos interpretadores de código.

Após realizar um levantamento das ferramentas existentes voltadas para o ensino de programação, e elaborar tabelas comparativas, é viável considerar que existem inúmeras possibilidades para se aprender a programar, mesmo entendendo que cada um internaliza o conhecimento de uma forma diferente, cada ferramenta possui uma particularidade. Ferramentas que tem como metodologia os jogos, são mais voltadas a crianças, por possuir um cenário mais lúdico. Enquanto que ferramentas como o VisuAlg, que dispõe de uma interface mais objetiva, é mais voltada para pessoas que no mínimo, tenham um conhecimento básico de lógica. Além disto, a disponibilidade de ferramentas gratuitas é uma realidade, o que facilita que mais pessoas de diferentes situações econômicas tenham acesso a programação de computadores.

Para escolher as ferramentas que seriam utilizadas neste trabalho, era necessário que que estas atendessem alguns requisitos:

- Ser gratuita;
- Ser em português;
- Ser de fácil manipulação, pois afinal, as atividades foram desenvolvidas em um curto período de tempo;
- Não ter uma linguagem específica, pois, cada linguagem possui uma série de comandos nativos, o que dificultaria no nivelamento dos estudantes no questionário final;
- Ter os mesmos comandos e lógica correspondentes, pois, as duas turmas precisariam responder o mesmo questionário, então, o ideal é que se pudesse ensinar lógica e resolver os exercícios de forma mais semelhante possível;

E por fim, que os tópicos trabalhados na ferramenta fossem semelhantes, isto é, nem todas as ferramentas pesquisadas, trabalhavam o conteúdo de estruturas de repetição, por exemplo.

Por estas razões, as ferramentas escolhidas foram a Scratch e o VisuAlg.

4. PROPOSTA DE ATIVIDADES

Este capítulo tem como objetivo, apresentar através do seu detalhamento, a realização das atividades propostas pelo plano.

Neste caso, as atividades foram realizadas com alunos do 8º e 9 ano do ensino fundamental e alguns alunos do 1º ano do ensino médio da escola de educação básica Profª Maria Garcia Pessi nas dependências da Universidade Federal de Santa Catarina, Unidade Mato Alto, em Araranguá.

4.1 Plano de Atividades

Este plano de atividades é dividido em quatro etapas, as três primeiras etapas estão relacionadas com as atividades que foram realizadas com duas turmas em períodos distintos (matutino e vespertino). Estas atividades estão divididas por nome, descrição e objetivos. Já a

quarta etapa, consiste na aplicação de uma avaliação feita após dado todo o conteúdo das atividades propostas. Esta avaliação é a mesma para as duas turmas.

4.1.1 Atividade 1: Conceitos Básicos computacionais

Turma 1: Aprendendo conceitos com o Scratch

Data: 16/05/16

Horário: 08:30 as 11:50

Duração (horas): 3 horas e 20 minutos;

Ferramenta utilizada: Scratch

Turma 2: Aprendendo conceitos com o VisuAlg

Data: 16/05/16

Horário: 13:30 as 16:50

Duração (horas): 3 horas e 20 minutos;

Ferramenta utilizada: VisuAlg

Pré-requisito para execução da tarefa: Capacidade de entender a lógica de uma sequência de comandos qualquer, como uma receita de bolo. Além disto, é desejável que o aluno já tenha tido algum contato com o computador.

Justificativa: Esta atividade foi escolhida porque, para desenvolver um programa de computador, é necessário entender a lógica e entender o que é uma linguagem de programação. Além disto, para entender como um computador executa comandos escritos por nós, é necessário saber quais os comandos que devem ser utilizados.

Objetivos:

Objetivo geral: Esta atividade tem como objetivo, ensinar aos alunos o conceito de programação, bem como os artefatos necessários para escrever o primeiro programa de computador.

Objetivos específicos:

Entender o conceito de algoritmo, lógica e programação;

Conhecer a ferramenta proposta;

Aprender a manipular variável dentro da ferramenta;

Implementação de algoritmos;

Conteúdo Programático:

O que é computador?

O que é lógica?

O que é algoritmo?

Lógica + Algoritmo

O que é pseudocódigo?

O que é linguagem de programação?

O que é programa de computador?

Descrição da ferramenta proposta

Variáveis

Tipos de dados

Atribuição de dados

Comentários

Entrada e saída de dados

Exercícios

Método de ensino/ Desenvolvimento de programa

Os assuntos serão apresentados em aulas expositivas, sempre com discussão e participação dos alunos. Estudos e exercícios, sempre como forma de estimular à participação dos alunos. Aulas práticas em laboratório de Informática.

Resultados esperados: Dos alunos, se espera o entendimento básico sobre o que é lógica, programação, e o desenvolvimento de pequenos programas que possam ser uteis no dia a dia dos mesmos.

4.1.2 Atividade 2: Estruturas de seleção e operadores

Turma 1: Estruturas de seleção e operadores com Scratch

Data: 17/05/16

Horário: 08:30 as 11:50

Duração (horas): 3 horas e 20 minutos;

Ferramenta utilizada: Scratch

Turma 2: Estruturas de seleção e operadores com VisuAlg

Data: 17/05/16

Horário: 13:30 as 16:50

Duração (horas): 3 horas e 20 minutos;

Ferramenta utilizada: VisuAlg

Pré-requisito para execução da tarefa: Capacidade de entender a lógica existente na utilização de operadores, bem como entender como funciona uma sequência de comandos onde há uma tomada de decisões para ser executada. Como os operadores são utilizados geralmente nas aulas de matemática, desejável que já se tenha este conhecimento.

Justificativa: Os operadores, sejam eles lógicos, relacionais ou aritméticos, são essenciais para entender a lógica computacional por trás dos programas de computador, sem eles, se torna impossível desenvolver algumas operações básicas, como a soma, por exemplo. No caso das estruturas de seleção, um programa pode ser executado de maneira muito mais rápida e eficaz se utilizar estruturas para diminuir o custo de processamento.

Objetivos:

Objetivo geral: Esta atividade tem como objetivo, ensinar aos alunos como manipular estruturas de seleção e como utilizar os operadores nestas estruturas.

Objetivos específicos:

Entender o conceito e a manipulação de operadores;

Entender o funcionamento da tabela verdade;

Implementação de estruturas de seleção simples e composta nos algoritmos;

Conteúdo Programático:

Operadores

Operadores Relacionais

Operadores Aritméticos

Operadores Lógicos

Tabela Verdade

Estruturas de Seleção

Seleção Simples

Seleção Composta

Método de ensino/ Desenvolvimento de programa

Os assuntos serão apresentados em aulas expositivas, sempre com discussão e participação dos alunos. Estudos e exercícios, sempre como forma de estimular a participação dos alunos. Aulas práticas em laboratório de Informática.

Resultados esperados: É esperado que os alunos entendam o conceito apresentado e consigam refazer os exercícios dados anteriormente utilizando os operadores e as estruturas de seleção.

4.1.3 Atividade 3: Estruturas de Repetição

Turma 1: Estruturas de repetição com Scratch

Data: 18/05/16

Horário: 08:30 as 11:50

Duração (horas): 3 horas e 20 minutos;

Ferramenta utilizada: Scratch

Turma 2: Estruturas de repetição com VisuAlg

Data: 18/05/16

Horário: 13:30 as 16:50

Duração (horas): 3 horas e 20 minutos;

Ferramenta utilizada: VisuAlg

Pré-requisito para execução da tarefa: Os alunos devem ter aprendido a desenvolver um programa de computador, da mesma forma que devem ter entendido o conceito de estruturas de seleção.

Justificativa: Um programa que realiza seu objetivo é um programa útil. Mas de nada adianta um programa útil se o mesmo desperdiçar muito espaço da memória, ou levar um custo de tempo muito alto para ser executado, portanto, esta atividade foi escolhida para que os alunos aprendam a otimizar o programa, bem como automatizá-lo, isto é, o programa passa a realizar uma linha de código diversas vezes, sem que o aluno precise escrever mais para

isto. Em resumo, as estruturas de repetição servem para diminuir o custo do programa e deixá-lo o mais simples possível.

Objetivos:

Objetivo geral: Esta atividade tem como objetivo, ensinar aos alunos o conceito de estruturas de repetição, bem como a melhor maneira de utilizá-los.

Objetivos específicos:

Entender o que são estruturas de repetição;

Implementar estruturas de repetição;

Conteúdo Programático:

O que são estruturas de repetição?

Estrutura Para Faça

Estrutura Enquanto Faça

Estrutura Repita Até

Método de ensino/ Desenvolvimento de programa

Os assuntos serão apresentados em aulas expositivas, sempre com discussão e participação dos alunos. Estudos e exercícios, sempre como forma de estimular à participação dos alunos. Aulas práticas em laboratório de Informática.

Resultados esperados: Dos alunos, se espera o entendimento básico sobre como escrever programas que utilizem laços de repetição e o que são, de fato, estas estruturas.

4.2 APLICAÇÃO DAS ATIVIDADES PROPOSTAS

As atividades foram desenvolvidas no período compreendido entre 16/05/2016 a 20/05/2016, em um dos laboratórios de informática da UFSC.

Estas atividades só foram possíveis, devido a colaboração de várias partes, são elas:

- Para escolher a escola participante, foram consideradas algumas variáveis como: deslocamento, número de alunos e a existência do ensino fundamental. A escola que atendeu todos estes requisitos foi a Maria Garcia Pessi, pois se encontra próximo a UFSC Mato Alto, possui séries do ensino fundamental e um grande número de alunos.

- Escolhida a escola, foi necessário agendar uma breve reunião com a diretora da escola para que a mesma autorizasse fazer o convite a estes alunos, bem como, sua participação nas atividades no horário inverso a aula.

- Com o aval da diretora, foi feito o convite para os alunos da 8ª série, 9º ano do ensino fundamental e 1º ano do ensino médio, este convite precisou da assinatura dos responsáveis, e foi novamente entregue a mim.

- Com as datas especificadas, foi feita a reserva do laboratório para os dias de atividades e deu-se o início das mesmas.

