



**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE GOIÁS
CAMPUS DE INHUMAS**

**Coordenação de Informática
Bacharelado em Informática**

Pedro Henrique Barbosa Rodrigues
Leandro Henrique Correia

**COMPARAÇÃO ENTRE FERRAMENTAS DE ENSINO DE PROGRAMAÇÃO ORIENTADA A
BLOCOS. MIT APP INVENTOR VS SCRATCH**

Inhumas - GO
2017

Pedro Henrique Barbosa Rodrigues
Leandro Henrique Correia

**COMPARAÇÃO ENTRE FERRAMENTAS DE ENSINO DE PROGRAMAÇÃO ORIENTADA A
BLOCOS. MIT APP INVENTOR VS SCRATCH**

Monografia apresentada ao Instituto Federal de
Educação, Ciência e Tecnologia de Goiás,
Campus Inhumas, como requisito parcial para
obtenção do título em Bacharel em Informática.
Orientador: Profº. Me. Alexandre Bellezi José.

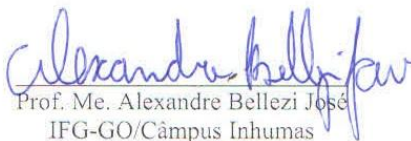
Inhumas - GO
2017

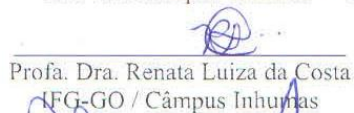
LEANDRO HENRIQUE CORREIA
PEDRO HENRIQUE BARBOSA RODRIGUES

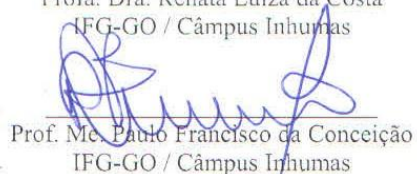
COMPARAÇÃO ENTRE FERRAMENTAS DE ENSINO DE PROGRAMAÇÃO
ORIENTADA A BLOCOS. MIT APP INVENTOR VS SCRATCH

Monografia apresentada à banca examinadora da Coordenação de Bacharelado em
Informática do Instituto Federal de Educação, Ciência e Tecnológica de Goiás – Câmpus
Inhumas, como parte dos requisitos para obtenção do título de Bacharel em Informática.

COMISSÃO EXAMINADORA


Prof. Me. Alexandre Bellezi José
IFG-GO/Câmpus Inhumas


Profa. Dra. Renata Luiza da Costa
IFG-GO / Câmpus Inhumas


Prof. Me. Paulo Francisco da Conceição
IFG-GO / Câmpus Inhumas

Inhumas, 23 de agosto de 2017.

Dados Internacionais de Catalogação da Publicação (CIP)

C824 Correia, Leandro Henrique
 Comparação entre ferramentas de ensino de programação orientada a blocos mit app inventor vs scratch. [Manuscrito] / Pedro Henrique Barbosa Rodrigues; Leandro Henrique Correia. – – Inhumas, 2017.
 85f.: il.

 Orientador: Prof. Me. Alexandre Bellezi José.
 Monografia (Graduação) – Instituto Federal de Educação, Ciência e Tecnologia de Goiás, Câmpus Inhumas, Curso de Bacharelado em Informática, 2017.
 Bibliografia.

 1. Programação de computadores - Ensino. 2. Mit app inventor.
 3. Scratch. 4. Software educativo. I. Título.

CDD 005.13

Código 14.2017

Ficha catalográfica elaborada pela bibliotecária Maria Aparecida de Castro,
CRB-1/2599.

Biblioteca Atena, Câmpus Inhumas
Instituto Federal de Educação, Ciência e Tecnologia de Goiás

DEDICATÓRIA

Dedico este trabalho a Jesus, aos meus pais, irmã e todos que me auxiliaram nessa conquista.

Pedro Henrique Barbosa Rodrigues

Dedico este trabalho a todos que me ajudaram nesta conquista.

Leandro Henrique Correia

AGRADECIMENTOS

Dedico aos seres celestiais que para muitos não existem, mas mesmo sem comprovações estão presentes em minha fé; Aos meus familiares, Vanir da Silva Barbosa minha mãe guerreira e minha graciosa irmã Lara Cristina Barbosa Rodrigues, ao meu pai Adilson Sebastião Rodrigues, onde todos sempre me incentivaram a persistir; Meu amigo Leandro Henrique Correia que desenvolveu juntamente este trabalho. Aos colegas que ainda tenho contatos desde o tempo de técnico até a presente graduação; A toda equipe do IFG, professores, técnicos administrativos, profissionais terceirizados. De coração serei sempre grato. Enfim agradeço a todos que influenciaram neste trabalho seja assistindo, respondendo os questionários ou torcendo por nós.

Pedro Henrique Barbosa Rodrigues

Agradeço a todos os envolvidos, direta e indiretamente, neste trabalho. Ao meu companheiro nessa empreitada, Pedro; ao orientador, professor Alexandre; aos meus colegas de curso, enfim, muita gente tem importância para essa conquista. E obrigado à minha família por todo o apoio que me foi dado.

Leandro Henrique Correia

Não é tanto por eu estar confiante que os cientistas têm razão, mas mais por estar confiante que os não cientistas estão errados.

(Isaac Asimov)

RESUMO

O presente trabalho apresenta um estudo sobre ferramentas de programação em bloco para a introdução de disciplinas voltadas para a lógica de programação no ensino médio e cursos superiores. Primeiramente, analisou-se o cenário do ensino de lógica e programação de computadores por meio do estudo de artigos científicos. Pesquisas bibliográficas foram realizadas para elucidar os principais aspectos relacionados à formação do aluno de informática, às dificuldades iniciais dos discentes e à evasão em cursos com disciplinas de algoritmos, causada principalmente pelas dificuldades enfrentadas e desmotivação dos discentes. Realizou-se uma pesquisa sobre ferramentas de programação em blocos. Duas das principais ferramentas, Scratch e MIT App Inventor foram avaliadas e comparadas para aprendizagem de lógica de programação, com a finalidade de definir qual é a mais indicada para uma proposta futura de ensino-aprendizagem. Utilizou a metodologia definida de pesquisa qualitativa para a realização deste processo. Aspectos técnicos foram levados em consideração numa comparação inicial entre as plataformas de programação citadas. Aplicou-se uma oficina voltada para o ensino de um algoritmo utilizando as duas ferramentas de programação em blocos: Scratch e MIT App Inventor. Considerando o uso das ferramentas de programação em blocos, desenvolveu-se um questionário para a verificação da preferência de *software*, dificuldades enfrentadas, entre outros dados. A partir da análise destes dados, pode-se extrair algumas conclusões, a principal foi de que a ferramenta Scratch teve uma superioridade de 35% quanto à preferência entre o público de amostragem na análise matemática. E também chegou-se às preferências de acordo com o perfil do grupo, dentre esses perfis, o público que não teve contato com programação de computadores, o alvo desta pesquisa.

Palavras-chaves: Ensino de programação, MIT App Inventor, Scratch, *Software* educativo.

ABSTRACT

This work presents a study on the use of block programming tools for the introduction of disciplines about logic and programming on the high school and degree courses. Firstly, the scenario of the logic and programming classes have been analyzed by scientific articles. Bibliographic research have been realized to elucidate the main aspects related to the student formation in informatics, to the initial difficulties of teachers and to the evasion in courses with algorithms, caused mainly by the faced difficulties and demotivation of teachers. A survey on block programming tools have been made. Two of the main ones, Scratch and MIT App Inventor have been evaluated and compared to the learning of programming logics, with the goal of define which one is the most indicated to a future propose of teaching-learning. The qualitative research methodology has been used to the realization of this process. Technical aspects was taken into consideration in an inicial comparison between the programming platforms cited. It has been applied a workshop turned to the teaching of an algorithm using the two block programming tools, Scratch and MIT App Inventor. Considering the educational programming proposal using the block programming tools, it has been developed a questionnaire to the analysis of software preference, difficulties faced, among other data. From the analysis of this data, some conclusions have been extracted and the principal was that the Scratch tool have had 35% more preference between the public of sampling in the mathematical analysis. And it has also arrived to preferences according to the group's profile, among them, the ones that haven't had contact with computer programming, the target of this research.

Keywords: Programming education, MIT App Inventor, Scratch, Educational software.

LISTA DE ILUSTRAÇÕES

Figura 1: Ambiente de desenvolvimento do Blockly	28
Figura 2: Ambiente de programação do Scratch	31
Figura 3: Ambiente de Programação do MIT App Inventor	33
Figura 4: Layout - MIT App Inventor	35
Figura 5: Programação em bloco - MIT App Inventor	36
Figura 6: Partes da interface com o usuário - Scratch	38
Figura 7: 2ª Apresentação da oficina	40
Figura 8: Fluxogramas do Algoritmo de IMC com o MIT App Inventor e o Scratch	42
Figura 9: IMC – AIA x Scratch	43
Figura 10: Preferência das ferramentas quanto à usabilidade	44
Figura 11: Análise da preferência de ferramenta de acordo com o contato/prática de programação	46
Figura 12: Qual o motivo que te levou a participar dessa oficina?	47
Figura 13: Conhecimento das ferramentas	48

LISTA DE TABELAS E QUADROS

Tabela 1: Preferências de usabilidade de acordo com o contato/prática com programação de computadores	45
Tabela 2: Preferência de ferramenta de acordo com o contato/prática de programação	45
Tabela 3: Dados comparativos entre as ferramentas	53
Quadro 1: Dificuldades e Soluções para a aprendizagem de algoritmo/lógica de programação	23
Quadro 2: Causas do baixo rendimento dos alunos na aprendizagem de Algoritmos parte1	24
Quadro 3: Causas do baixo rendimento dos alunos na aprendizagem de algoritmos parte 2	25
Quadro 4: Quadro de sugestões para que a oficina fique mais interessante, despertando o interesse do uso das ferramentas em horas vagas	47
Quadro 5: Quadro de análise de dados	50

LISTA DE ABREVIATURAS E SIGLAS

AIA : App Inventor for Android

App: Aplicativo

ATP: Algoritmos e Técnicas de Programação

DJ: Disc Jockey

IDE: Integrated Development Environment

IFG : Instituto Federal de Educação, Ciência e Tecnologia de Goiás

IMC: Índice de Massa Corporal

LSQ: Learning Styles Questionnaire

MIT: Massachusetts Institute of Technology

PHP : Personal Home Page/Hypertext Preprocessor

TIC : Tecnologias de Informação e Comunicação

TDIC: Tecnologias Digitais de Informação e Comunicação

WEB: Word Wide Web

SUMÁRIO

INTRODUÇÃO	15
Trabalhos relacionados	17
Motivação	18
Objetivos	19
Objetivo geral.....	19
Objetivos específicos	19
Estrutura do trabalho	19
1. REFERENCIAL TEÓRICO	20
1.1 Estilos de aprendizagem e importância do lúdico	20
1.2 O processo tradicional de ensino de lógica de programação.....	21
1.3 Fatores que influenciam na dificuldade de aprendizagem	22
1.4 Abordagens de ensino-aprendizagem utilizando Tecnologias Digitais de Informação e Comunicação	26
1.4.1 Ferramentas de lógica de programação em blocos	27
1.4.1.1 Blockly	28
1.4.1.1 Ferramentas de programação em blocos com Arduino	29
1.4.1.3 NoBug's Snack Bar.....	30
1.4.1.4 Scratch	30
1.4.1.5 MIT App Inventor	32
2. METODOLOGIA	34
2.1 MIT App Inventor	35
2.2 Scratch	37
2.3 Oficina	38
2.3.1 Descrição da organização da oficina	39
2.3.2 Algoritmo utilizado nas oficinas	41
2.3.3 Questionário.....	43
3. RESULTADOS	44

4. CONCLUSÃO E TRABALHOS FUTUROS	54
4.1 Conclusão	54
4.2 Trabalhos futuros	55
REFERENCIAL BIBLIOGRÁFICO	56
APÊNDICE	60
APENDICE A – Questionário.....	60
APENDICE B – Respostas do questionário	66

INTRODUÇÃO

O atual ambiente educacional está intimamente relacionado com o uso de tecnologias. As TIC (Tecnologias de Informação e Comunicação) oferecem novos recursos e estratégias, permitindo grande flexibilidade, adaptação de interesses e características no ambiente educacional, principalmente no que tange às metodologias de ensino.

A partir desta perspectiva, as TIC estão relacionadas com a informática na educação, em um esforço para acompanhar as mudanças no processo de ensino-aprendizagem em diferentes habilidades, interesses, motivações, culturas e línguas que são encontrados nas salas de aula. Por um lado como um motivador e, por outro, como um recurso de ensino que oferece múltiplas possibilidades de intervenção de acordo com as necessidades e características de cada aluno, podendo fazer contextos diferentes disponíveis e elementos que facilitam a igualdade de oportunidades na educação. Dessa maneira, facilita a comunicação, acesso e processamento de informações, atividades e conteúdo do currículo, bem como a participação do discente no processo da educação e como integrante da sociedade (ALBA, ROIG E BERNAL, 2011), (BENIGNO, BOCCONI e OTT, 2007) e (MARQUÉS, 2010).

Professores, das mais diversas áreas, percebem a necessidade de atualizar suas metodologias de ensino, tornar o conteúdo de mais fácil absorção e para isso utilizam uma gama de elementos. Nos cursos que possuem disciplinas de programação de computadores em suas matrizes curriculares, não é diferente, pois tais disciplinas vêm carregadas de dificuldades. Entretanto, os docentes cada vez mais, possuem recursos que podem ser utilizados para driblar essa perspectiva. Uso de táticas pedagógicas, desafios, tecnologia e até mesmo ferramentas alternativas de programação. O presente trabalho, foca neste último elemento.

Os conhecimentos lógicos e matemáticos são necessários para qualquer aluno que precise programar computadores, devido sua necessidade intrínseca. Assim um método didático e prático para iniciá-los na tarefa de programação de computadores é algo altamente desejável. Como afirma Demo (2011, p.10): “[...] a alfabetização não acontece mais apenas na escola ou em ambientes restritos [...]”.

Praticamente todos os processos envolvendo informação utilizam tecnologia. Com isso, faz-se necessário realizar pesquisas e criações para aprimoramento, acompanhamento dos meios de ensino e aprendizagem com os avanços tecnológicos. Isso pode acarretar

alterações significativas e positivas de indicadores de qualidade do ambiente acadêmico, que podem gerar maior produção e diminuir a margem de erros profissionais.

Nos últimos anos, diversas pesquisas, como mostram Pereira e Rapkiewicz (2004), indicam uma preocupação crescente com o processo de ensino-aprendizagem de programação nos cursos superiores. Com esta preocupação buscou-se estudar na literatura os principais aspectos relacionados à formação do aluno de informática, suas dificuldades iniciais e encontrar alguma forma de reduzir a evasão em cursos com disciplinas de algoritmos.

Surgem, então, novas possibilidades educacionais no campo da programação de computadores, uma vez que seu foco não é apenas formar desenvolvedores, mas sim pessoas capacitadas para pensar de forma lógica e eficiente. Algumas ferramentas foram criadas para abordar a atividade de programação de forma alternativa, que fosse diferente das linhas de código, mas que seguissem os mesmos princípios lógicos e semânticos relativos à forma tradicional de programar (SCRATCH, 2012), (MIT APP INVENTOR, 2016). Assim, têm-se as ferramentas de programação *puzzle*.

Dentre as ferramentas que foram desenvolvidas com o propósito de oferecerem métodos alternativos de programar, algumas ganharam relativo destaque, como o Blockly. Há também o SCRATCH (DERENE, 2013); a linguagem de programação LOGO que pode contribuir para a aprendizagem de conceitos matemáticos relativos a geometria, mesmo nas interações iniciais de crianças (GREGOLIN, 2015); o MIT (Massachusetts Institute of Technology) App (Aplicativo) Inventor voltado para a programação de *smartphones* para estimular as pessoas a desenvolverem seus próprios aplicativos, dentre outras que serão mencionadas posteriormente.

Nesse contexto, jogos e ferramentas que transponham a forma tradicional podem ser de grande ajuda. Um dos aspectos centrais de jogos é seu potencial para fomentar a exploração e descoberta autoguiada de soluções (MARCHIORI; BLANCO; TORRENTE; ORTIZ; MANJON, 2011). As escolas devem repensar que tipo de cidadão irá formar, ou seja, quais são as competências que os estudantes devem possuir, porque eles precisam de novas habilidades tais como saber como pesquisar, analisar, sintetizar e desenvolver um espírito crítico ao fluxo massivo de conteúdo a que eles têm acesso (CACEBRO e LLORENTE, 2006). Pesquisou-se artigos que sugerem uma forma alternativa para apurar o raciocínio e a capacidade de estruturação lógica nas instituições, envolvendo os discentes de forma a interessar em aprofundar nas possibilidades das ferramentas.

Trabalhos Relacionados

O trabalho (JUNIOR et al, 2012) fez uso da ferramenta Blockly como recurso para ensino de programação para alunos do ensino médio e fundamental. Para tal estudo, foi utilizada uma turma do curso técnico integrado ao ensino médio da área de informática e o conteúdo de programação foi introduzido por meio da proposta com a ferramenta de programação em bloco.

Os alunos executaram atividades que simulam desafios lógicos. As atividades possuíam níveis diferentes de complexidade e trabalhavam com conceitos como laços de repetição e estruturas de decisão. Houve a aplicação de questionários para obter dados e avaliar a resposta dos alunos. A proposta demonstrou-se eficiente no estímulo dos alunos a respeito do desenvolvimento de conhecimento.

Duda et al. (2015) descreve um estudo voltado para a elaboração de aplicativos matemáticos com o MIT App Inventor, no Instituto Federal do Paraná, campus Irati, com o intuito de aprimorar as noções de lógica de programação dos alunos, além de trabalhar o conhecimento matemático. Chegou-se a desenvolver alguns aplicativos para a solução de funções matemáticas e afins. O projeto ainda não foi finalizado, mas obteve bons resultados e destacou o potencial pedagógico da ferramenta MIT App Inventor.

Na pesquisa com o Scratch (SILVA, 2015) foi possível observar o quanto a ferramenta ajuda na autonomia, na hora de resolver problemas, aumentando assim a capacidade dos alunos de sistematizar e criar. O Scratch age como um motivador para o desenvolvimento de algoritmos. No trabalho foi considerado para análise, as notas dos alunos na disciplina ATP (Algoritmos e Técnicas de Programação) e questionários aplicados. Foi notado que a ferramenta Scratch promove o trabalho colaborativo e em suas propostas futuras, utilizamos a ideia de aplicar a oficina em eventos relacionados a tecnologia.

Entretanto, nestes trabalhos citados não houve a comparação de ferramentas de programação, para identificar a que fosse mais apta para tais propostas. Foram realizados apenas o uso das ferramentas e o acompanhamento dos resultados obtidos com isso. É importante verificar a ferramenta mais adequada para o uso pedagógico e encontrar um resultado para esta questão envolve a análise das plataformas de programação em blocos mais populares e disponíveis para o uso em ambientes de ensino.

Motivação

O uso da tecnologia vem se desenvolvendo, conseqüentemente o mercado e a exigência de capacitação profissional da área se tornam mais exigentes. Os estudantes, instituições, profissionais devem procurar aperfeiçoar aquilo que foi aprendido.

Para Santos e Costa (2006) o fato do estudante não conseguir, de forma simples, construir o raciocínio lógico pode ser considerado como um dos principais fatores que corroboram para os altos índices de evasão nas disciplinas de programação e, também dos cursos superiores que possuem estas áreas como elementos curriculares. Isso motiva a busca e pesquisa de métodos e ferramentas que despertam o interesse e facilitam o estudo da lógica e programação, podendo evidenciar assim a capacidade que todos têm de pensar logicamente.

O aperfeiçoamento da aprendizagem e ensino deve ser realizado constantemente, implicando em uma transformação do que e como as pessoas aprendem, como organizar e dirigir a escola para que se torne uma base educacional eficaz e de qualidade, capaz de treinar estudantes como adultos relevantes na sociedade de hoje (TRAHTEMBERG, 2000). Entretanto, num cenário real e periódico, levando em conta as estruturas físicas e psicológicas das instituições, diversos parâmetros influenciam na formação de um bom conhecedor lógico-matemático. Assim, as ferramentas de programação em blocos facilitam o alcance do objetivo, pois reduzem frustrações sintáticas e cognitivas vividas por novatos enquanto aprendem pelo meio do uso de programação textual (CHADHA; TURBAK, 2014).

O prazer em praticar algo otimiza a tarefa, assim é também com o ato de aprender. Como mencionado no presente trabalho, a evasão em disciplinas de programação tem como uma das razões a desmotivação do discente por causa das dificuldades encontradas nos cursos. (CHAVES DE CASTRO et al., 2003), (RODRIGUES, 2002) e (DELGADO et al., 2004). Logo, vê-se a necessidade de motivar esse aluno e prender sua atenção ao conteúdo lecionado, até que o mesmo aprenda os conceitos e consiga estruturar o raciocínio lógico, item fundamental para um programador. Este trabalho aborda uma linha das possibilidades encontradas nas propostas futuras do trabalho de Silva (2015), que ressaltou a importância de estimular crianças e adolescentes para novas aprendizagens envolvendo o uso de computadores e dispositivos móveis.

Objetivos

Objetivo geral

O presente trabalho propõe a pesquisa e comparação entre duas ferramentas digitais como método de ensino inicial do conteúdo de lógica de programação: **Scratch** e **MIT App Inventor for Android**.

Objetivos específicos

- Articular proposta de incentivo para o despertar do interesse em programação;
- Avaliar ferramentas alternativas e qualificadas para aprendizagem de lógica de programação;
- Desenvolver atividades para apresentação de oficinas e técnicas de lógica de programação junto aos discentes.

Estrutura do Trabalho

Elaborou-se um sólido referencial teórico para compreender melhor a situação do ensino de programação nas instituições de cursos superiores. O capítulo 1 apresenta informações gerais e situa o leitor a respeito da questão educacional abordada pelo presente trabalho, assim como, as alternativas de ferramentas digitais que podem agregar à parte introdutória do ensino de programação.

O capítulo 2, referente à metodologia descreve as ferramentas utilizadas na comparação e o que foi comparado entre as duas. Relata-se também, as oficinas elaboradas para o ensino de programação por meio de métodos alternativos, assim como todos os processos executados para a obtenção de dados significativos.

O capítulo 3 apresenta os resultados obtidos por meio de análises estatísticas, matemáticas e representações gráficas dos dados obtidos nos processos descritos no capítulo anterior.

O capítulo 4 descreve as considerações finais do presente trabalho e apresenta propostas de trabalhos futuros.

1. REFERENCIAL TEÓRICO

1.1 Estilos de aprendizagem e importância do lúdico

Um dos conceitos discutidos a partir das teorias de ensino são os estilos de aprendizagem e como criar metodologias de ensino correlacionadas. Estilos de aprendizagem são conceitos que abordam novas formas de se pensar a maneira que os indivíduos aprendem. Trata-se de métodos de construir aprendizado bastante específico para cada ser humano, de acordo com suas habilidades e características pessoais.

Segundo Amaral e Barros (2007), o conceito de estilos de aprendizagem contribui para o desenvolvimento de metodologias de ensino sob a perspectiva de uso das tecnologias como ferramenta educacional, pois essa visão teórica considera as diferenças de indivíduo para indivíduo, o que a torna flexível e assim permite estruturar as especificidades voltadas à utilização de tecnologias.

Honey e Mumford (1986), a partir dos estudos de Kolb (1976), desenvolveram um método que visa classificar os estilos de aprendizagem, chamado de *Learning Styles Questionnaire* (LSQ), de forma a identificar a proficiência de aprendizagem de cada indivíduo. Entretanto, o LSQ é mais voltado para a questão empresarial. Na área de ensino, os estudos de Alonso e Gallego (2002) desenvolvem a classificação com quatro classes de formas de aprendizagem, são elas: ativo, reflexivo, teórico e pragmático.

A teoria dos estilos de aprendizagem tem por objetivo identificar a forma predominante que um indivíduo possui para aprender, e não reduzir as pessoas a apenas um estilo. Assim, pretende-se ampliar as formas de se transmitir conhecimento, pois os indivíduos de uma sociedade possuem distinções quanto a maneira de aprender, e assim, quanto maior a variedade de abordagens para o ensino, as pessoas terão mais opções para desenvolver seu aprendizado. (AMARAL; BARROS, 2007). Com a disposição da tecnologia, esses estilos podem ser mais facilmente englobados em um processo de ensino.

O tipo de aprendizagem que a influência da tecnologia potencializa nos contextos atuais passa necessariamente por dois aspectos: primeiramente, a flexibilidade e a diversidade e, em seguida, os formatos. A aprendizagem do indivíduo sobre os temas e assuntos do mundo deve ser realizada de maneira flexível, com diversidade de opções de línguas, ideologias e reflexões (AMARAL; BARROS, 2007, p. 3).

Segundo Vygotsky (1988), aprendizado e desenvolvimento estão diretamente relacionados. Aprendizagem é o processo pelo qual o indivíduo adquire informações, habilidades, aptidões, etc., a partir de seu contato com o meio em que vive, ou seja, a

sociedade. Assim, o aprendizado está atrelado à interação social e os meios lúdicos são uma grande ferramenta para que esse processo ocorra de uma boa maneira. A partir de atividades lúdicas o indivíduo desenvolve suas inteligências e habilidades sociais (BATISTA, 2012). Batista (2012, p. 23) explica:

O professor precisa conhecer a bagagem de conhecimento prévio que cada criança traz consigo, e agir no sentido de ampliar suas noções matemáticas, ou seja, é necessário respeitar a criança na sua inteligência, no seu aprendizado construído, para que a aprendizagem seja significativa e prazerosa.