Todos os alunos que entregaram os convites devidamente assinados, participaram das atividades. A faixa etária destes alunos variava entre 12 e 16 anos, no turno da manhã haviam onze alunos, com média de idade menor, já no turno da tarde, haviam dezesseis alunos. Isto é, 27 alunos no total, sendo 17 meninos e 10 meninas. Os alunos do 8º, 9º e 1º ano do ensino médio estavam dispostos nas duas turmas, portanto, na turma da manhã, haviam alguns alunos do 8º e 9º ano, e na parte da tarde, haviam alunos do 9º e 1º ano do ensino médio.

Na seção a seguir, serão apresentadas de forma mais detalhadas as atividades desenvolvidas com estes alunos.

ATIVIDADE 1 – CONCEITOS BÁSICOS COMPUTACIONAIS

Em um primeiro momento, conceitos variados sobre computação foram apresentados aos alunos, com o auxílio de uma apresentação power point. Com o objetivo de realizar um apontamento bilateral, os alunos ao serem questionados, discutiam sobre os conceitos que lhe eram familiares e também sobre conceitos que até então, eram desconhecidos.

Dado o conteúdo introdutório, alguns alunos foram convidados a demonstrar no quadro, o que eles entendiam por algoritmo, e também, como deveria ser um algoritmo para “Fritar um ovo”. Feito isto, os demais alunos foram convocados para discutir se os algoritmos criados estavam corretos ou se necessitavam de alguma correção.

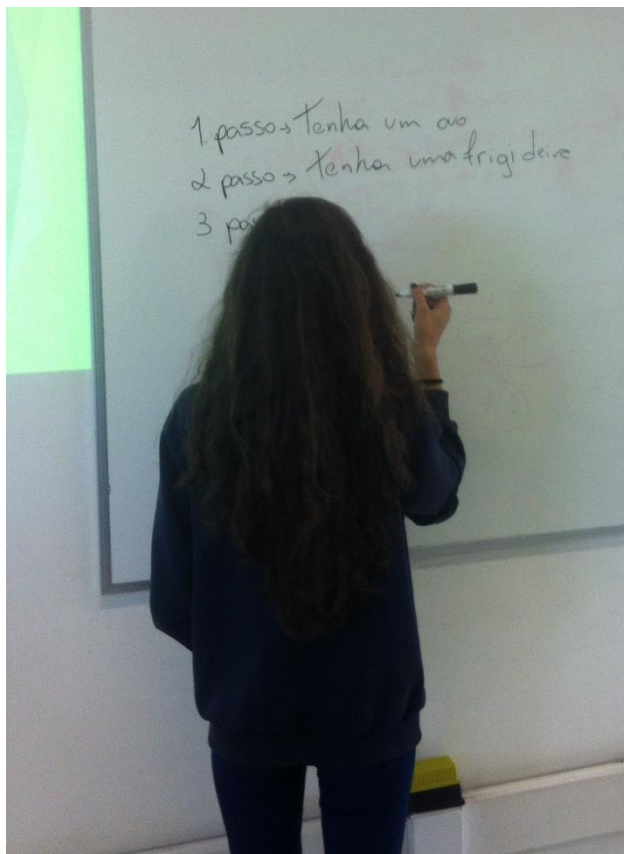


Figura 9: Aplicação da atividade (do autor)

A seguir, as ferramentas escolhidas para os alunos utilizarem foram apresentadas: Scratch e VisuAlg, utilizadas nos turnos da manhã e tarde, respectivamente. Os alunos resolveram alguns exercícios com ajuda, e depois, foram desafiados a resolverem sozinhos. Além disto, alguns alunos eram convidados a responderem perguntas sobre o conteúdo em horas diversas, com o intuito de praticar o que acabara de aprender. Estas perguntas eram feitas por “rodada”, onde uma sequência de números aleatórios foi escrita no quadro (com a ajuda dos alunos) e cada número correspondia a um aluno, portanto, aquele que respondera corretamente à pergunta, só responderia novamente depois que os demais também o fizessem.

Acredita-se que os objetivos desta atividade foram alcançados, pois todos os alunos conseguiram fazer os exercícios de forma individual, bem como, responderem as perguntas propostas.

Após esta primeira atividade, os alunos se mostraram mais desenvoltos e participativos. Inclusive, alguns alunos sugeriram outros algoritmos para serem descritos.

ATIVIDADE 2 – ESTRUTURAS DE SELEÇÃO E OPERADORES

Novamente com o auxílio de uma apresentação prévia sobre o assunto, os alunos resolveram alguns exercícios, desta vez, alguns alunos sentaram em dupla, mas os demais resolveram individualmente. Alguns alunos foram chamados para resolver os exercícios na frente de todos, onde, cada aluno era responsável por pelo menos, uma linha de código. Desta forma, os alunos prestavam atenção, para dar continuidade ao algoritmo, sem perder a lógica. Foi um momento que todos participaram da construção do exercício.



Figura 10: Aplicação da atividade 2(do autor)

No decorrer da atividade, surgiram algumas dúvidas, a proposta era que estas fossem sanadas pelos próprios alunos, para que os mesmos interagissem entre si.

ATIVIDADE 3 – ESTRUTURAS DE REPETIÇÃO

Assim como as atividades anteriores, os alunos tiveram uma apresentação prévia do conteúdo e logo depois foram propostos alguns exercícios para serem resolvidos. Foi sugerido que os mesmos reescrevessem os exercícios anteriores, para adicionar as estruturas de repetição e testar o seu funcionamento. Esta atividade pode ser considerada a mais complexa dada, pois compreende muitos fatores lógicos de programação. E isto pode ser observado, pois foi a atividade que levou mais tempo para ser concluída, além do surgimento de diversas dúvidas.

Mais uma vez, alguns alunos foram convidados a resolver os exercícios na frente de todos, mas o método foi um pouco diferente: Três estruturas de repetição foram apresentadas,

então, para a resolução do exercício proposto, cada aluno escolhido deveria escrever o código utilizando uma das estruturas, sem que houvesse repetição. Desta forma, os discentes conseguiam acompanhar a resolução independente da estrutura escolhida.

Após aplicação de todas as atividades, foi notável a empolgação dos alunos quanto ao término do curso. Embora grande parte lamentou pelo fim das atividades, o sentimento geral era de dever cumprido, pois os alunos assumiram um compromisso de ir, voluntariamente a Universidade, participar de atividades nunca vistas antes, etc. De um modo geral, ambas turmas conseguiram acompanhar o suficiente para resolver uma parcela significativa de exercícios propostos. Considerando que cada um tem um ritmo de aprendizagem, alguns alunos tiveram mais dificuldades do que outros para resolver as atividades individualmente, por conta disto, estes alunos recorriam com mais frequência ao monitor, ou aos colegas para pedir ajuda.

5. RESULTADOS

Neste capítulo serão apresentados os resultados colhidos através da aplicação de um questionário elaborado com 17 questões discursivas e objetivas sobre as atividades realizadas com os alunos e também o detalhamento do mesmo com seus respectivos objetivos.

5.1 QUESTIONÁRIO

O questionário foi desenvolvido em duas partes: A primeira parte conta com questões dissertativas onde o aluno precisou resolver os exercícios relacionados a todas as atividades propostas. A segunda parte conta com questões objetivas sobre o conteúdo dado, e sobre como ele foi dado. Estas questões possuem 5 opções para serem escolhidas. São elas:

- 1- Nada (o aluno acredita que não aprendeu nada com as atividades).
- 2- Insuficiente (o aluno acredita que aprendeu alguma coisa, porém, não o suficiente para resolver os exercícios).
- 3- Suficiente (o aluno acredita que aprendeu o suficiente para resolver boa parte dos exercícios, mas ainda possui alguma dificuldade).
- 4- Muito (o aluno acredita que aprendeu o bastante para resolver todos ou quase todos os exercícios, sem apresentar dificuldade).
- 5- Tudo (o aluno acredita que aprendeu todo o conteúdo que foi trabalhado).

As questões foram desenvolvidas com níveis de dificuldade diferentes, são eles:

Fácil: As questões deste nível são consideradas fáceis por se tratar de conceitos de fácil compreensão, sem que haja a necessidade de um conhecimento prévio, onde o aluno consegue de forma mais livremente, raciocinar e explicar tal conceito.

Médio: As questões deste nível são consideradas com “dificuldade média” por tratarem de conceitos que, para serem entendidos, conceitos anteriores devem ter sido internalizados pelo aluno, mesmo assim, não são complexas.

Difícil: As questões deste nível, foram consideradas desta forma, pois necessitam de um entendimento geral de todos os mecanismos associados ao conteúdo, não basta entender apenas um conceito, é preciso entendê-lo, organizá-lo e aplicá-lo.

Para responder ao questionário, os alunos foram transferidos para uma sala de aula comum na UFSC, e tiveram 3 horas para finalizá-lo. Para cada aluno, foram disponibilizados: caneta, lápis, folha de rascunho e o questionário. Antes de começarem a responder, todas as questões foram lidas em voz alta, individualmente, e discutidas sobre o objetivo da questão, com o intuito de não ocorrer algum problema na interpretação. A medida que os alunos finalizavam o questionário, estes retornavam para o laboratório de informática para que, aqueles que ainda estivessem resolvendo as questões, não fossem prejudicados.

Como dito anteriormente, estas atividades foram realizadas com duas turmas e um total de 27 alunos, sendo 10 meninas e 17 meninos. A seguir, o questionário realizado.

Questão nº 1: Defina com suas palavras: o que é um programa de computador.

Nível de dificuldade: Fácil

Justificativa: Levando em consideração que as ferramentas utilizadas têm como objetivo, o ensino de programação, o aluno, antes de aprender a programar, deve aprender o que é, um programa de computador.

Objetivo: Internalizar o conceito e promover discussão sobre o mesmo.

Turma 1: Scratch – 11 alunos

Quantidade de acertos totais: 4

Quantidade de acertos parciais: 4

Quantidade de erros: 0

Quantidade de brancos: 3

Turma 2: VisuAlg – 16 alunos

Quantidade de acertos totais: 6

Quantidade de acertos parciais: 1

Quantidade de erros: 4

Quantidade de brancos: 5

Questão nº 2: Defina com suas palavras: O que é uma variável, se possível, cite um tipo de dado que pode ser guardado em uma variável.

Nível de dificuldade: Médio

Justificativa: Em um programa de computador, quando é solicitado que se leia ou escreva dados da memória, uma variável deve ser criada. Portanto, as variáveis são uma parte essencial da programação, desta forma, o aluno necessita saber sobre as mesmas para manipulá-las.

Objetivo: Fazer com que os alunos entendam o que é uma variável, através de analogias práticas, para desenvolver programas simples com clareza.

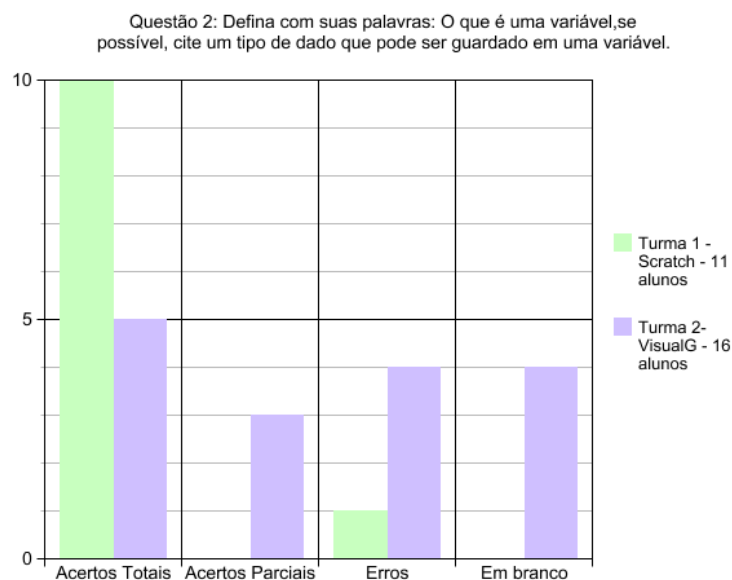


Figura 11: Gráfico da questão 2

Turma 1: Scratch – 11 alunos

Quantidade de acertos totais: 10

Quantidade de acertos parciais: 0

Quantidade de erros: 1

Quantidade de brancos: 0

Turma 2: VisuAlg – 16 alunos

Quantidade de acertos totais: 5

Quantidade de acertos parciais: 3

Quantidade de erros: 4

Quantidade de brancos: 4

Questão nº 3: São tipos de dados de uma variável:

- a) Real, Fimse, Valor
- b) Inteiro, Real, Nome
- c) Caractere, Real, Inteiro
- d) Inteiro, Idade, Nome

Nível de dificuldade: Fácil

Justificativa: O aluno deve entender que existem diferentes tipos de dados que podem ser armazenados em uma variável.

Objetivo: Explicar sobre os diversos tipos de dados, e de qual forma estes devem ser utilizados em um programa de computador.

Turma 1: Scratch – 11 alunos

Quantidade de acertos totais: 10

Quantidade de acertos parciais: 0

Quantidade de erros: 1

Quantidade de brancos: 0

Turma 2: VisuAlg – 16 alunos

Quantidade de acertos totais: 14

Quantidade de acertos parciais: 0

Quantidade de erros: 2

Quantidade de brancos: 0

Questão nº 4: Quais são os comandos para:

- Mostrar uma mensagem na tela requisitando algum dado a ser digitado pelo usuário.
- Capturar (guardar) este dado.

Nível de dificuldade: Médio

Justificativa: Quando um programa precisa que algum dado seja digitado pelo usuário, por exemplo, em um cadastro, dois comandos são utilizados para requisitar e obter este dado, portanto, é crucial que o aluno entenda e saiba quando e como usar estes comandos.

Objetivo: O aluno deve aprender a desenvolver um programa utilizando os comandos de entrada e saída de dados.

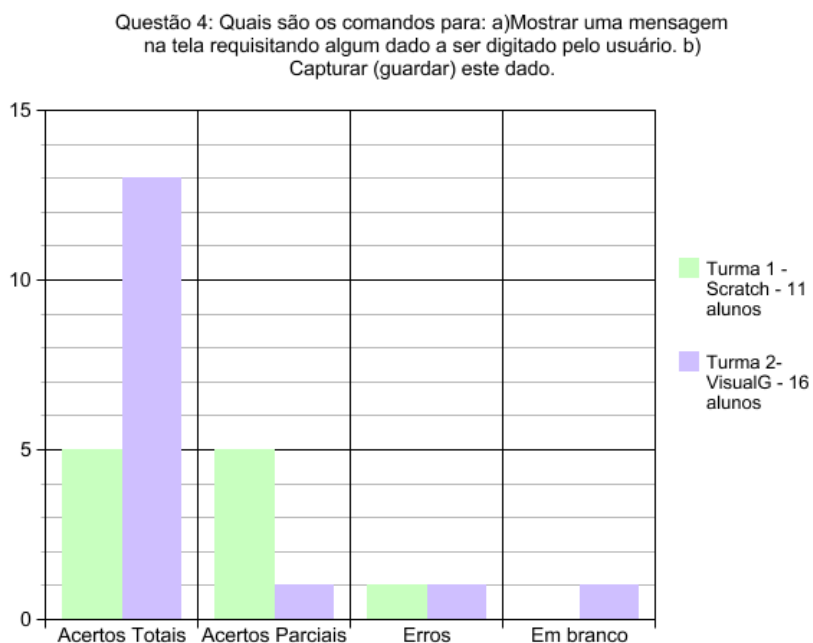


Figura 12: Gráfico da questão 4

Turma 1: Scratch – 11 alunos

Quantidade de acertos totais: 5

Quantidade de acertos parciais: 5

Quantidade de erros: 1

Quantidade de brancos: 0

Turma 2: VisuAlg – 16 alunos

Quantidade de acertos totais: 13

Quantidade de acertos parciais: 1

Quantidade de erros: 1

Quantidade de brancos: 1

Questão nº 5: Escreva como deve ser feita a atribuição do valor 10 a variável X, de tipo inteiro.

Nível de dificuldade: Médio

Justificativa: As variáveis de um programa, podem ter seus valores modificados diversas vezes, para isto, é necessário entender como é feita a atribuição de valor.

Objetivo: O aluno deve entender como atribuir valor a uma variável e o impacto que isto tem, em um programa.

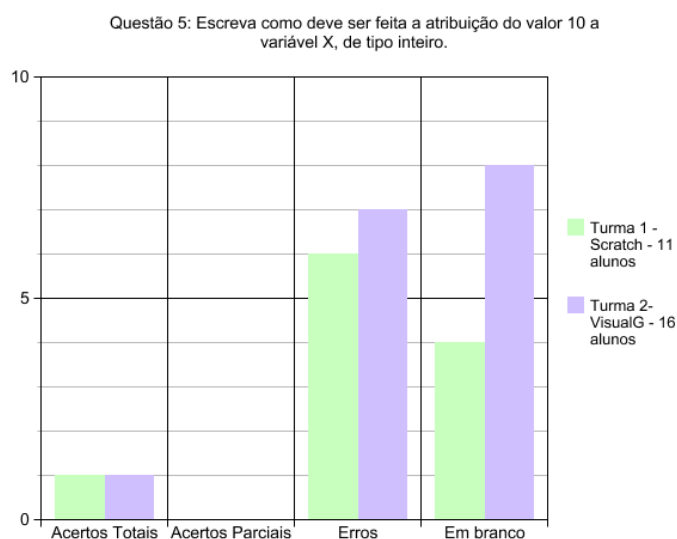


Figura 13: Gráfico da questão 5

Turma 1: Scratch – 11 alunos

Quantidade de acertos totais: 1

Quantidade de acertos parciais: 0

Quantidade de erros: 6

Quantidade de brancos: 4

Turma 2: VisuAlg – 16 alunos

Quantidade de acertos totais: 1

Quantidade de acertos parciais: 0

Quantidade de erros: 7

Quantidade de brancos: 8

Questão nº 6: Faça um programa que peça para o usuário digitar três notas. Considerando as notas digitadas, o programa deve calcular a média do aluno e se esta média for maior ou igual a 7, deve-se mostrar na tela a mensagem “Aluno Aprovado”, se a média for menor que 7, deve-se mostrar na tela a mensagem “Aluno Reprovado”.

Nível de dificuldade: Difícil

Justificativa: As estruturas de seleção servem para otimizar o programa, ao testarem o máximo possível de alternativas com o mínimo de linhas. O que torna essencial para um programa de grande escala.

Objetivo: O aluno deve aprender a manipular variáveis utilizando as estruturas de seleção se, então/senão para resolver problemas onde um trecho de código só será executado se determinada condição incorporar um estado específico (verdadeira ou falsa, por exemplo).

Turma 1: Scratch – 11 alunos

Quantidade de acertos totais: 3

Quantidade de acertos parciais: 5

Quantidade de erros: 0

Quantidade de brancos: 3

Turma 2: VisuAlg – 16 alunos

Quantidade de acertos totais: 4

Quantidade de acertos parciais: 7

Quantidade de erros: 1

Quantidade de brancos: 4

Questão nº 7: Faça um programa que peça para o usuário digitar 2 números quaisquer. O programa deve retornar a soma destes dois números, até que o segundo número digitado seja -1. Deve-se usar a estrutura de repetição: Repita ate.

Nível de dificuldade: Difícil

Justificativa: As estruturas de repetição são essenciais para tornar um programa de computador mais ágil e objetivo, desta forma, o aluno ao implementar estas estruturas em diferentes programas, mostra o entendimento das mesmas.

Objetivo: O aluno deve compreender como realizar as mudanças necessárias em um programa simples para um programa que utilize laços de repetição.

A justificativa e o objetivo da questão 7,8 e 9 são os mesmos.