Piaget (1998) acredita que os jogos são essenciais no processo de desenvolvimento e maturação intelectual do indivíduo, pois a atividade lúdica é campo obrigatório das atividades intelectuais da criança e conseqüentemente, indispensável à prática educacional. Segundo este autor, os jogos não são uma forma de divertimento inútil, mas são meios que contribuem e enriquecem o desenvolvimento intelectual.

A tecnologia, com um grande leque de possibilidades, sejam elas de ferramentas, metodologias ou funcionalidades, oferece novos meios de construção de conhecimento. O presente trabalho compara duas ferramentas que podem ser usadas como um desses recursos que a tecnologia permite. Assim, pode-se tentar a utilização das plataformas de programação em blocos nas abordagens de ensino de lógica de programação. O presente trabalho compara duas ferramentas a fim de identificar a mais apta para tal proposta.

1.2 O processo tradicional de ensino de lógica de programação

Diversas teorias sobre abordagens de ensino-aprendizagem afirmam que o aprendizado só ocorre com a experiência e a prática. O aprendizado de lógica de programação também segue o mesmo conceito, pois trata-se uma atividade humana relacionada à resolução de problemas (NÁVRAT, 1996), cujo aprender está intimamente relacionado com a capacidade do indivíduo de trabalhar a abstração.

A lógica de programação já é introduzida com o uso de alguma linguagem, seja algo mais próximo do algoritmo com pseudocódigo, ou mesmo com linguagens como: Java, C, PHP, Python. O estudo de lógica de programação é considerado de grande dificuldade por discentes da área de informática, e já existem pesquisas para melhorar a processo de ensino-aprendizagem nos cursos relacionados, (NEVES & COELLO, 2006) e (COSTA et al., 2010).

Há a metodologia tradicional de apresentar de forma expositiva o conteúdo de

determinada disciplina, concentrando-se na solução de problemas por meio de pseudocódigo (NOBRE & MENEZES, 2002). Entretanto, este modo não consegue facilmente motivar os discentes a se interessarem pelas disciplinas de programação, pois não é transmitida com clareza a importância de certos conteúdos para a formação dos alunos (BORGES, 2000).

Segundo Schots et al. (2009), a programação não é uma atividade simples, pelo contrário, trata-se de uma tarefa complexa, pois requer o entendimento de linguagens específicas, sintaxe e a lógica da programação dos algoritmos. Portanto, a forma com a qual os conceitos abstratos são abordados nos primeiros semestres dos cursos é significativa e pode ser decisiva para a motivação dos estudantes. Diversos alunos perdem o interesse dadas às dificuldades encontradas e isso ocasiona altas taxas de reprovação e até evasão. São muito frequentes os casos de evasão de curso, fato que tem relação íntima com as dificuldades de aprendizagem (CASTRO et al., 2003).

Autores como (CASTRO et al, 2003), (ROSSO & DANIELE, 2000) e (SANTOS et al, 2003) sugerem a implantação de outras abordagens para o ensino-aprendizado das disciplinas introdutórias à programação. São elas: o uso de ferramentas computacionais, a utilização de formas alternativas ou a integração de ambas (PEREIRA & RAPKIEWICZ, 2004). Dentro dessa classificação, destacamos para esse trabalho o uso de ferramentas digitais.

Algumas metodologias de ensino-aprendizagem têm utilizado jogos e métodos alternativos como ferramentas de ensino, tanto para a lógica quanto a programação. De fato, tem se mostrado eficiente quando bem elaborado e corretamente conduzido pelos professores responsáveis. Estudos como (MARQUES; MARQUES, 2011) apontam indicativos relacionando melhores competências para a execução de tarefas do ensino por alunos inteirados com jogos computacionais, além de reforçar a importância da integração de estudo e jogos como um poderoso agente de ensino.

Novas abordagens vêm sendo utilizadas a fim de tornar a aprendizagem de lógica de programação algo mais intuitivo. Jogos e ferramentas interativas que exploram o aspecto gráfico proporcionam um meio lúdico para o entendimento e desenvolvimento de algoritmos, ao explorarem aspectos de cognição e de colaboração (SILVA, 2007).

1.3 Fatores que influenciam na dificuldade de aprendizagem

Borges (2000) aponta um conjunto de fatores que podem levar às dificuldades de aprendizagem no ensino de lógica de programação, como: a falta de acompanhamento

docente individualizado; dificuldade do estudante em desenvolver o raciocínio lógico, a ponto de decorar conteúdos; falta de persistência mediante a complexidade e dificuldade imposta pela disciplina de algoritmos; baixa capacidade de abstração no início do curso superior; o “modelo tradicional” de ensino, onde os discentes são meros receptores, não participativos; ausência, em algumas instituições, de metodologias de ensino focadas nos conteúdos de lógica, as quais incentivam a utilização de aplicações de apoio à aprendizagem.

No entanto, muitas foram as iniciativas de estudo para estas questões que trazem fortes consequências no processo de ensino e aprendizagem em lógica (Quadro 1). Adoção de teorias voltadas à construção do raciocínio lógico, apoiadas pela aprendizagem significativa, proposta por Ausubel et al. (1978), que explora a construção de materiais didáticos significativos, buscando de forma simples uma conexão entre os conteúdos apresentados com o processo de aquisição do conhecimento do aluno, possibilitam que surjam mudanças nesse processo devido novas informações assimiladas. Um grande diferencial nesta teoria é o foco no discente, por meio da identificação e ligação de conceitos subsunçores, já reconhecidos, com as novas informações recebidas recentemente que se ligam à base de conhecimento do estudante, construindo assim um aprendizado significativo.

Outra teoria apresentada (Quadro 1) é o modelo construtivista no contexto de informática. Nesse caso, o estudante, a partir da interação com o meio, é induzido a construir o conhecimento sobre lógica e algoritmos, utilizando os sistemas de programação em blocos visuais e recursos de mundos virtuais. Acompanhando o desenvolvimento tecnológico e formando o atual socioconstrutivismo.

segundo este referencial, o conhecimento não é uma representação da realidade, mas um mapeamento das ações e operações conceituais que provaram ser viáveis na experiência do indivíduo. Portanto, a aprendizagem é um resultado adaptativo que tem natureza social, histórica e cultural. (BOIKO; ZAMBERLAN, 2001, p. 51).

Dificuldades e soluções para a aprendizagem de Algoritmo/ Lógica de programação

Dificuldades	Autores	Soluções Propostas	Teoria de Aprendizagem
Desenvolvimento do Raciocínio Lógico	Silva(2009) Souto(2009) Vieira(2008) Falckembach(2011)	Estratégia Ascendente de solução de problemas Jogos Educacionais	Construtivismo

Criação de Abstrações	Silva(2009) Falckembach(2011) Machado(2011)	Ilustração de Algoritmos <i>Software</i> Especialista	Construtivismo
Inexperiência em Práticas Comerciais/Industriais	Mattos(2008)	Simulação de Práticas Comerciais/Industriais em Jogos e Estudos de Caso	Socio- Construtivismo
Impossibilidade de Acompanhamento Individualizado	Oliveira(2008) Pimentel(2003)	Monitoria com alunos que dominem o conteúdo	Socio- Construtivismo
Disparidade de Conhecimento e Ritmo de Aprendizagem	Pimentel(2003) Machado(2012) Ausubel(2003)	Aulas de Nivelamento, Objetos de Aprendizagem	Aprendizagem Significativa
Baixa Compreensão de enunciados dos algoritmos	Pimentel(2003) Ausubel(2003)	Aulas de Interpretação de texto	Aprendizagem Significativa

Quadro 1 – Dificuldades VS soluções. **Fonte:** Adaptado de (FARIA & LIMA, 2011).

Nitidamente o contexto envolve todo um enredo, que vai desde o incentivo e apoio dentro de casa até o ambiente acadêmico com o relacionamento docente-discente presente nos ambientes de ensino (Quadros 2 e 3).

Causas do baixo rendimento dos alunos na aprendizagem de Algoritmos - parte 1

Causas relacionadas diretamente ao docente
Incapacidade de reconhecimento de habilidades dos alunos impedindo traçar estratégias pedagógicas apropriadas;
Falta de criatividade do docente para apresentar novas técnicas de solução de algoritmos;
Inabilidade em promover a cooperação e colaboração entre os alunos;
Formação pedagógica deficitária;

Visão instrumental sobre o ensino;
Uso de metodologia inadequada, gerando a desmotivação para aprendizagem de Algoritmo;
Planejamento e execução das aulas sem objetivos preestabelecidos ou mal dimensionados;
Ensino conjunto de algoritmo e linguagem de programação;
Ensino focado em determinada linguagem de programação.

Quadro 2 – Causas relacionadas aos professores para o baixo rendimento dos alunos na aprendizagem de Algoritmos. **Fonte:** Adaptado de Faria (2013).

Causas do baixo rendimento dos alunos na aprendizagem de Algoritmos - parte 2

Causas relacionadas diretamente ao discente
Deficiência no raciocínio operatório-formal, que é a base para o raciocínio lógico;
Hábito de decorar ao invés de compreender o conteúdo;
Dificuldade de formar abstrações;
Baixa capacidade de interpretação e compreensão de enunciados dos algoritmos e de textos em geral;
Falta de experiência com práticas comerciais e industriais;
Baixo nível de conhecimento em Matemática;
Ineficiência em armazenar e classificar mentalmente o conteúdo estudado.

Quadro 3 – Causas relacionadas aos alunos para o baixo rendimento dos alunos na aprendizagem de Algoritmos. **Fonte:** Adaptado de Faria (2013).

Seria um erro esquecer o fator fundamental que é o próprio elemento que busca pelo conhecimento, no caso o aluno apto para desenvolver-se profissionalmente deve, segundo Farias et al. (2012) e Gerhard & Brusilovsky (2001), apresenta algumas competências, como: possuir a capacidade de abstração e raciocínio lógico para propor uma solução; reconhecer a

sintaxe e semântica de uma linguagem de programação; entender a forma correta de utilizar um ambiente de programação; reconhecer o processo de compilação, teste e depuração do código fonte e; por fim desenvolver a habilidade de solucionar problemas. Ressaltando a necessidade do empenho pessoal na absorção e prática dos conhecimentos que se forem guiados por bons métodos aumentam a possibilidade do acerto.

1.4 Abordagens de ensino-aprendizagem utilizando Tecnologias Digitais de Informação e Comunicação – TDIC

O uso de ferramentas computacionais na educação surge quando começaram a ser comercializados os primeiros computadores, ainda nos anos de 1950. Nessa época, os computadores tinham a simples função de armazenar informação e transmiti-la ao aprendiz. Nos dias de hoje, a utilização do computador como uma ferramenta de ensino é amplamente mais diversificada. A expressão “Informática na Educação” refere-se à inserção do computador no processo de ensino-aprendizagem de conteúdos curriculares diversos previstos na matriz educacional de diferentes níveis.

Quando o computador é usado para transmitir determinadas informações para o aluno, é apenas uma máquina. Porém, quando o aluno utiliza-o para construir o conhecimento, o computador passa a ser uma ferramenta muito mais poderosa e com amplos recursos. Empregar o computador com essa intenção implica em entendê-lo como uma ferramenta que possui uma nova maneira de representar o conhecimento, buscando novas ideias e conceitos. Também requer a análise cuidadosa do que significa ensinar e aprender, assim como rever o papel do professor nesse contexto.

Existem diversas contribuições do uso de TDIC para ensino-aprendizagem com diferentes abordagens pedagógicas. A exemplo disto pode-se citar o uso de blogs para produção textual, o uso de simuladores, uso de aplicativos de celular, etc. Destaca-se, entretanto, que o uso da TDIC não garante que haja uma abordagem menos tradicional. A postura pedagógica do professor que conduz a situação de aprendizagem é fundamental para o desenrolar de tal abordagem.

O trabalho (ZIEDE et. al., 2016) apresenta uma pesquisa sobre o cenário das escolas do município de Caçador, SC, quanto ao uso das TDIC no ambiente escolar. Descreve-se o uso dessas ferramentas e como isso deveria acontecer na prática, mas ressalta a

importância da capacitação dos professores para utilizar estes recursos, assim como, o planejamento por parte da escola, visando adicionar ao planejamento pedagógico as TDIC. Ressaltando a importância do planejamento, e utilização das ferramentas mais adequadas para o processo de ensino-aprendizagem.

Seguem alguns exemplos identificados de uso de tecnologias da informação sendo utilizado em processos de ensino. Pacheco e Ferreira (2016) descreve o uso de TDIC para inteirar os participantes a respeito de questões políticas e históricas. (HEINEN et. al., 2015) apresenta uma abordagem utilizando a ferramenta Blockly para criar um ambiente de programação para um robô, e dessa forma desenvolver uma oficina que estimule alunos a programarem. O autor justifica o uso da plataforma pela facilidade de utilização da linguagem de programação em blocos, e pelo Blockly se adequar a proposta com o modelo robótico.

Nas próximas seções lista-se alguns *software* que exercitam a lógica e técnica de programação de computadores. A grande maioria utiliza o paradigma de programação utilizando blocos, como recurso didático.