Turma 1: Scratch – 11 alunos

Quantidade de acertos totais: 3

Quantidade de acertos parciais: 2

Quantidade de erros: 2

Quantidade de brancos: 4

Turma 2: VisuAlg – 16 alunos

Quantidade de acertos totais: 4

Quantidade de acertos parciais: 2

Quantidade de erros: 4

Quantidade de brancos: 6

Questão nº 8: Faça um programa que peça para o usuário digitar uma quantidade indefinida de nomes até que seja digitado “fim”. O programa deve retornar à quantidade de nomes digitados, para isto, deve ser utilizada a estrutura de repetição: Enquanto faça.

Nível de dificuldade: Difícil

Questão 8: Faça um programa que peça para o usuário digitar uma quantidade indefinida de nomes até que seja digitado "fim". O programa deve retornar a quantidade de nomes digitados, para isto, deve ser utilizada a estrutura Enquanto Faça.

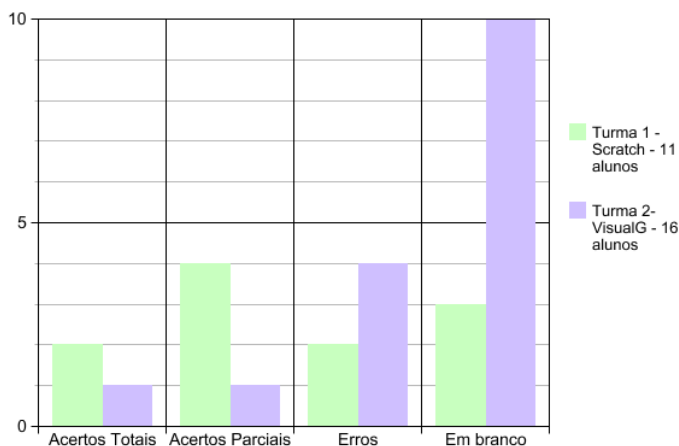


Figura 14: Gráfico da questão 8

Turma 1: Scratch – 11 alunos

Quantidade de acertos totais: 2

Quantidade de acertos parciais: 4

Quantidade de erros: 2

Quantidade de brancos: 3

Turma 2: VisuAlg – 16 alunos

Quantidade de acertos totais: 1

Quantidade de acertos parciais: 1

Quantidade de erros: 4

Quantidade de brancos: 10

Questão nº 9: Faça um programa que mostre na tela a mensagem “Estou programando” dez vezes. Para isto, utilize a estrutura de repetição: Para faça.

Nível de dificuldade: Difícil

Turma 1: Scratch – 11 alunos

Quantidade de acertos totais: 8

Quantidade de acertos parciais: 2

Quantidade de erros: 0

Quantidade de brancos: 1

Turma 2: VisuAlg – 16 alunos

Quantidade de acertos totais: 3

Quantidade de acertos parciais: 0

Quantidade de erros: 5

Quantidade de brancos: 8

Após análise da primeira parte do questionário, referente as questões dissertativas e específicas do conteúdo, pode-se constatar algumas observações. No que diz respeito a porcentagem de acertos e erros, a questão que obteve menos acertos foi a de número 5. Apenas 10% dos alunos da turma da manhã acertaram, já na turma da tarde, apenas 6.25% responderam corretamente. O conteúdo desta questão era o de atribuição de valor em uma variável, este conteúdo foi dado na primeira aula, o que pode ter resultado em poucos acertos, pois o conteúdo não estava tão “fresco”. Em contrapartida, para os demais tópicos como laços de repetição e estruturas de seleção, a atribuição de valores também foi utilizada, inclusive, algumas perguntas do questionário, para que fossem resolvidas corretamente, era necessário atribuir valor em uma variável. Alguns casos como, o aluno acertar uma questão que necessitava atribuir um valor a uma variável, mas errar a questão 5, aconteceram. Portanto, é discutível as seguintes hipóteses:

Hipótese 1: O aluno não entendeu o que pedia a questão.

Hipótese 2: O aluno decorou as questões que envolviam atribuição de valor, e quando foi desafiado a escrever apenas a atribuição, não conseguiu.

Quanto as questões que obtiveram mais acertos, considerando acertos totais e parciais nas duas turmas, as mais pontuadas foram as questões 3 e 4. Cada uma obteve 90% e 87.5% de acertos nas turmas da manhã e tarde, respectivamente. É plausível considerar as seguintes hipóteses:

Hipótese 1: Ambas questões foram repetidamente exercitadas durante as atividades, pois os tópicos: tipos de variáveis e entrada e saída de dados são a base de qualquer programa de computador, o que leva a crer que os alunos compreenderam este conteúdo.

Quanto a porcentagem de respostas em branco, na turma da manhã, 55% das questões tiveram respostas em branco, enquanto que na parte da tarde, 88% das questões tiveram respostas em branco. Isto significa que, os alunos da parte da manhã, tentaram resolver os exercícios mais do que os alunos da tarde. Com isto, é possível cogitar as seguintes hipóteses:

Hipótese 1: A faixa etária dos alunos da manhã era menor do que a dos alunos da tarde, portanto, é razoável dizer que, isto pode se dar ao fato que, o aluno, quando mais novo, tem menos medo de errar, e por isto, se sente motivado a tentar, mesmo que isto não resulte em acerto. Enquanto que o aluno mais velho, prefere nem tentar, do que vir a errar. Aos alunos, foi recomendado que não assinassem seu nome na folha do questionário, para que toda correção fosse feita de modo imparcial, além de deixar os alunos mais confortáveis na hora de responder, principalmente as questões abertas. Mesmo assim, os alunos da parte da tarde, podem ter se sentido constrangidos por não saberem responder à questão inteira, e por isto, preferiram deixá-la em branco.

Como explicado anteriormente, as seguintes questões possuem cinco opções a serem marcadas, elas serão identificadas apenas pela numeração, são elas:

1- Nada, 2- Insuficiente, 3- Suficiente, 4- Muito, 5- Tudo.

Questão nº 10: De um modo geral, o quanto você considera ter aprendido em relação ao conteúdo dado?

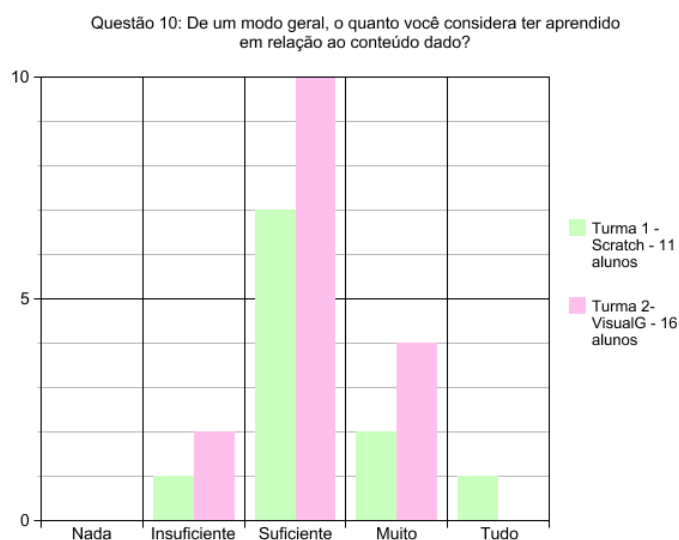


Figura 15: Gráfico da questão 10

Turma 1: Scratch – 11 alunos

- 1- 0
- 2- 1 aluno marcou esta opção
- 3- 7 alunos marcaram esta opção
- 4- 2 alunos marcaram esta opção
- 5- 1 aluno marcou esta opção

Turma 2: VisuAlg – 16 alunos

- 1- 0
- 2- 2 alunos marcaram esta opção
- 3- 10 alunos marcaram esta opção
- 4- 4 alunos marcaram esta opção
- 5- 0

Questão nº 11: O quanto você considera ter aprendido em relação ao conteúdo da Aula 1 (O que é algoritmo, lógica, linguagem de programação, variáveis, tipos de dados, atribuição de dados, entrada e saída de dados)?

Turma 1: Scratch – 11 alunos

- 1- 0
- 2- 0
- 3- 2 alunos marcaram esta opção
- 4- 5 alunos marcaram esta opção
- 5- 2 alunos marcaram esta opção

Obs: Dois alunos não estavam presente nesta atividade.

Turma 2: VisuAlg – 16 alunos

- 1- 0

2- 2 alunos marcaram esta opção

3- 8 alunos marcaram esta opção

4- 4 alunos marcaram esta opção

5- 1 aluno marcou esta opção

Obs: Um aluno não estava presente nesta atividade.

Questão nº 12: O quanto você considera ter aprendido em relação ao conteúdo dado da Aula 2: (Operadores relacionais, operadores lógicos, operadores aritméticos, tabela verdade, estruturas de seleção simples e composta)?

Turma 1: Scratch – 11 alunos

1- 0

2- 2 alunos marcaram esta opção

3- 4 alunos marcaram esta opção

4- 4 alunos marcaram esta opção

5- 1 aluno marcou esta opção

Turma 2: VisuAlg – 16 alunos

1- 0

2- 3 alunos marcaram esta opção

3- 10 alunos marcaram esta opção

4- 3 alunos marcaram esta opção

5- 0

Questão nº 13: O quanto você considera ter aprendido em relação ao conteúdo da Aula 3 (Estruturas de repetição, para faça, enquanto faça, repita até e comando interrompa)?

Turma 1: Scratch – 11 alunos

1- 0

- 2- 1 aluno marcou esta opção
- 3- 7 alunos marcaram esta opção
- 4- 1 aluno marcou esta opção
- 5- 2 alunos marcaram esta opção

Turma 2: VisuAlg – 16 alunos

- 1- 0
- 2- 1 aluno marcou esta opção
- 3- 9 alunos marcaram esta opção
- 4- 5 alunos marcaram esta opção
- 5- 0

Obs: Um aluno não estava presente nesta atividade.

Questão nº 14: Você faria este curso ou algum semelhante novamente?

As opções que poderiam ser marcadas são: 1- Sim, 2- Não, 3- Talvez.