1.4.1 Ferramentas de lógica de programação em blocos

É necessário mencionar a importância da linguagem LOGO desenvolvida pelo MIT, surgida na década de 60, sendo um dos estudos pioneiros no uso de uma linguagem de programação como uma ferramenta de ensino para crianças. A proposta era criar ações e desenhos por meio de comandos definidos. O cursor, representado por uma tartaruga, ao se mover, deixa um rastro, criando assim, figuras. A própria linguagem permitia a construção de funções e outras utilidades com diferentes graus de sofisticação. Papert, o criador da linguagem foi influenciado pela teoria de construtivismo de Piaget, com quem chegou a trabalhar (PAPERT, 2008).

Já no final dos anos 2000, começam a surgir algumas ferramentas de programação em blocos como o Scratch, App Inventor, Blockly e plataformas de desenvolvimento para o micro-controlador italiano, Arduino. A forma de se programar com essas ferramentas é análoga a montar objetos com o famoso brinquedo LEGO, que se resume em peças que podem ser encaixadas, para criar algo funcional, etc.

É interessante mencionar que ao limitar os encaixes entre os blocos, linguagens de programação como o Blockly, Scratch e o App Inventor, têm como objetivo diminuir a

quantidade de erros praticados no código (PASTERNAK, 2009). Assim, impede-se a possibilidade de construir código incoerente e reduz os erros de sintaxe. Esse é um recurso positivo para essas ferramentas, no que diz respeito ao uso delas como uma introdutória à programação.

A seguir, uma breve descrição de algumas ferramentas de programação em blocos, gratuitas e disponíveis para uso:

1.4.1.1 Blockly

Blockly é um editor de programação visual em blocos desenvolvido pela Google (Figura 1). Possui código aberto e, é bastante similar ao Scratch, entretanto, a principal diferença é que este editor consegue transformar o código puzzle em JavaScript, Python, PHP (Personal Home Page ou Hypertext Preprocessor), etc.

Ambiente de desenvolvimento do Blockly

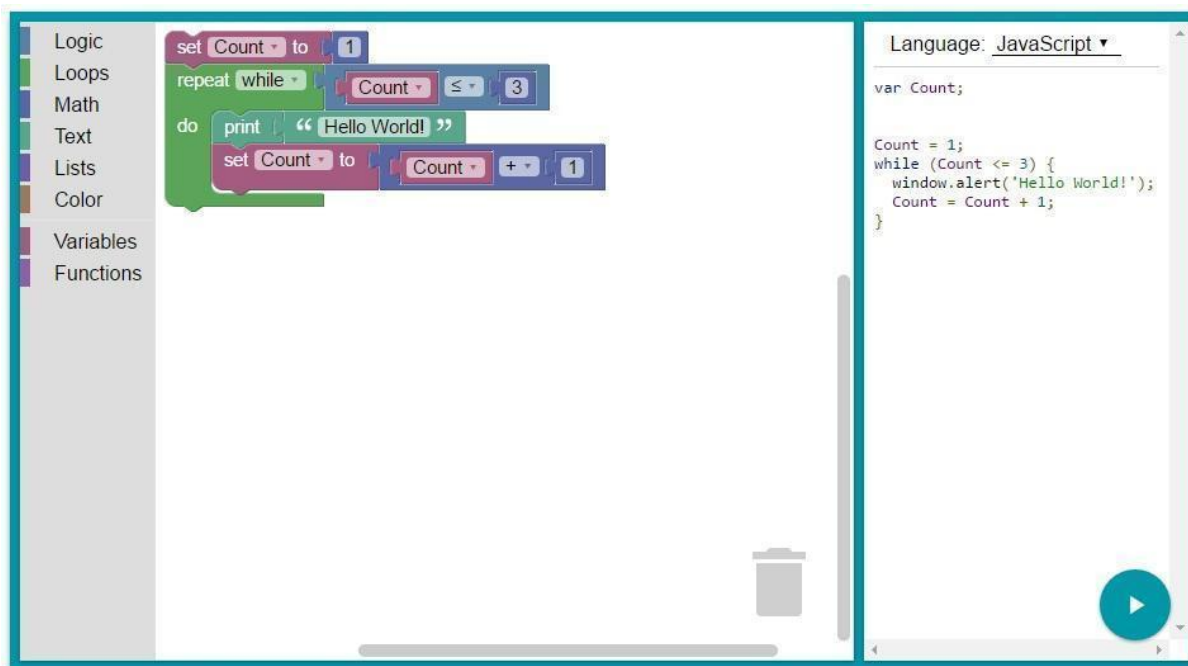


Figura 1 – Interface do Blockly **Fonte:** Página WEB – Blockly | Google Developers¹

¹Disponível em: <<https://developers.google.com/blockly/>>. Acesso em: 02/02/2017.

Anteriormente, o Blockly era implantado em conjunto com uma aplicação Java, mas atualmente é acessado via *browser*, e possui versões para Android e iOS. Blockly tem o propósito de ensinar programação a pessoas sem experiência com esta prática (VANDEVELDE et al., 2013).

Essa ferramenta também conta com um módulo que adiciona suporte a conversão da lógica criada com blocos (programação *puzzle*) para código Arduino (linguagem baseada em C/C++), conhecido como BlocklyDuino. Trata-se de uma ferramenta conhecida, com algumas pesquisas já realizadas apontando o seu uso como um recurso voltado para o ensino (HEINEN, et. al., 2015).

1.4.1.2 Ferramentas de programação em blocos com Arduino

O Arduino é um micro-controlador para prototipagem de ferramentas de *software*, desenvolvido na Itália em 2005. Seu baixo custo o popularizou no mundo, com diversas instituições de ensino utilizando-o como uma ferramenta em projetos de eletrônica, programação e afins. Existem algumas ferramentas que permitem ao usuário programar por meio de blocos as rotinas implementadas em Arduino. São elas:

Ardublock: trata-se de plataforma gráfica e similar ao Scratch. É uma extensão feita em Java e funciona integrada a IDE original do Arduino.

MiniBloq: é um ambiente de programação gráfica, onde grupos de blocos são dispostos de forma linear para criar um programa. Ao contrário do ArduBlock, o MiniBloq é um programa *stand-alone*, isso significa que não é necessário ter a IDE do Arduino instalada para poder usá-lo.

S4A: É uma versão modificada do Scratch original, o que permite ser utilizado em conjunto com o Arduino. O usuário pode acessar recursos do Arduino com os blocos gráficos relativos à ferramenta de programação.

Modkit: é uma plataforma de programação gráfica, que usa blocos para micro-

controladores. Ela permite programar Arduino e qualquer *hardware* compatível com simples blocos gráficos. Os blocos gráficos dessa ferramenta são inspirados no ambiente de programação do Scratch. O Modkit roda em um navegador WEB (Word Wide Web) e requer *widget* de *desktop* para se comunicar com o a placa de desenvolvimento (Arduino). É uma ferramenta gratuita e trata-se de uma aplicação em nuvem.

1.4.1.3 NoBug's Snack Bar

O **NoBug's Snack Bar** é um jogo gratuito de fundo educacional que usa blocos gráficos de programação como o Scratch. O jogo compartilha da mesma tecnologia do projeto Code.org do governo norte-americano (VAHLICK et al, 2015). Na abordagem do jogo, o jogador controla um funcionário de uma lanchonete e o próprio jogo controla os clientes que fazem pedidos. Com os comandos em blocos, o personagem vai ao cliente, recebe o pedido, executa as tarefas necessárias para realizar o pedido, entrega-o, etc.

Esse jogo foi desenvolvido em HTML5, Java e *framework* Blockly2, e roda em um navegador WEB. No conjunto de comandos há blocos: de condição (se... então, senão) e outros para realizarem diversas tarefas únicas. Com o progresso do jogo, os pedidos tornam-se mais complexos, exigindo mais combinações de comandos para realizá-los. O No Bug's Snack Bar possui dezesseis missões abordando conceitos como manipulação de variáveis, estruturação de ações e trabalhando condicionais.

1.4.1.4 Scratch

O Scratch é uma ferramenta de programação gráfica que possibilita a criação de histórias interativas, animações, simulações, jogos e músicas, além de ser possível disponibilizar na Web todo o conteúdo desenvolvido (Figura 2). Criada no Media Lab do Instituto de Tecnologia de Massachusetts, inspirada nas linguagens LOGO e Squeak (SCRATCH, 2012), foi disponibilizada para usuários no dia 15 de maio de 2007. Trata-se de uma ferramenta gratuita e tem, como um de seus objetivos, facilitar a introdução de conceitos de lógica e programação para alunos em diversas faixas etárias e diferentes níveis de escolaridade (SCRATCH, 2012).

A programação do Scratch é realizada com blocos de comandos que são encaixados uns aos outros de forma sequencial, como na programação tradicional. Há uma

preocupação com sintaxe e por isso os blocos possuem encaixes específicos, permitindo ao usuário uma melhor compreensão da lógica de programação.

Ambiente de programação do Scratch

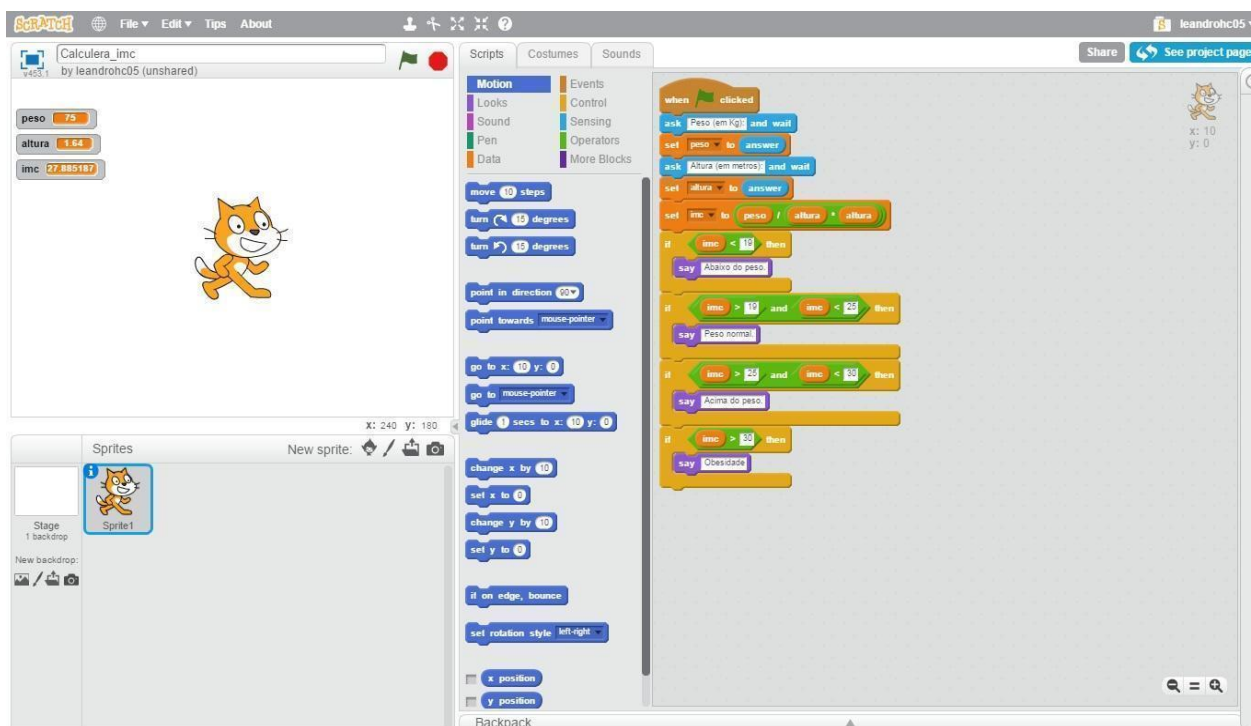


Figura 2 – Interface Scratch. **Fonte:** próprio autor

Cada bloco da linguagem contém um comando separado, que pode ser agrupado livremente caso se encaixe com outro. Os comandos podem ser modificados nos menus. Scratch se inspirou na forma como os DJs (Disc Jockey) fazem a *mixagem* de sons para criarem novas músicas. Com essa linguagem consegue-se *mixar* diversos tipos de mídias, como imagens, sons e outros programas.

Com o Scratch é possível desenvolver programas que controlam e misturam gráficos, animações (por exemplo: com o gatinho, marca registrada da ferramenta), texto, música e sons (que o usuário pode gravar e utilizar). A ferramenta está disponível online, bastando o usuário se cadastrar e assim pode criar seus próprios programas com a opção de compartilhá-los com os outros usuários do Scratch, tudo de forma gratuita. Também há a opção off-line da ferramenta, sendo disponibilizado também gratuitamente para *download* e instalação com passo a passo detalhado no próprio site.

O Scratch, assim como o Blockly e o MIT App Inventor são ferramentas desenvolvidas no MIT com o intuito de democratizar a programação, e algumas delas não são voltadas para o ensino, embora comecem a ser usadas como tal. É uma ferramenta mundialmente conhecida e bastante popular, com milhões de usuários espalhados pelo mundo e com uma infinidade de programas compartilhados (SCRATCH, 2012).