Turma 1: Scratch – 11 alunos

- 1- 9 alunos marcaram esta opção
- 2- 0
- 3- 2 alunos marcaram esta opção

Turma 2: VisuAlg – 16 alunos

- 1- 8 alunos marcaram esta opção
- 2- 2 alunos marcaram esta opção
- 3- 6 alunos marcaram esta opção

Questão nº 15: Você acha que o tempo deste curso foi o suficiente?

As opções que poderiam ser marcadas são: 1- Sim, 2- Sim, mas deveria ser menor, 3- Não, 4- Não, deveria ser maior.

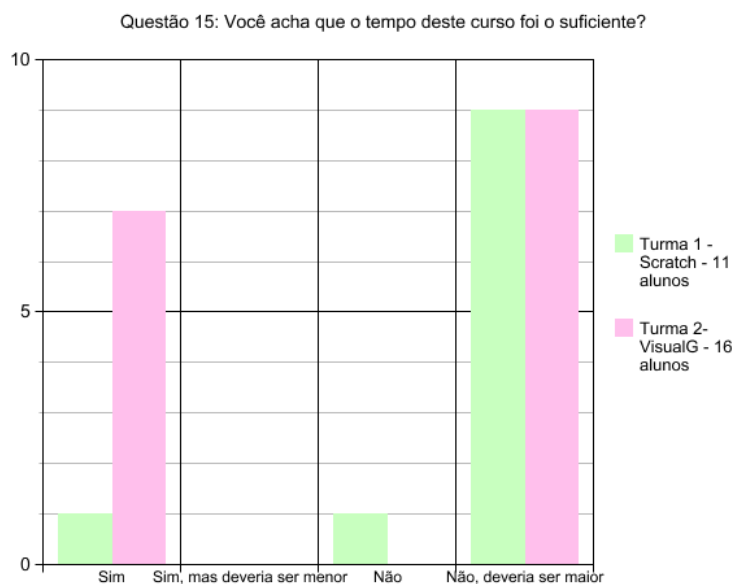


Figura 16: Gráfico da questão 15

Turma 1: Scratch – 11 alunos

1- 1 aluno marcou esta opção

2- 0

3- 1 aluno marcou esta opção

4- 9 alunos marcaram esta opção

Turma 2: VisuAlg – 16 alunos

1- 7 alunos marcaram esta opção

2- 0

3- 0

4- 9 alunos marcaram esta opção

Questão nº 16: Qual sua opinião em relação ao curso como um todo? (Ferramenta escolhida, didática, instrutor)?

As opções que poderiam ser marcadas são: 1- Gostei, 2- Gostei, mas deveria melhorar nos seguintes aspectos, 3- Não gostei, 4- Não gostei, deveria melhorar nos seguintes aspectos:



Figura 17: Gráfico da questão 16

Turma 1: Scratch – 11 alunos

1- 9 alunos marcaram esta opção

2- 2 alunos marcaram esta opção

3- 0

4- 0

Obs: Respostas da opção 2: Ambas são iguais “ O curso deveria ser maior”.

Turma 2: VisuAlg – 16 alunos

1- 13 alunos marcaram esta opção

2- 3 alunos marcaram esta opção

3- 0

4- 0

Obs: Respostas da opção 2:

Aluno 1: “ O curso deveria ser maior”.

Aluno 2: “O curso deveria ser mais longo”.

Aluno 3: “As explicações para nós foram muito rápidas, as vezes não dava pra entender”.

Questão nº 17: Este espaço existe para você escrever críticas, sugestões ou qualquer outra coisa que você ache importante sobre qualquer aspecto do curso.

Turma 1: Scratch – 11 alunos

“Eu adorei o curso, aprendi coisas muito interessantes”.

“Achei muito top e interessante, faria de novo”.

“ Eu só tenho a elogiar, ela foi uma excelente professora, soube a hora de brincar, e a hora de ensinar”.

Turma 2: VisuAlg – 16 alunos

“Achei bom, apenas acho que deveria ser mais completo”.

“ Foi um bom curso, porém pouco tempo, ou seja, curto, mas entendi bem e achei a instrutora admirável e divertida com os alunos”.

“Gostei bastante do modo da instrutora explicar, acho que não precisa mudar nada, está muito bom”.

“Muito obrigado pelo ensinamento”.

“O curso é muito bom, só acho que poderia durar mais que uma semana”.

“Deveria melhorar o tempo, ser um pouco maior, é muito bom (legal) vir pra cá, você é uma ótima instrutora, lembrando que vai dar uma saudade de você e das suas brincadeiras inesperadas”.

“Gostei, ela teve paciência para explicar, sempre fez brincadeiras, algumas vezes se atrapalhava, mas faz parte”.

“Gostei muito, só as vezes achava que a instrutora explicava muito rápido, mas mesmo assim, aprendi bastante. Computadores, sala, ambiente, muito bons”.

“ Gostei muito de fazer o curso, queria que tivesse mais tempo pois assim, poderia aprender mais. Mas mesmo assim, foi muito bom, espero pode fazer mais cursos”.

“ Sobre a instrutora, ela é muito boa, explica bem, mas eu que não consegui entender muito bem, mas agora que eu estava entendendo um pouco, o curso acabou. Então acho que o curso deveria ser mais longo, tipo 1 mês, todos os dias, ou 3 dias por semana. Muito obrigado pela oportunidade”.

Após análise da segunda parte do questionário, obteve-se alguns resultados, e aqueles que são relevantes para o trabalho, serão descritos. Quanto a primeira questão da segunda parte, aproximadamente 63% dos alunos de ambos turnos, acreditam ter aprendido o suficiente de todo o conteúdo dado. Na questão 14, ao serem perguntados sobre a possibilidade de refazer este curso ou semelhante, 81% dos alunos da manhã responderam que sim, fariam novamente, enquanto que na parte da tarde, apenas 50% responderam positivamente à questão. Na pergunta seguinte, sobre a duração do curso, 81% dos alunos do turno da manhã responderam que o tempo não foi o suficiente, e por isto, deveria ter uma duração maior, na parte da tarde 56.2% concordaram com a afirmação anterior. Por fim, 81% dos alunos de ambos turnos responderam que gostaram do curso considerando todos os seus aspectos.

Ainda sobre as questões, algumas inconsistências foram observadas. Exatos 5 alunos da parte da manhã e 10 da parte da tarde, marcaram opções como “Muito”, “Tudo” ou “Suficiente” nas questões 11,12 e 13 principalmente, mas deixaram em branco ou erraram as perguntas de conteúdo correspondente. Exemplos:

Aluno 1: Nas questões 11, 12 e 13, marcou a opção “Suficiente”, porém, deixou as questões correspondentes (6,7,8 e 9) em branco. O que sugere que, diferentemente do que ele marcou, não aprendeu o suficiente para resolver os exercícios.

Aluno 2: Na questão 13, sobre estruturas de repetição, marcou a opção “Muito”, porém, errou as 3 questões correspondentes ao conteúdo. O que sugere que, o aluno não compreendeu o mesmo.

Aluno 3: Na questão 13, marcou a opção “Tudo”, porém, errou duas questões sobre estruturas de repetição.

No total, 45% dos alunos da manhã, responderam as questões com alguma inconsistência lógica, na parte da tarde, esta porcentagem aumentou para 62%. Realizando um senso geral, pode-se afirmar que, os alunos do turno da manhã, obtiveram mais êxito do que os alunos da tarde, levando em consideração o número de acertos, erros e questões em branco. Na parte da manhã, os alunos tiveram mais acertos parciais porque tentaram mais, enquanto que a tarde, os alunos optaram por tentar menos, deixando questões em branco, e assim, resultando em uma quantidade menor, de acertos parciais.

Hipótese 1: A ferramenta utilizada na parte da manhã foi o Scratch, que é uma ferramenta elaborada com uma linguagem visual. Já a ferramenta da parte da tarde foi o VisuAlg, que é uma ferramenta para interpretar códigos, semelhante a um compilador comum. Por ser

uma linguagem visual, o Scratch pode ter sido um facilitador na compreensão de alguns conteúdos mais complexos, como os laços de repetição, pois a ferramenta conta com recursos visuais, como imagens, animações e blocos de cores diferentes que sinalizam suas funções. Por esta razão, os alunos que aprenderam as atividades utilizando esta ferramenta, podem ter se lembrado dos exercícios, puxando de sua memória, as imagens/blocos correspondentes. Já na parte da tarde, os alunos que utilizaram a ferramenta VisuAlg não tem esse recurso, em contrapartida, os códigos são executados em uma tela preta que lembra o *prompt* de comando do sistema operacional Windows, o que também pode facilitar o entendimento, visto que através desta interface é possível acompanhar o que ocorre a cada linha do código.

Para compreender as diferenças das ferramentas, a tabela a seguir foi criada para destacar os principais comandos de programação utilizados e suas correspondências de acordo com cada uma.

COMANDOS	SCRATCH	VISUALG
Entrada de Dados	Pergunte (e espere) resposta	Leia (variável)
Saída de Dados	Diga (expressão) por x segundos	Escreva (variável)
Atribuição de valor	Mude (variável) para 3	Variável <- 3
Operador de diferença	Não (variável/valor)	< > (variável/valor)
Tipos de variáveis e Declaração de variável	Não existem os tipos, não tem restrição de caractere especial (? , _ , *), os nomes das variáveis não podem ser iguais.	Existem os tipos (inteiro, real, caractere, lógico), existe a restrição de caractere especial, isto é, uma variável não pode começar com um número, por exemplo. Os nomes das variáveis não podem ser iguais.