1.4.1.5 MIT App Inventor

Surgido em 2009, o projeto MIT App Inventor ou AIA, tem como objetivo democratizar o desenvolvimento de *software* e capacitar todas as pessoas, especialmente os jovens, a passarem de consumidores de tecnologia à criadores de suas próprias ferramentas computacionais e aplicações. Colideraram o projeto, Mark Friedman e o professor do MIT, Hal Abelson, em conjunto com os colaboradores da Google, Sharon Perl, Liz Looney e Ellen Spertus (MIT APP INVENTOR, 2016).

O AIA foi desenvolvido como um aplicativo que transforma linguagem complexa de codificação em uma forma mais simples e intuitiva de programação de ferramentas de *software* ou aplicativos, baseando-se em blocos de construção visual (método *puzzle*), assim como diversas ferramentas já mencionadas. O diferencial é que com o AIA o usuário pode utilizar o algoritmo para criar uma aplicação compatível com o sistema operacional Android.

O MIT App Inventor funciona como um serviço WEB administrado por uma equipe no centro do MIT para aprendizagem móvel, sendo disponibilizado a partir de 2010, online e com opção off-line. Desenvolvido na linguagem Java, é mais flexível quando usado em celulares com sistema operacional Android e computadores que possuem tal plataforma. Estima-se que o MIT App Inventor possua três milhões de usuários espalhados por quase duzentas nações, criando diversas aplicações de código aberto e, portanto disponíveis para o público (MIT APP INVENTOR, 2016).

Ambiente de programação do MIT App Inventor

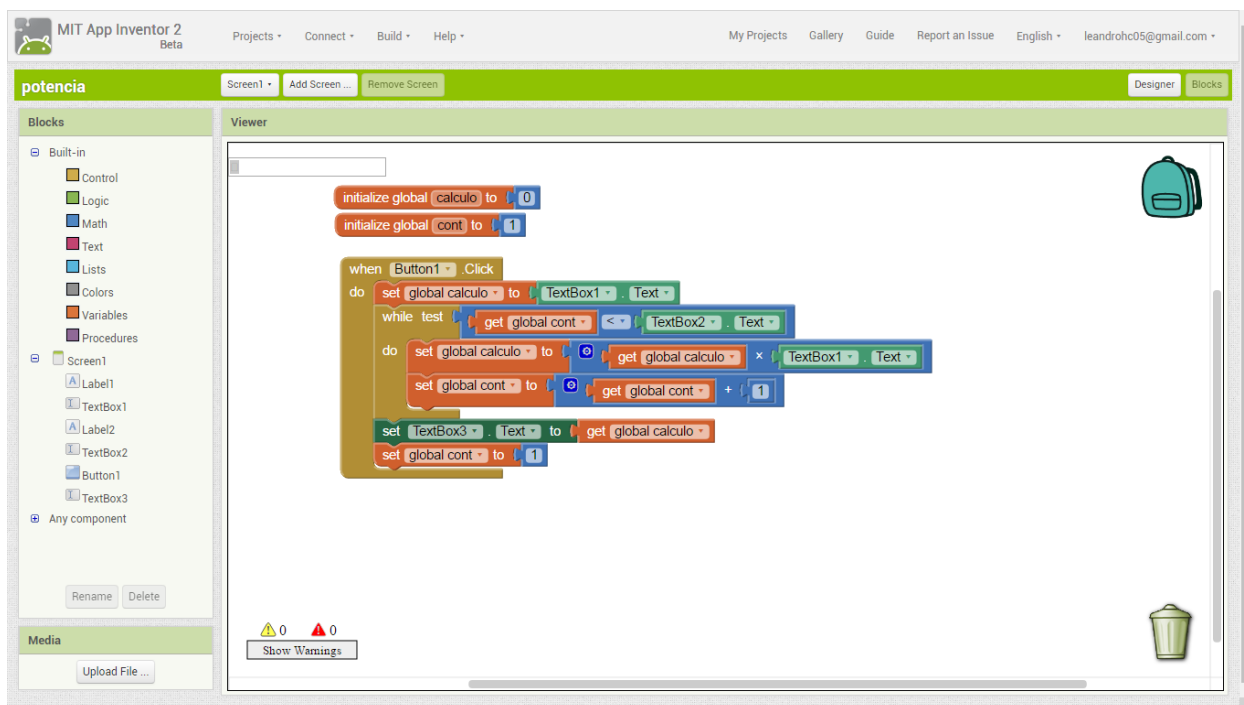


Figura 3 – Exemplo de programação em blocos com o MIT App Inventor de calculadora de potência. **Fonte:** próprio autor.

O AIA é baseado no Scratch e LOGO, mas possui como diferencial a disponibilidade de utilizar recursos do sistema operacional Android, pois a ferramenta é voltada para a produção de aplicativos, com o adicional de ser criado pelo próprio usuário, conforme suas necessidades (Figura 3). Com o AIA é possível acessar dados e aplicações do sistema, como mensagens, geolocalização, etc. dando um atrativo maior ao desenvolvimento com esta plataforma. No MIT App Inventor o aluno não tem um *feedback* imediato de suas aplicações desenvolvidas, pois é necessário gerar um QR Code ou o arquivo .apk para instalar o aplicativo no dispositivo móvel. Somente assim é possível visualizar e verificar o funcionamento das mesmas ou utilizar um emulador que a própria ferramenta disponibiliza para download e instalação com passa a passo detalhado.

2. METODOLOGIA

Esta pesquisa foi do tipo qualitativa. Os instrumentos utilizados para coleta de dados foram principalmente, a elaboração de uma oficina com a proposta de ensinar a programação em bloco com algumas das ferramentas abordadas, e também houve a aplicação de questionários durante a oficina, para os alunos participantes da mesma.

Segundo Minayo (2003) a pesquisa qualitativa tem como objetivo entender determinado fenômeno com maior precisão. Trabalha-se com descrições, comparações e interpretações dos pesquisadores, e justamente por isso, trata-se de um método em que os participantes são bem ativos, porém podem moldar o rumo da pesquisa.

Godoy (1995) afirma que uma pesquisa qualitativa não deve buscar valorar ou medir os fenômenos estudados, muito menos utilizar processos estatísticos nas análises dos dados. O pesquisador precisa estar em contato com os objetos de estudo, inteirado aos eventos, e assim tentar compreender a posição dos indivíduos relacionados à pesquisa.

Diante do estudo das ferramentas mencionadas anteriormente, considerando todas as mencionadas, optou-se por fazer um estudo comparativo entre o Scratch e o MIT App Inventor. A escolha destas duas ferramentas deve-se a grande popularidade de ambas. O Scratch possui até o mês de julho de 2017 segundo estatísticas de seu site: 24.545.513 projetos compartilhados, 20.611.426 usuários registrados, 127.182.208 comentários publicados, 3.705.893 estúdios criados, 104.858.250 visualizações de página 18.712.190 visitas. O MIT atende a mais de 6 milhões de usuários cadastrados, além de ser usado em vários projetos de alunos de instituições para resolução de soluções do cotidiano, apresentado em sua página WEB.

Outro fato que levou a escolha dessas ferramentas foi a não necessidade de periféricos eletrônicos, como no caso das ferramentas que utilizam Arduino. Buscou-se identificar as semelhanças e diferenças entre elas, e a forma isso poderia refletir na abordagem com as oficinas. O estudo leva em consideração principalmente aspectos técnicos do Scratch e MIT App Inventor e o *feedback* dos questionários aplicados nas oficinas.

De acordo com a complexidade da solução, programar em blocos também torna-se uma tarefa tão exigente quanto a programação com linhas de código. Ferramentas como o MIT App Inventor e o Scratch dispõem de diversos recursos e alguns deles são avançados e exigem um conhecimento mais aprofundado, assim como acontece na forma mais recorrente de programar.

Os recursos mais avançados dessas ferramentas em alguns casos estão relacionados com propriedades dos sistemas operacional, outros recursos de dispositivos móveis, enfim. Portanto, pode-se afirmar que ao aumentar muito a complexidade de um algoritmo, a dificuldade para fazê-lo será praticamente a mesma, independente da ferramenta utilizada para programar.

2.1 MIT App Inventor

O *layout* do AIA, diferentemente das outras ferramentas de programação em blocos, é dividido em duas fases, pois trata-se de uma plataforma voltada para a criação de aplicativos para o sistema operacional Android. Na primeira (Figura 4), desenvolve-se a interface da aplicação e cria-se as variáveis, campos de texto, caixas de combinações e afins. Já na segunda, o usuário conta com os blocos de programação para elaborar o algoritmo de como se dará a execução dessa aplicação (Figura 5).

Layout - MIT App Inventor

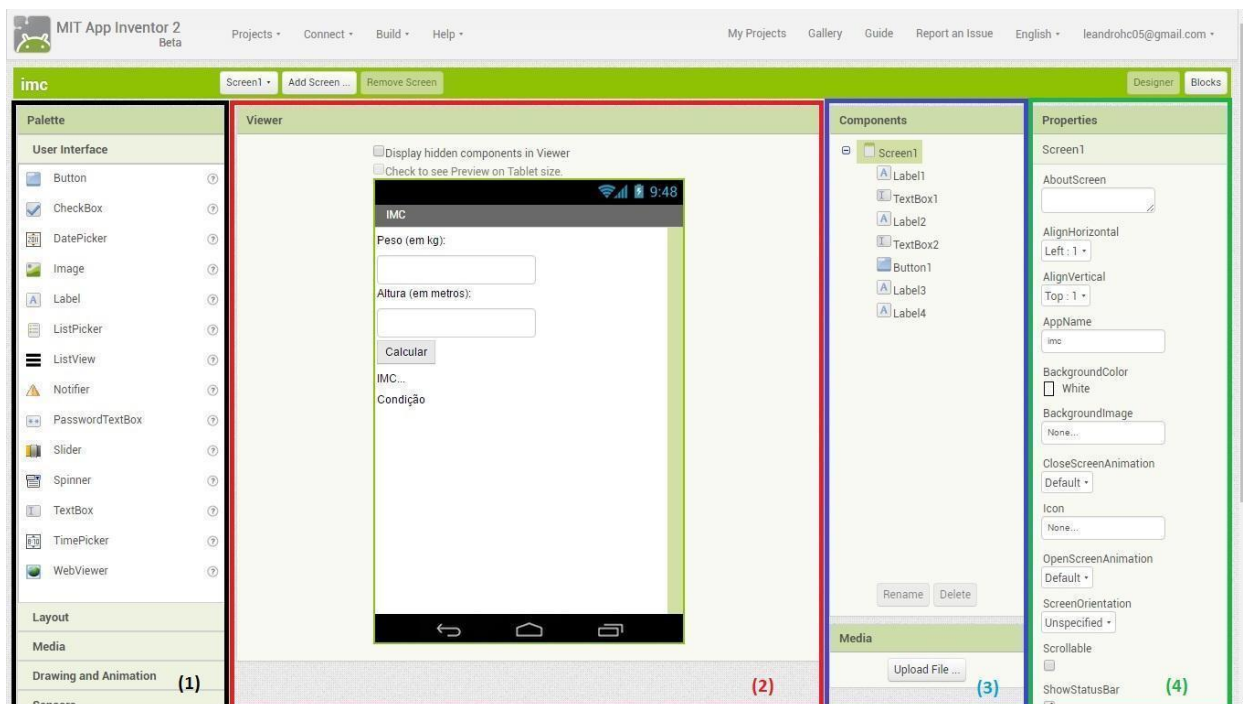


Figura 4 – Layout de criação de interface do AIA e partes (em destaque). **Fonte:** próprio autor.

A interface de desenvolvimento está disposta da seguinte maneira (Figura 4), da esquerda para a direita: paleta de componentes disponíveis para uso na criação dos aplicativos (botões, caixas de texto, caixas de seleção, textos, etc), visualização da interface da aplicação (Viewer) de forma similar a alguns ambientes de programação em que o desenvolvedor tem um *feedback* gráfico do que está desenvolvendo, lista de componentes utilizados e propriedades (normalmente relacionada com algum componente selecionado).