Estruturas de Repetição	Repita, sempre se, repita até	Para faça, enquanto faça, repita até
Parada de um laço	Usa-se o comando “pare tudo” para forçar uma parada no meio do código	Usa-se o comando “interrompa”

Tabela 7: Comparação entre as ferramentas VisuAlg e Scratch

Quanto a observação direta ao longo das atividades com os alunos, foi possível constatar as seguintes coisas:

- Os alunos se sentiram motivados do início ao fim do curso;
- Os alunos começaram a ser mais participativos e desenvolvidos a partir do 2º dia de atividades;
- A preferência geral era a de resolver os exercícios em dupla, devido a isto, é válido afirmar que o desempenho dos alunos quando avaliados em dupla, era melhor do que individualmente;
- Alguns alunos se destacaram, por exemplo, quando requisitados para desenvolver um jogo onde cujo objetivo era de realizar a soma de dois números, algumas duplas tomaram a frente, e programaram não só a soma, mas as demais operações básicas, subtração, divisão e multiplicação;
- Em contrapartida, alguns alunos tinham dificuldade em acompanhar, e mesmo com o auxílio do monitor e dos demais colegas, optavam por não concluir alguns exercícios, mesmo assim, não deixavam de acompanhar a resolução destes, o que sugere que, estes alunos tinham interesse no conteúdo, porém faltava o entendimento do mesmo;
- No turno da manhã, a sala foi composta majoritariamente por meninos, contendo apenas 1 menina, enquanto que na parte da tarde, a turma estava mais balanceada, mas com as meninas em maior número. Particularmente, a turma da manhã foi mais difícil de “conter”, pois em alguns momentos, os meninos estavam muito agitados e se desconcentravam com facilidade. Já na parte da tarde, a turma era mais tranquila, e as meninas se mostraram mais focadas no decorrer das atividades.

6. TRABALHOS RELACIONADOS

Um dos trabalhos identificados pelo seu teor semelhante a este, foi o desenvolvido em 2014 intitulado “Ensino de lógica de programação no ensino fundamental utilizando o Scratch: um relato de experiência”, onde a ferramenta Scratch foi utilizada para o ensino de conceitos básicos computacionais para uma turma do 9º ano do ensino fundamental. Desenvolvido através de um projeto de extensão da Universidade de Pernambuco, os alunos criaram jogos utilizando comandos como laços de repetição e condição, segundo Oliveira et al. (2014). Os pontos em comum com este trabalho foram o público alvo, estudantes do ensino fundamental de uma escola pública, a metodologia utilizada, isto é, em um primeiro momento foi dada a introdução aos conceitos de programação, e depois os alunos desenvolveram na ferramenta.

Outro trabalho relacionado, elaborado por Bini e Koscianski (2009), tem como proposta relatar o experimento feito com a ferramenta Scratch para ensinar programação para os anos iniciais do ensino técnico em programação. Os pontos em comum foram a metodologia empregada, através de exercícios de fixação, e a ferramenta escolhida. Em contrapartida, o presente trabalho foi desenvolvido com turmas do ensino fundamental que não possuíam conhecimento prévio em programação, enquanto que o trabalho produzido por Bini e Koscianski (2009) as atividades foram feitas com alunos que já conheciam conceitos básicos como variáveis, comandos de entrada e saída, laços de repetição e estruturas de seleção.

Quanto a análise de ferramentas, o trabalho criado por Sena et al. (2015) realizou uma abordagem mais complexa onde o mesmo categorizou as ferramentas em diferentes níveis, de acordo com seu objetivo/gênero. Estes níveis foram divididos em 3 tipos: Tipo 1, movimentação de *avatares* através de um programa; Tipo 2, Avaliação imediata do código produzido; Tipo 3, Criação de *avatares* autônomos. Enquanto que o presente trabalho, analisou as ferramentas disponíveis comparando-as através de suas características como disponibilidade, conteúdo e idioma.

Uma outra abordagem semelhante, foi produzida com alunos de Licenciatura em Matemática, onde, o objetivo era o de “contextualizar o ensino de lógica e algoritmos por meio da programação de jogos educacionais” (FABRI, 2008). Os jogos, foram desenvolvidos com o auxílio da linguagem LOGO, para promover o aprendizado em um ambiente criativo, e se possível, mais eficaz. Os pontos em comum encontrados foram a metodologia de ensino, utilizando ferramentas não tradicionais, e a proposta de exercícios e desafios. Sobre o método de avaliação, o presente trabalho desenvolveu um questionário contendo questões sobre o conteúdo, e a partir dos resultados obtidos, pode-se observar o desempenho dos estudantes em relação as atividades propostas. Enquanto que o trabalho relacionado, avaliou os estudantes através das notas obtidas com as provas e trabalhos da disciplina de lógica de programação, que obteve um aumento favorável no desempenho acadêmico, quando em 2006, 50% dos alunos não conseguiram alcançar a nota 6 nas avaliações, já em 2007, apenas 35% deles não obtiveram este mínimo.

A partir dos trabalhos analisados, é possível afirmar que, a temática “Ensino de Programação” é relativamente atual, se comparada a outros temas computacionais que exploram o histórico. Além disto, o tema é relevante, pois busca diferentes métodos para que os alunos consigam compreender de forma mais eficaz, a programação. Percebe-se que há uma dificuldade proeminente no que diz respeito a disciplina, e isto se estende a todos os níveis de educação, como é possível perceber nos trabalhos citados. Por conta disto, a pesquisa para se descobrir meios facilitadores de aprendizagem é um fator importantíssimo para aperfeiçoar o ensino de programação.

7. CONSIDERAÇÕES FINAIS

Neste trabalho, foi apresentada a aplicação de duas ferramentas de ensino de programação em turmas de alunos provenientes do ensino fundamental de uma escola pública, com o objetivo de entender os motivos pelo qual o ensino de programação nas escolas, não seja uma prática mais comum, além de constatar através dos resultados obtidos com um questionário, qual ferramenta é a mais adequada para ensinar os conceitos básicos iniciais da programação.

No decorrer das atividades propostas, pode-se observar o entusiasmo dos alunos ao lidarem com as ferramentas, bem como, a criatividade que os mesmos aplicavam na resolução dos exercícios. Levando em consideração que a programação é flexível quanto as respostas certas, diferentes soluções para o mesmo problema foram criadas, o que representa um bom desempenho do raciocínio lógico.

As turmas foram separadas por turno e ferramenta, fazendo uma avaliação geral, ambas gostaram do curso, mas ficaram insatisfeitas quanto ao tempo de duração. As atividades duraram 4 dias, com duração de 3 horas e 20 minutos, totalizando aproximadamente 14 horas semanais. Por esta razão, muitos sugeriram que o curso deveria ser estendido para 1 mês, com mais horas por dia.

Após avaliar as respostas do questionário, é possível afirmar que, para este caso, em específico, a ferramenta Scratch se mostrou mais satisfatória para ser utilizada como ferramenta de apoio ao ensino de programação, pois, os alunos que utilizaram-na tiveram uma porcentagem de acertos maior, do que os alunos que trabalharam com a ferramenta VisuAlg, além disto, estes tentaram solucionar a maior parte dos exercícios, enquanto que os alunos que utilizaram o VisuAlg, deixaram boa parte das questões em branco.

Ao realizar uma busca sobre o tema proposto, foi possível encontrar vários trabalhos relacionados, porém, recentes. O que indica que ensinar programação para crianças e adoles-

centes é uma prática relativamente nova e não tão desenvolvida, uma justificativa para o tal é a falta de conhecimento na disciplina por parte dos docentes que não são da área, ou até mesmo na dificuldade em não saber como aplicar a programação, nas disciplinas diárias como português e matemática.

Nos tópicos a seguir, são apresentadas algumas propostas para aplicação em trabalhos futuros:

- Aplicação de ferramentas para o ensino de programação com idosos;
- Aplicação de ferramentas para o ensino de programação com alunos da 1ª fase de cursos voltados a computação;
- Aplicação de ferramentas para o ensino de programação com crianças das séries iniciais;
- Aplicação de ferramentas que utilizem linguagem de programação (C, Java, Delphi) com alunos do ensino médio;

8. REFERENCIAS

ADAMI, Anna. **Codecademy**. Disponível em:

<<http://www.infoescola.com/informatica/codecademy/>>. Acesso em: 07 fev. 2016.

ALMEIDA, Cíntia Soares de. **Dificuldades de aprendizagem em Matemática e a percepção dos professores em relação a fatores associados ao insucesso nesta área**. 2006. 13 f. TCC (Graduação) - Curso de Matemática, Universidade Católica de Brasília – Ucb, Brasília, 2006. Cap. 2. Disponível em:

<<https://www.ucb.br/sites/100/103/TCC/12006/CinthiaSoaresdeAlmeida.pdf>>. Acesso em: 07 fev. 2016.

ALVAREZ, Luciana. **Ensino de programação é aposta de colégios em todo o mundo: A linguagem ajuda no raciocínio lógico e na criatividade, além de facilitar o aprendizado de disciplinas como o português e a matemática**. 2014. Disponível em:

<<http://revistaeducacao.uol.com.br/textos/211/aposta-no-futuroo-ensino-de-programacao-tem-se-espalhado-como-330266-1.asp>>. Acesso em: 07 fev. 2016.

AIVES, A. A. e COSTA, M. R. (2004) “**algolPUC: Uma Ferramenta de Apoio à Programação para Iniciantes**”, III Workshop de Educação em Computação e Informática do estado de Minas Gerais (WEIMIG’ 2004). Belo Horizonte, MG, Brasil.

BELLONI, Maria Luiza. **Adolescentes com idade entre 15 e 19 anos são os que mais acessam internet no Brasil, diz IBGE**.2015. Disponível em:

<http://www.brasilpost.com.br/2015/04/29/acesso-internet-2013_n_7165436.html>. Acesso em: 07 fev. 2016.

BINI, Elena Mariele; KOSCIANSKI, André. **O ENSINO DE PROGRAMAÇÃO DE COMPUTADORES EM UM AMBIENTE CRIATIVO E MOTIVADOR**. In: ENCONTRO NACIONAL DE PESQUISA EM EDUCAÇÃO EM CIÊNCIAS, 7., 2009, Florianópolis.

lis. **Artigo**. Florianópolis: Enpec, 2009. p. 1 - 11. Disponível em:

<<http://posgrad.fae.ufmg.br/posgrad/viiienpec/pdfs/675.pdf>>. Acesso em: 06 nov. 2016.

BRASSCOM, Associação Brasileira das Empresas de Tecnologia da Informação e Comunicação. **Índice Brasscom de Convergência Digital - IBCD**. 6. ed. São Paulo: Brasscom, 2012. 104 p.