Programação em bloco - MIT App Inventor

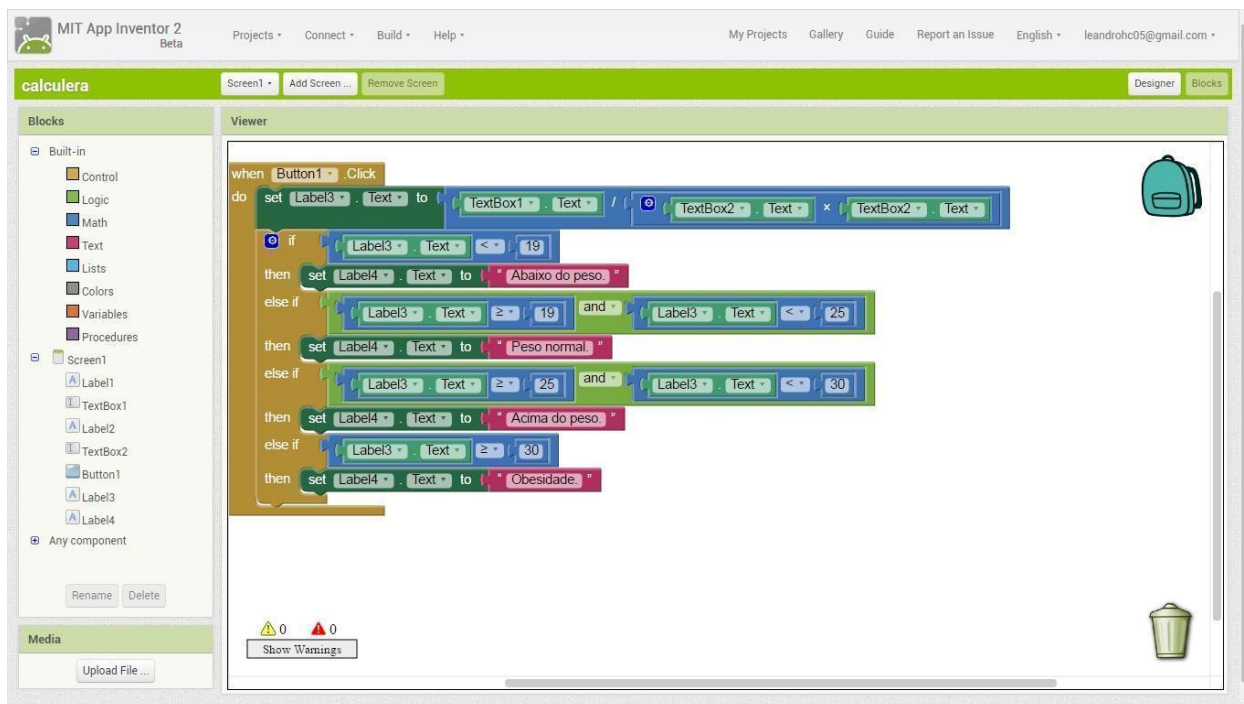


Figura 5 – Programação em blocos do AIA, exemplo IMC. **Fonte:** próprio autor.

A paleta de componentes (Figura 4, marcado em preto) contém uma disposição de itens funcionais que podem ser utilizados na aplicação. Além dos básicos como botões e caixas de seleção, conta com outros itens mais específicos do AIA, que fazem uso de informações de GPS, acelerômetro, recursos de câmera e tela. No presente trabalho, nas oficinas utilizando um algoritmo para o ensino das ferramentas, fez-se o uso dos itens mais básicos, dada a proposta. Todos os recursos utilizados estavam dispostos na opção “User Interface”.

A parte do *Viewer* (Figura 4, marcado em vermelho) apresenta ao usuário a visualização da aplicação e é o local onde ocorre o desenvolvimento da interface da aplicação.

Basta selecionar um item na paleta e arrastar para a figura da aplicação, que este será adicionado. O usuário monta a interface por meio dos itens disponíveis e conforme as necessidades da aplicação. Ao lado tem-se a lista com os componentes utilizados e o nome (com a opção de ser alterado ou excluído) de cada um, e ao selecionar um deles, o usuário tem acesso às propriedades do item (Figura 4, marcado em azul).

As propriedades dos itens usados apresentam informações, tipos, dimensões, tamanho de fonte, etc., com a opção de se alterar diversos aspectos e funcionalidades. Por exemplo, pode-se alterar a cor da aplicação; adicionar alguma imagem como plano de fundo; alterar cores, textos, tamanho da fonte dos *Labels*; alterar o texto de fundo das caixas de texto, *placeholder*; etc. (Figura 4, marcado em verde).

Na fase de programação do AIA (Figura 5), a ferramenta dispõe, do lado esquerdo os grupos de blocos básicos e os relacionados com os itens da aplicação adicionados ainda na fase da interface. Este é um grande diferencial do AIA para os demais, o usuário terá uma série de funcionalidades para os componentes da aplicação por ele adicionados. Os blocos básicos são divididos em: *Control*, *Logic*, *Text*, *Lists*, *Colors*, *Variables* e *Procedures*, e contém peças relativas a “If else”, “While do”, operadores lógicos, operações matemáticas, comparações, *Strings*, números, cores, vetores etc. Há uma gama de peças que permitem a programação da aplicação.

A questão do AIA dar um feedback somente com a execução do aplicativo criado em um dispositivo móvel ou emulador pode ser um aspecto negativo, pois o usuário não tem como testar o algoritmo antes de “salvar” e definir a aplicação.

2.2 Scratch

O *layout* do Scratch é dividido em três partes (Figura 6). Da esquerda para a direita, uma contendo os atores até então, utilizados no algoritmo, normalmente o gatinho já tradicional da ferramenta. Ele interage com o usuário conforme o algoritmo desenvolvido, podendo falar, perguntar e captar informações, além de realizar movimentos diversos e apresentar resultados finais. Os atores podem ser alterados e novos podem ser adicionados, permitindo ao usuário a construção de alguma animação, por exemplo.

No centro da tela da ferramenta (Figura 6, marcado em vermelho) há os grupos de blocos de programação e as abas para alterar dinâmicas dos atores, com a opção de criar e usar os sons disponíveis. Há a opção de criar variáveis e vetores, blocos para uso de sons,

para perguntar e captar respostas do usuário, criar movimentos, etc. Ao lado direito, o local onde o usuário montará sua sequência lógica referente ao algoritmo. Os blocos são encaixados, assim como no MIT App Inventor (Figura 6, marcado em azul).

Partes da Interface com Usuário - Scratch

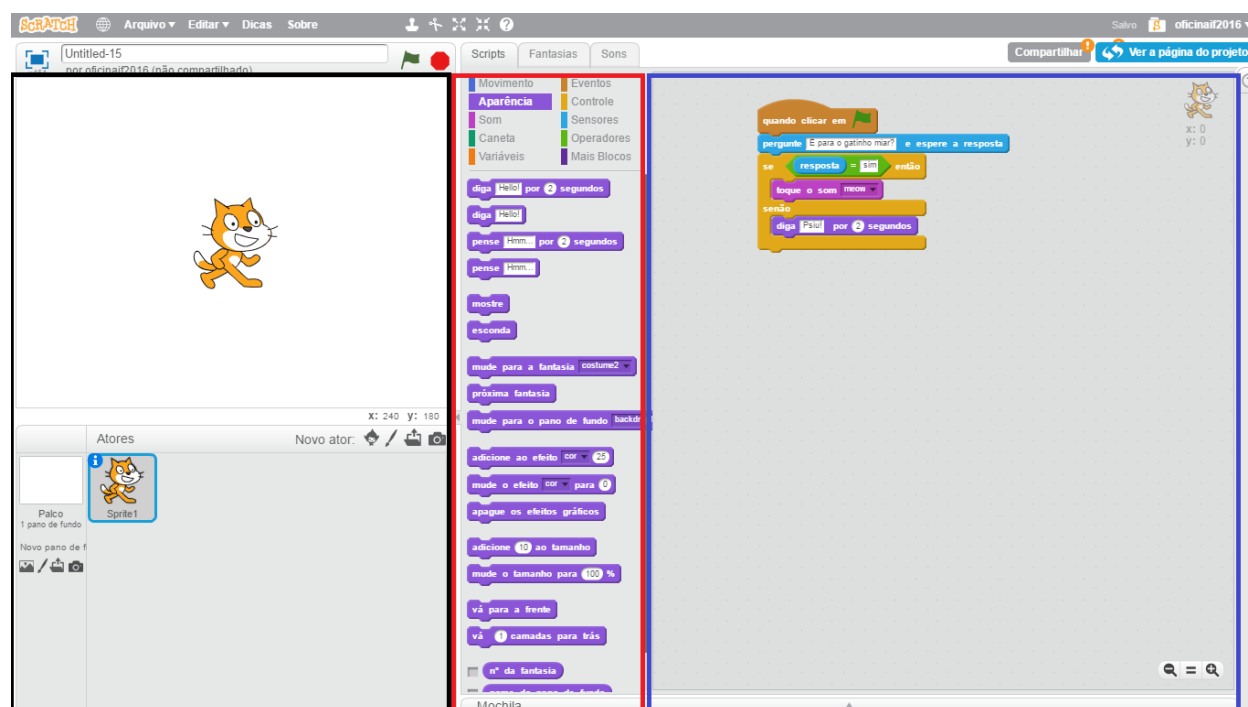


Figura 6 – Destaque das partes do Scratch: Atores (em preto), opções de blocos (em vermelho) e espaço para programar (em azul). **Fonte:** próprio autor.

Um aspecto do Scratch é que se trata de uma ferramenta mais voltada para a criação de animações, até por se tratar de uma plataforma voltada para indivíduos na faixa etária entre 8 e 16 anos. Embora possa se adequar a proposta de execução de algoritmos que seguem a linha mais tradicional estrutura, como se aprende inicialmente.

2.3 Oficina de aprendizagem de lógica de programação

Foi elaborada uma oficina voltada para ensinar pessoas a utilizarem as ferramentas: Scratch e MIT App Inventor. O objetivo foi identificar qual ferramenta os participantes da oficina preferiram. Uma vez, obtida a resposta, pode-se sugerir o uso da

ferramenta preferida como uma abordagem para a introdução de lógica de programação.

2.3.1 Descrição da organização da oficina

Para a realização da oficina foram utilizadas as dependências do Instituto Federal de Ciências, Ensino e Tecnologia de Goiás, Campus Inhumas. Foram três momentos, o primeiro realizado com alunos do ensino superior e médio. Para tal a oficina fez parte da grade de programação do evento SECITEC, quinta edição, ocorrida no dia 20 de outubro de 2016. O segundo foi realizado na casa de um dos autores, com um grupo de pessoas com formações diversas, no dia 17 de novembro de 2016 (Figura 7); e o terceiro momento foi durante a Semana de acolhimento aos calouros de informática, no dia 17 de fevereiro de 2017, realizada com alunos do ensino médio técnico em informática (1º ano).

A abordagem da oficina consiste na demonstração prática de programação utilizando as duas ferramentas. Foi desenvolvido um exemplo simples para ser utilizado, uma calculadora de IMC (Índice de Massa Corporal). O exemplo abordou a inserção de valores para variáveis, um cálculo aritmético e uso de condições (if e else), que são elementos básicos e iniciais do aprendizado de lógica e programação.

A execução da oficina ficou dividida em três partes: no primeiro momento os participantes foram informados a respeito do presente trabalho, sua justificativa e motivação. Na segunda parte era demonstrado e ensinado o uso do AIA e Scratch. No terceiro momento, foi feito o desenvolvimento da calculadora de IMC.

Na primeira e terceira oficina o exemplo era feito em uma ferramenta tendo o acompanhamento de todos os participantes, que reproduziam em seus computadores, o algoritmo. Na segunda oficina, isso não foi possível, e nenhum dos participantes deste momento, chegou a de fato fazer o algoritmo estudado. É importante salientar que o Scratch foi sempre a primeira ferramenta apresentada nas oficinas.

Os conceitos foram explicados conforme apareciam no algoritmo, por exemplo: ao ser necessário criar uma variável, o responsável pela oficina fazia uma breve explicação do funcionamento desse elemento. Em seguida, o mesmo exemplo foi feito em outra ferramenta como a mesma preocupação em explicar os recursos e elementos necessários. As peculiaridades Scratch e MIT App Inventor foram descritas seguindo o mesmo preceito. Por fim, era aplicado um questionário desenvolvido para os alunos para darem um *feedback* sobre a qualidade da oficina e aspectos principalmente ligados à comparação entre as ferramentas.

Todos os participantes envolvidos nas oficinas responderam o questionário.

2ª Apresentação da oficina



Figura 7 – Segundo momento das oficinas na casa de um dos autores, com um grupo de pessoas com formações diversas. **Fonte:** próprio autor.

2.3.2 Algoritmo utilizado nas oficinas

Conforme relatado anteriormente, o problema utilizado nas oficinas foi o de cálculo de índice de massa corporal. O algoritmo abaixo representa seu funcionamento;

- 1) Solicitar e armazenar o peso do usuário;
- 2) Solicitar e armazenar a altura do usuário;
- 3) Calcular o IMC e armazenar o resultado;
- 4) Exibir o resultado;
- 5) Verificar a condição do usuário de acordo com o resultado;
- 6) Exibir ao usuário a sua condição;
- 7) Encerrar.

O processo de elaboração dos algoritmos feitos no Scratch e no AIA apresentaram algumas diferenças razoáveis, embora os princípios lógicos fossem os mesmos. A diferença é que no MIT App Inventor o usuário programa aquilo que será a interface de uma aplicação e somente depois o algoritmo em si, dadas as especificidades da ferramenta, pois é voltada para a criação de *App*. Os dados necessários são solicitados em caixas de textos, como em formulários WEB tradicionais. Já no Scratch não há essa divisão, o usuário utiliza blocos para solicitar os dados necessários. É importante mencionar que no Scratch não tem bloco único para “else if”.