CABRAL, Ronaldo Vieira; ALENCAR, Anderson Fernandes de. **O ENSINO DE MATEMÁTICA E A INFORMÁTICA: USO DO SCRATCH COMO FERRAMENTA PARA O ENSINO E APRENDIZAGEM DA GEOMETRIA**. 2015. 12 f. Dissertação (Mestrado) - Curso de Matemática, Universidade Federal Rural de Pernambuco, Ufrpe, Pernambuco, 2015. Cap. 4. Disponível em:

<http://www.editorarealize.com.br/revistas/conedu/trabalhos/TRABALHO_EV045_MD1_SA8_ID1488_08092015105409.pdf>. Acesso em: 07 fev. 2016.

CODECOMBAT. **Code Combat: O que é**. Disponível em:

<<http://br.codecombat.com/about#contact>>. Acesso em: 21 set. 2016.

COOPER, Stephen; DANN, Wanda; PAUSCH, Randy. **ALICE: A 3-D TOOL FOR INTRODUCTORY PROGRAMMING CONCEPTS**. Disponível em:

<<http://www.alice.org/publications/AToolForIntroProgramming.PDF>>. Acesso em: 21 set. 2016.

CROCHIK, José Leon. **O computador no ensino e a limitação da consciência**. São Paulo: Casa do Psicólogo, 1998. 189 p. 189 f. Disponível em:

<<https://books.google.com.br/books?id=mhKnKJS-1V4C&printsec=frontcover&hl=pt-BR#v=onepage&q&f=false>>. Acesso em: 22 fev. 2016.

CTRL+KIDS. **SOBRE A CTRL+KIDS**. Disponível em: <<http://ctrlkids.com.br/sobre-a-ctrlkids/>>. Acesso em: 08 fev. 2016.

DANTAS, Ricardo Fidelis; COSTA, Francisco Eudes Almeida da. **CODE: O ensino de linguagens de programação educativas como ferramentas de ensino/aprendizagem**. In: SIMPÓSIO HIPERTEXTO E TECNOLOGIAS NA EDUCAÇÃO, 5., 2013, Recife, p. 1 - 14. Disponível em: <<http://www.nehte.com.br/simposio/anais/Anais-Hipertexto-2013/CODE - O ensino de linguagens de programação educativas como ferramentas de ensino-aprendizagem.pdf>>. Acesso em: 18 nov. 2016.

DETERS, Janice Inês et al. **O Desafio de Trabalhar com Alunos Repetentes na Disciplina de Algoritmos e Programação**. Florianópolis, p.1-10. Disponível em:

<[http://proativa.virtual.ufc.br/sbie/CD_ROM_COMPLETO/workshops/workshop2/O Desafio de Trabalhar com Alunos Repetentes na.pdf](http://proativa.virtual.ufc.br/sbie/CD_ROM_COMPLETO/workshops/workshop2/O%20Desafio%20de%20Trabalhar%20com%20Alunos%20Repetentes%20na.pdf)>. Acesso em: 18 nov. 2016.

DU, Jie; WIMMER, Hayden; RADA, Roy. **“Hour of Code”: Can It Change Students’ Attitudes toward Programming?** Journal Of Information Technology Education: Innovations in Practice. California, p. 52-73. mar. 2016. Disponível em:

<<http://www.jite.org/documents/Vol15/JITEv15IIPp053-073Du1950.pdf>>. Acesso em: 19 nov. 2016.

FABRI, José Augusto. **O Ensino de Lógica de Programação e o Desenvolvimento de Jogos Educacionais: Um Caso Aplicado aos Alunos do Curso de Licenciatura Plena em Matemática**. São Paulo: Faculdade de Tecnologia de Ourinhos, 2008. 8 p. Disponível em:

<<https://engenhariasoftware.files.wordpress.com/2008/04/jogologo.pdf>>. Acesso em: 06 nov. 2016.

G1, Portal de Notícias. **Alunos do ES têm dificuldade em leitura e matemática, aponta MEC**. 2015. Disponível em: <<http://g1.globo.com/espirito-santo/educacao/noticia/2015/09/alunos-do-es-tem-dificuldade-em-leitura-e-matematica-aponta-mec.html>>. Acesso em: 07 fev. 2016.

GENIOAZUL. **Missão**. Disponível em: <<http://genioazul.com.br/>>. Acesso em: 08 fev. 2016.

GERALDES, Wendell Bento. **PROGRAMAR é BOM PARA AS CRIANÇAS? UMA VISÃO CRÍTICA SOBRE O ENSINO DE PROGRAMAÇÃO NAS ESCOLAS**. Texto Livre: Linguagem e Tecnologia. Minas Gerais, p. 105-117.2014.

GIRLSWHOCODE. **Mission & Vision**. Disponível em: <<http://girlswhocode.com/about-us/>>. Acesso em: 08 fev. 2016.

GOMES, Anabela de Jesus. **Ambiente de suporte à aprendizagem de conceitos básicos de programação**. 2000. 129 f. Dissertação (Mestrado) - Curso de Engenharia Informática, Departamento de Engenharia Informática, Universidade de Coimbra, Coimbra, 2000. Disponível em: <http://files.isec.pt/DOCUMENTOS/SERVICOS/BIBLIO/teses/Tese_Mest_Anabela-Gomes.pdf>. Acesso em: 30 ago. 2016.

GOMES, Tancicleide C. S. et al. **Avaliação de um Jogo Educativo para o Desenvolvimento do Pensamento Computacional na Educação Infantil**. In: CONGRESSO BRASILEIRO

DE INFORMÁTICA NA EDUCAÇÃO, 4., 2015, Maceió. Anais... . Maceió: Cbie-laclo, 2015. p. 1349 - 1358.

GOUWS, Lindsey; BRADSHAW, Karen; WENTWORTH, Peter. **Computational Thinking in Educational Activities: An evaluation of the educational game Light-Bot**. In: ACM CONFERENCE ON INNOVATION AND TECHNOLOGY IN COMPUTER SCIENCE EDUCATION, 13., 2013, Canterbury. Proceeding. New York: Acm, 2013. p. 10 - 15. Disponível em: <<http://dl.acm.org/citation.cfm?id=2466518>>. Acesso em: 19 nov. 2016.

GROS, Begõna. **The impacto of digital games in education**. First Monday, Chicago, v.8, n.7,jun.2003. Disponível em:<http://www.mackenty.org/images/uploads/impact_of_games_in_education.pdf>. Acesso em: 11 set. 2016.

HAPPYCODE. **SOBRE NÓS**. Disponível em: <<http://www.happycode.com.br/>>. Acesso em: 08 fev. 2016.

JENKINS, T. 2002. **On the difficulty of learning to program**. In Proceedings of the 3rd Annual Conference of the LTSN Centre for Information and Computer Sciences, 53–58.

KIDSCANCODE. **Our Mission**. 2015. Disponível em: <<http://kidscancode.org/>>. Acesso em: 08 fev. 2016.

KIDS, Real Programming 4. **Canada's #1 Computer Programming Camps 4 Kids**. Disponível em: <<http://www.realprogramming.com/>>. Acesso em: 08 fev. 2016.

LIMA, Jennifer; TEICHRIEB, Veronica; KELNER, Judith. **Alice: uma Ferramenta de Autoria**: Universidade Federal de Pernambuco, Centro de Informática. 2004. Disponível em: <https://www.gprt.ufpe.br/grvm/wcontent/uploads/Publication/Books&Chapters;/2004/Alice_RealidadeVirtual_UmaAbordagemPratica.pdf>. Acesso em: 21 set. 2016.

MADCODE. **Quem somos**. 2015. Disponível em: <<http://madcode.com.br/>>. Acesso em: 08 fev. 2016.

MADEWITHCODE. **About**. 2015. Disponível em: <<https://www.madewithcode.com>>. Acesso em: 08 fev. 2016.

MARRON, Assaf; WEISS, Gera; WIENER, Guy. **A decentralized approach for programming interactive applications with JavaScript and blockly**. In: PROGRAMMING SYSTEMS, LANGUAGES AND APPLICATIONS BASED ON ACTORS, AGENTS, AND DECENTRALIZED CONTROL ABSTRACTIONS, 2., 2012, Arizo-

na. **Proceeding**. ACM,v.1,p.59-70. Disponível em:

<<http://dl.acm.org/citation.cfm?id=2414648>>. Acesso em: 21 set. 2016.

MEC, Portal. **ProInfo - Apresentação**. 2015. Disponível em:

<<http://portal.mec.gov.br/proinfo/proinfo>>. Acesso em: 07 fev. 2016.

MEDEIROS FILHO, Dante Alves; MOURA, Ernani Guilherme Groff. **A Metodologia de Ensino da Khan Academy para a Área Tecnológica**. Universidade Estadual de Maringá- Departamento de Informática. Disponível em:

<[http://www.espweb.uem.br/site/files/tcc/2011/Ernan Guilherme Groff Moura - A metodologia de ensino da Khan Academy para a area tecnologica.pdf](http://www.espweb.uem.br/site/files/tcc/2011/Ernan%20Guilherme%20Groff%20Moura%20-%20A%20metodologia%20de%20ensino%20da%20Khan%20Academy%20para%20a%20area%20tecnologica.pdf)>. Acesso em: 19 nov. 2016.

MIT, Massachusetts Institute Technology -. **MIT FACTS: MISSION**. 2016. Disponível em:

<<http://web.mit.edu/facts/mission.html>>. Acesso em: 07 fev. 2016.

OLHAR DIGITAL. **Code Club Brasil: nova plataforma de programação para crianças**. 2014. Disponível em: <<http://olhardigital.uol.com.br/noticia/code-club-brasil-nova-plataforma-de-programacao-para-criancas/44102>>. Acesso em: 07 fev. 2016.

OLHAR DIGITAL. **Escolas brasileiras ensinam programação**. 2013. Disponível em:

<<http://olhardigital.uol.com.br/video/escolas-brasileiras-ensinam-programacao/39115>>. Acesso em: 07 fev. 2016.