Na figura 8, é possível observar o fluxograma do algoritmo para as duas ferramentas utilizadas nas oficinas.

Fluxogramas do Algoritmo de IMC com o MIT App Inventor e o Scratch

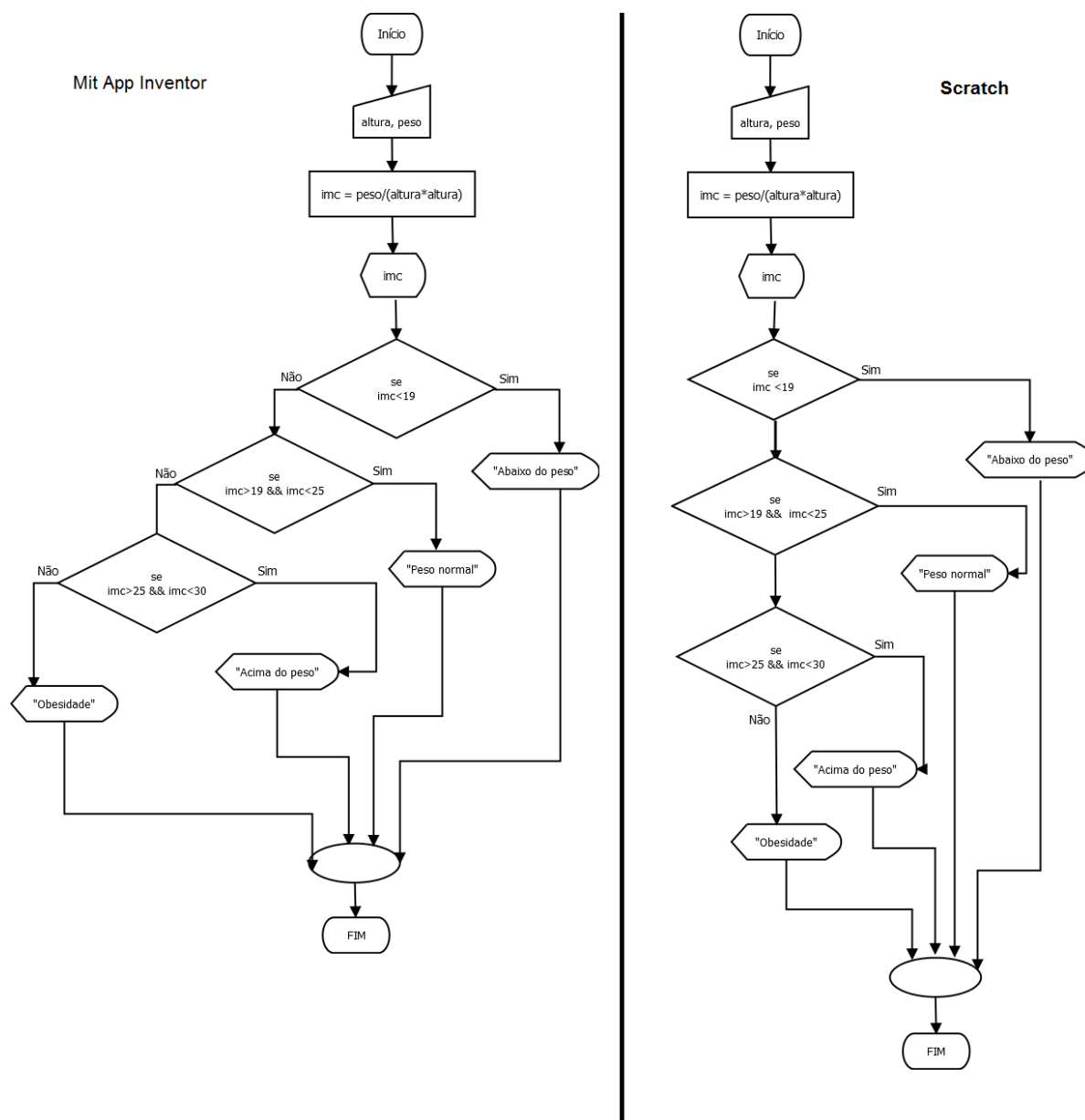
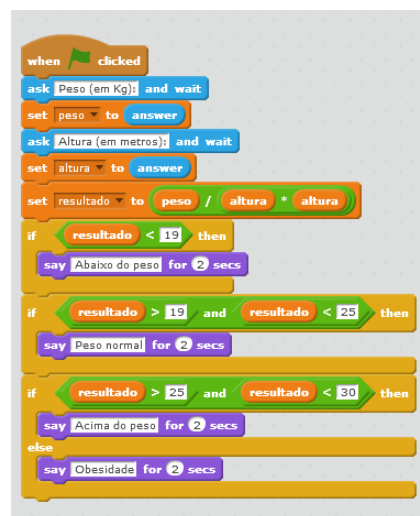
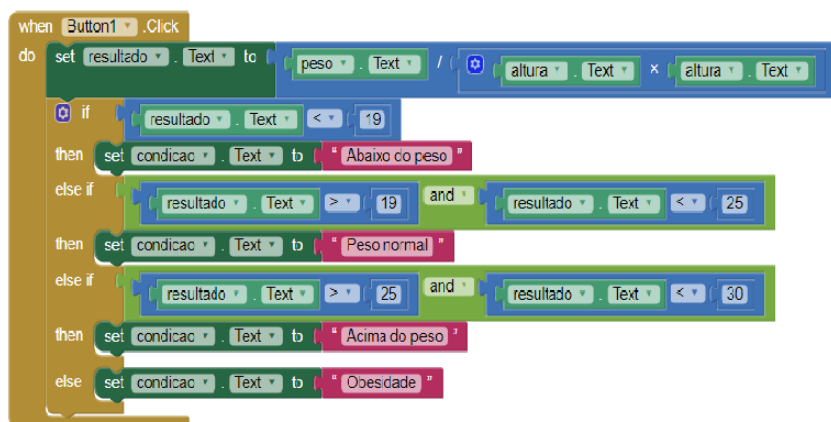


Figura 8 – Fluxogramas do algoritmo para cálculo de IMC, desenvolvidos nas duas ferramentas analisadas. **Fonte:** próprio autor.

Como já mencionado, a forma de programar no AIA difere da empregada no Scratch. Entretanto houve uma similaridade muito grande na montagem dos blocos, de certa maneira, justificado pelo parentesco com o Blockly, das duas ferramentas. Os fluxogramas dos algoritmos (Figura 8) indica a semelhança, pois basicamente, tem-se os mesmos itens para as

duas ferramentas. Isso pode ser verificado no resultado do conjunto de blocos, referente aos algoritmos, do Scratch e do AIA (Figura 9).

IMC – AIA x Scratch



Figuras 9 – Comparação entre blocos usados para a construção do algoritmo de IMC, nas ferramentas MIT App Inventor e Scratch. Fonte: próprio autor.

2.3.3 Questionário

Foi desenvolvido um questionário com o intuito de obter informações dos discentes participantes das oficinas realizadas. Como pode ser visto no Apêndice A. Baseou-se no trabalho realizado por Pacheco e Ferreira (2016) para uma abordagem leve e objetiva que resultasse em uma quantidade de informações suficientes para o avanço do estudo.

O objetivo do questionário era separar em perfis, baseando-se no conhecimento e entendimento de programação de computadores, e obter uma resposta sobre a preferência a respeito das ferramentas utilizadas na oficina. As perguntas eram claras e em sua maioria, com respostas de múltipla escolha para permitir uma melhor análise dos dados. As perguntas discursivas tinham como objetivo permitir a identificação do interesse sobre as ferramentas de programação usadas no trabalho.

3. RESULTADOS

Os valores descritos a seguir são resultados de todas as oficinas aplicadas, totalizando trinta e nove pessoas. Desse total, foram divididas em grupos conforme característica definida pelo próprio participante. Adota-se a recorrente divisão baseada na experiência do indivíduo com programação de computadores: “não teve contato com programação”, “contato inicial”, “intermediário” e “avançado”. Essa divisão é feita tendo como base uma questão do próprio questionário, onde o participante se classifica de acordo com este critério.

Em relação à usabilidade da ferramenta, de acordo com o exemplo demonstrado, perguntou-se: “No exemplo para cálculo do IMC qual ferramenta você considera mais fácil o manuseio e criação do programa”, do questionário (Apêndice A). A maioria, totalizando vinte e uma pessoas, preferiram o Scratch, enquanto dezesseis acharam melhor o MIT App Inventor, e duas pessoas não gostaram de nenhuma das ferramentas (Figura 10).

Preferência das ferramentas quanto à usabilidade

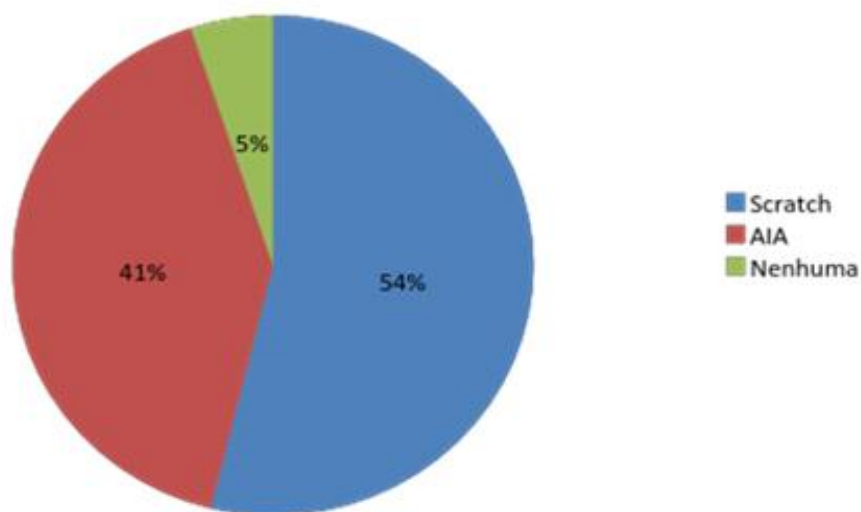


Figura 10 – Preferência de acordo com o exemplo de IMC trabalhado na oficina. **Fonte:** próprio autor.

É interessante notar que, dentre as pessoas que tiveram contato e prática de programação (Intermediário e avançado) houve uma preferência maior pela ferramenta do MIT

em detrimento da Scratch seis (AIA), quatro (Scratch) em um total de dez (Tabela 1). Considera-se a opinião desse grupo baseada em suas noções prévias de programação e a percepção de quais recursos uma ferramenta de programação deve possuir. Entretanto, é necessário reafirmar que o presente trabalho está voltado principalmente para as pessoas que não tem conhecimento e prática de programação de computadores.

Tabela 1 – Preferências de usabilidade de acordo com o contato/prática com programação de computadores

Contato/Prática	AIA	Scratch	Nenhuma
Não teve contato	3	5	0
Teve contato	7	12	2
Intermediário/Avançado	6	4	0

Fonte: próprio autor.

Em relação aos outros grupos “não teve contato” e “contato inicial”, correspondente às pessoas que nunca programaram ou nem tinham conhecimento a respeito de linguagens de programação, e as pessoas que já tinham uma noção do que se tratava, as respostas do primeiro grupo indicaram uma preferência pelo Scratch em detrimento ao AIA, com resultado de cinco (Scratch) para três (AIA). Quanto ao grupo dos que possuem conhecimento básico, ficou da seguinte maneira: sete (AIA), doze (Scratch) e duas reprovaram ambas ferramentas (Figura 11).

Tabela 2 – Preferência de ferramenta de acordo com o contato/prática de programação

Nível quanto experiência com programação	AIA	Scratch	Total
Não tiveram contato	3	5	8
Contato inicial	8	13	21
Intermediário	7	2	9
Avançado	1	0	1
Total	19	20	39

Fonte: próprio autor.