OLIVEIRA, Millena Lauyse Silva de et al. **Ensino de lógica de programação no ensino fundamental utilizando o Scratch: um relato de experiência**. In: XXXIV CONGRESSO DA SOCIEDADE BRASILEIRA DE COMPUTAÇÃO, 34., 2014, Brasília. **Artigo**. Brasília: Csbcc, 2014. p. 1525 - 1533. Disponível em:

<<http://www.lbd.dcc.ufmg.br/colecoes/wei/2014/0022.pdf>>. Acesso em: 06 nov. 2016.

ORO, Neuza Terezinha et al. **Programação de Computadores e Matemática: potencializando a aprendizagem**. In: CONFERÊNCIA INTERAMERICANA DE EDUCAÇÃO MATEMÁTICA, 14., 2015, México. **Proceeding**. México: Ciaem, 2015. p. 1 - 12. Disponível em: <http://xiv.ciaem-redumate.org/index.php/xiv_ciaem/xiv_ciaem/paper/viewFile/683/313>. Acesso em: 08 set. 2016.

PEZZINI, Clenilda Cazarin; SZYMANSKI, Maria Lidia Sica. **FALTA DE DESEJO DE APRENDER: Causas e Consequências**. Disponível em:

<<http://www.diaadiaeducacao.pr.gov.br/portals/pde/arquivos/853-2.pdf>>. Acesso em: 18 nov. 2016.

POCRIFKA, Dagmar Heil; SANTOS, Taís Wojciechowski. **LINGUAGEM LOGO E A CONSTRUÇÃO DO CONHECIMENTO**. In: CONGRESSO NACIONAL DE EDUCAÇÃO - EDUCERE, 9., 2009, Curitiba. **Artigo**. Curitiba: Congresso Nacional de Educação - Educere, 2009. p. 2470 - 2479. Disponível em:

<http://www.pucpr.br/eventos/educere/educere2009/anais/pdf/2980_1303.pdf>. Acesso em: 07 fev. 2016.

PRADO, Maria Elisabette B. B.. **LOGO - LINGUAGEM DE PROGRAMAÇÃO E AS IMPLICAÇÕES PEDAGÓGICAS**. Brasília: Proinfo, 1999. (Informática para Mudança na Educação). Disponível em:

<http://www.nied.unicamp.br/oea/mat/LOGO_IMPLICACOES_bette_nied.pdf>. Acesso em: 21 set. 2016.

PROGRAMADOR, Jovem. **Descrição**. 2015. Disponível em:

<<http://jovemprogramador.org/>>. Acesso em: 08 fev. 2016.

QUEIRÓS, Ricardo. **Innovative Teaching Strategies and New Learning Paradigms in Computer Programming**. Portugal: Igi Global, 2014. 313 p.

REPENNING, Alexander et al. **Scalable Game Design: A Strategy to Bring Systemic Computer Science Education to Schools through Game Design and Simulation Creation**. 2014. University of Colorado Boulder. Disponível em:

<https://sgd.cs.colorado.edu/wiki/images/4/41/TOCE_2015_Repenning.pdf>. Acesso em: 19 nov. 2016.

RESNICK, Mitchel. **Vamos ensinar crianças a escrever códigos**. Usa: Tedxbeaconstreet, 2012. Son., color. Disponível em:

<https://www.ted.com/talks/mitch_resnick_let_s_teach_kids_to_code?language=pt-br>. Acesso em: 21 set. 2016.

RIBEIRO, Maria Thereza Pillon. **Inclusão digital e cidadania**. Artigo. UNESP, São Paulo. Disponível em: <<http://www2.faac.unesp.br/blog/obsmidia/files/Maria-Thereza-Pillon-Ribeiro.pdf>>. Acesso em: 18 nov. 2016.

SENA, Alexandre C. et al. **Avaliação e Taxonomia de Jogos para Ensino de Programação de Computadores**. In: SBGAMES, 14., 2015, Teresina. **Artigo**. p. 1 - 10.

SHAPIRO, Jordan. **Tech: President Obama Wants Every Kid To Learn Coding--For All The Wrong Reasons**. Forbes, New York, v. 1, n. 1, p.1-2, jan. 2016. Disponível

em: <<http://www.forbes.com/sites/jordanshapiro/2016/01/31/president-obama-wants-every-kid-to-learn-coding-for-all-the-wrong-reasons/#6332d3167b25>>. Acesso em: 19 nov. 2016.

SLANY, Wolfgang. **Catroid: a mobile visual programming system for children**. In: INTERNATIONAL CONFERENCE ON INTERACTION DESIGN AND CHILDREN, 11., 2012, Gerny. **Proceeding**. Gerny: Idc, 2012. p. 300 - 303. Disponível em:

<<http://dl.acm.org/citation.cfm?id=2307151>>. Acesso em: 08 set. 2016.

SOUZA, Clarisse Sieckenius de. **Scalable Game Design Brasil**. PUC -Rio - Departamento de Informática. Disponível em: <<http://www2.serg.inf.puc-rio.br/docs/sgd-br/ScalableGameDesign-Brasil-Summary.pdf>>. Acesso em: 19 nov. 2016.

SOUZA, Cláudio Morgado de. VisuAlg - **Ferramenta de Apoio ao Ensino de Programação**. **Revista Teccen**, Rio de Janeiro, v. 2, n. 2, p.1-9, set. 2009. Disponível em:

<<http://www.uss.br/pages/revistas/revistateccen/V2N22009/ArtigoVisuAlgSOUZA.pdf>>.

Acesso em: 22 set. 2016.

STROUSTRUP, Bjarne. **Programming: Principles and Practice Using C++**. 2. ed.

Crawfordsville: Pearson Education, Inc., 2014. 137 p. Disponível em:

<<http://ptgmedia.pearsoncmg.com/images/9780321992789/samplepages/9780321992789.pdf>

>. Acesso em: 18 nov. 2016.

SUPERGEEKS. **QUEM SOMOS**. 2016. Disponível em:

<<http://campinas.supergeeks.com.br/quem-somos/>>. Acesso em: 08 fev. 2016.

TAROUCO, Liane Margarida Rockenbach et al. Jogos educacionais. **Revista Renote: Novas Tecnologias na Educação**, Porto Alegre, v. 2, n. 1, p.1-7, 1 mar. 2004. Disponível em:

<<http://seer.ufrgs.br/index.php/renote/article/view/13719/8049>>. Acesso em: 29 fev. 2016.

TYNKER. **About Tynker**. 2015. Disponível em: <<https://www.tynker.com/hour-of-code/>>.

Acesso em: 08 fev. 2016.

WECANCODEIT. **About us**. Disponível em: <<https://wecancodeit.org/>>. Acesso em: 08 fev. 2016.

9. APÊNDICE

Apêndice A: Termo de autorização de imagem e participação

TERMO DE AUTORIZAÇÃO DE USO DE IMAGENS E DEPOIMENTOS

Eu, inscrito (a) no CPF sob nº, portador(a) da Cédula de Identidade nº, responsável legal pela(o) menor, portadora da Cédula de Identidade nº, depois de conhecer e entender que o objetivo geral do projeto de extensão que será utilizado para o Trabalho de Conclusão de Curso, para se obter o Bacharelado em Tecnologias da Informação e Comunicação, intitulado "Análise de diferentes metodologias para o ensino de programação " é aplicar a ferramenta de ensino de programação, Visualg em turmas do 8º e 9º ano do ensino fundamental da Escola Básica Profa. Maria Garcia Pessi no período compreendido entre 16/05/2016 e 21/05/2016, no horário entre 13:30 e 16:50 e local: Laboratório de Informática 207 do Campus UFSC Araranguá – Cidade Alta

AUTORIZO através do presente termo, os pesquisadores responsáveis pelo projeto acima descrito, a realizar as fotos que se façam necessárias, a gravação de vídeos e a colher depoimentos da menor supracitada, bem como a veiculação de sua imagem e depoimentos em qualquer meio de comunicação para fins didáticos, de pesquisa e divulgação de conhecimento científico, a elaboração de produtos e divulgação de projetos audiovisuais sem quaisquer ônus financeiros e restrições a nenhuma das partes. Fica ainda autorizada, de livre e espontânea vontade, para os mesmos fins, a concessão de direitos da veiculação das imagens, vídeos e depoimentos da menor supracitada, não recebendo para tanto qualquer tipo de remuneração. Ao mesmo tempo, libero a utilização destas fotos e/ou depoimentos para fins científicos e de estudos (livros, artigos, vídeos, slides e transparências), em favor dos projetos da pesquisa, acima especificados, obedecendo ao que está previsto nas Leis que resguardam os direitos das crianças e adolescentes (Estatuto da Criança e do Adolescente - ECA, Lei no 8.069/1990), e das pessoas com deficiência (Decreto no 3.298/1999, alterado pelo Decreto no 5.296/2004).

Araranguá, maio de 2016.

Apêndice B: Carta Convite à direção da escola Maria Garcia Pessi



UNIVERSIDADE FEDERAL DE SANTA CATARINA CAMPUS ARARANGUÁ

Prezados,

Eu, Amanda Irizaga, aluna do curso de Tecnologias da Informação e Comunicação, com a coordenação da Professora Eliane Pozzebon estou desenvolvendo meu Trabalho de Conclusão de Curso intitulado como “Análise de diferentes metodologias para o ensino da programação”. Por conta disto, um dos requisitos deste trabalho é a aplicação de duas ferramentas voltadas a programação em turmas do ensino fundamental de alguma escola da região, dito isto, venho por meio deste documento, convidar a escola para que me conceda os alunos para aplicação das ferramentas.

Esta aplicação acontecerá no laboratório de informática nas dependências da UFSC – Campus Mato Alto, no horário inverso as aulas regulares da escola, isto é, os alunos que estudam pela manhã, utilizarão a ferramenta à tarde, e os alunos que estudam a tarde, utilizarão a ferramenta pela manhã.

A duração deste projeto será de uma semana, com as datas compreendidas entre 16/05/2016 e 21/05/2016, com horários definidos entre 08:30 e 11:50 (para a primeira turma) e 13:30 até 16:50 (para a segunda turma).