Análise da preferência de ferramenta de acordo com o contato/prática de programação

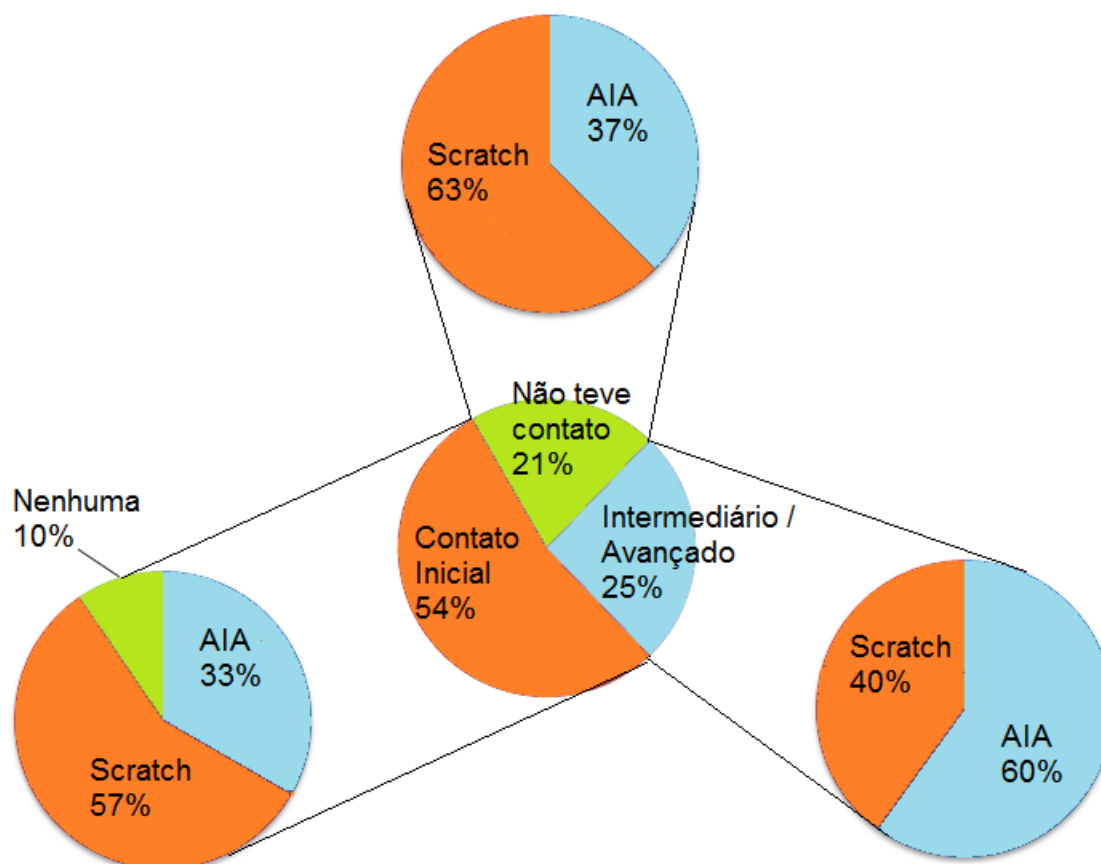


Figura 11 – Porcentagens de preferência de ferramenta para os três perfis analisados: Não teve contato com programação, teve contato inicial e nível intermediário/avançado. **Fonte:** próprio autor.

Quanto à preferência/afinidade pessoal, foi perguntada qual foi a ferramenta preferida por indivíduo. Os resultados foram: vinte pessoas preferiram o Scratch e dezenove o AIA. Quanto à análise por grupos de acordo com experiência em programação, abordada e explicada anteriormente, tem-se: nos que não tiveram contato com a programação a disposição foi cinco x três para o Scratch; conhecimento básico, treze x oito para o Scratch; intermediário, sete x dois para o AIA; e avançado contando com apenas uma pessoa que optou pelo AIA (Tabela 2).

Com interesse na resposta das pessoas para a oficina foi realizada a seguinte pergunta (treze do questionário): “Qual sua sugestão para que a oficina fique mais interessante

e que nos ajude a despertar o interesse do uso das ferramentas?” (Apêndice A). As respostas indicaram uma boa aceitação do modo que o trabalho foi realizado. E quanto às sugestões, propriamente ditas, muitas indicavam preocupações em tornar a oficina, de fato, em algo muito interessante e longo, indicando um bom *feedback* por parte da amostragem. Algumas respostas indicaram um desejo de dar sequência ao projeto, ou mesmo o uso mais desafiador das ferramentas empregadas (Quadro 4).

Quadro de sugestões para que a oficina fique mais interessante, despertando o interesse do uso das ferramentas em horas vagas

Aluno	Resposta
A	“Essa oficina tem que ir ao contato de outros alunos mostrando como é fácil e divertido programar.”
B	“Eu já achei ela muito interessante. Não tenho idéia como fazer ela mais interessante ainda.”
C	“Pra mim nao precisa mudar nada.”
D	“Haver mais oficinas.”
E	“Uma programação mais complexa.”
F	“Para mim está ótimo.”
G	“Criar desafios interagindo com os amigos.”
H	“Pedir para os participantes levarem celulares para interagir com o programa criado.”

Quadro 4 - Quadro com algumas das sugestões para próximas oficinas. **Fonte:** próprio autor.

Qual o motivo que te levou a participar dessa oficina?

39 responses

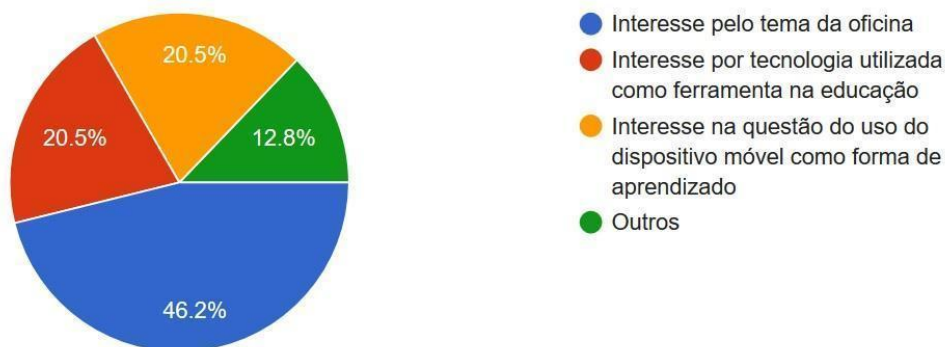


Figura 12 – Motivação do público em relação a oficina. **Fonte:** próprio autor.

A análise da motivação que levou as pessoas a participarem da oficina indicou que o público demonstra grande interesse quanto à busca por aprendizagem e o uso de ferramentas computacionais e dispositivos móveis como um recurso lúdico de ensino (Figura 12). Por se tratar de aparelhos eletrônicos que as pessoas cada vez mais possuem e utilizam, os dispositivos móveis foram lembrados como uma poderosa ferramenta. Possivelmente, indicando uma junção das duas ideias com maiores chances, devido à motivação pelo tema da oficina “Aprender o básico do Scratch ou MIT App Inventor?”.

Conhecimento das ferramentas

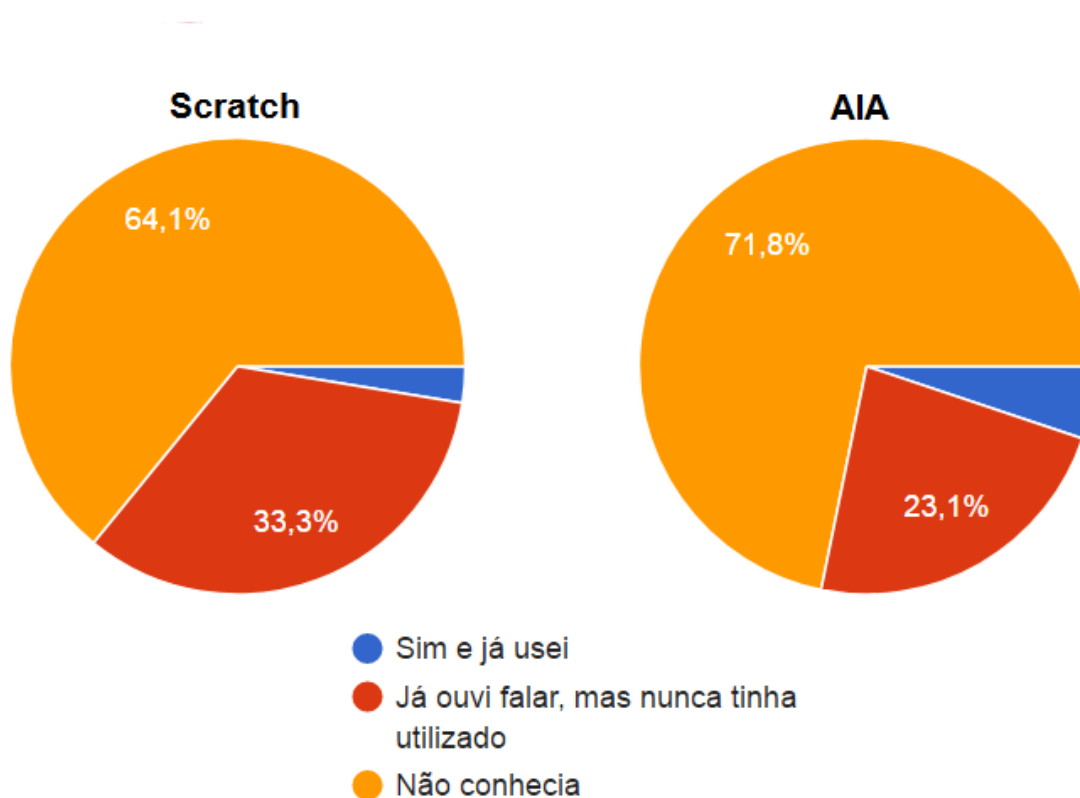


Figura 13 - Popularidade das ferramentas de acordo com o questionário. **Fonte:** próprio autor.

Quanto ao conhecimento das ferramentas usadas, foi perguntado o grau de familiaridade de cada participante com o Scratch e MIT App Inventor (Figura 13). Tratava-se de duas perguntas de múltipla escolha, com as opções: conhecimento e uso; conhecimento e não uso; ou desconhecimento. O parâmetro avaliado, nesta questão, oferece-nos um indicativo da

popularidade. Assim como acontece nas linguagens de programação codificadas, onde algumas são mais populares, conseqüentemente com mais usuários ativos e mais materiais desenvolvidos.

Por fim, tem-se uma análise de quatro perguntas de múltipla escolha, que solicitam respostas positivas a respeito de uma das ferramentas em detrimento da outra, com exceção de uma, que tinha como uma de suas alternativas a opção de selecionar “as duas”. Para a avaliação das respostas, foi utilizado o método de valoração com pesos, de forma que deu-se um peso para Scratch ou AIA de acordo com a resposta. O peso usado foi um, para cada vez que alguém optasse por uma das alternativas em uma das quatro perguntas avaliadas, seria acrescido o peso ao valor do somatório individual da ferramenta, por ex.: três respostas afirmativas para AIA, peso três ($1+1+1$), uma para Scratch, peso (1).

Considerou-se para a resposta da pergunta vinte e dois: “Considere que um professor pretende utilizar uma das ferramentas para introduzir seus alunos em programação. Qual das ferramentas, você julga ser mais apta para isso?”. Foi permitido ao usuário responder “as duas ferramentas”, portanto, soma-se a pontuação um para cada ferramenta, quando assim for respondido. Utilizando teoria de conjuntos, subtraiu do resultado a intersecção ($AIA \cap Scratch$). A intersecção é um subconjunto que dois conjuntos ou mais compartilham e com a subtração de conjuntos, descarta este conjunto, para que o mesmo não seja duplicado na análise dos dados. No nosso cenário, uma resposta positiva que englobe duas respostas.

Continuando o processo, chegou-se ao peso que indica o grau de aceitação de Scratch e AIA, oitenta e um e sessenta respectivamente. O quadro seis contém uma representação da somatória de pesos para cada aluno de acordo com a metodologia descrita. Isso leva ao valor geral de 35% de preferência a mais pelo Scratch em relação ao MIT App Inventor.

As questões utilizadas para montagem do quadro de pesos estão descritas a seguir para facilitar a leitura do texto.

Questão vinte: “Você gostou mais de qual das ferramentas apresentadas?”.

Questão vinte e dois: “Considere que um professor pretende utilizar uma das ferramentas para introduzir seus alunos em programação. Qual das ferramentas, você julga ser mais apta para isso?”.

Questão vinte e três: “No exemplo para cálculo do IMC qual ferramenta você considera mais fácil o manuseio e criação do programa?”.

Questão vinte e cinco: “Em qual das ferramentas você compreendeu melhor os métodos de inicialização das variáveis?”.

Quadro de análise de dados

Questão 20	Questão 22	Questão 23	Questão 25	AIA	Scratch	Intersecção
MIT App Inventor	As duas, pois são bastante similares	MIT App Inventor	Scratch	3	2	1
MIT App Inventor	MIT App Inventor	MIT App Inventor	Scratch	3	1	0
MIT App Inventor	MIT App Inventor	MIT App Inventor	MIT App Inventor	4	0	0
MIT App Inventor	MIT App Inventor	MIT App Inventor	MIT App Inventor	4	0	0
MIT App Inventor	As duas, pois são bastante similares	MIT App Inventor	Scratch	3	2	1
MIT App Inventor	Scratch	Scratch	MIT App Inventor	2	2	0
MIT App Inventor	As duas, pois são bastante similares	MIT App Inventor	MIT App Inventor	4	1	1
MIT App Inventor	As duas, pois são bastante similares	MIT App Inventor	MIT App Inventor	4	1	1
Scratch	Scratch	Scratch	Scratch	0	4	0
Scratch	Scratch	Scratch	Scratch	0	4	0
Scratch	Scratch	MIT App Inventor	Scratch	1	3	0
Scratch	Scratch	Scratch	Scratch	0	4	0
Scratch	As duas, pois são bastante similares	Scratch	Scratch	1	4	1

MIT APP INVENTOR	MIT App Inventor	Scratch	MIT App Inventor	3	1	0
Scratch	Scratch	Scratch	Scratch	0	4	0
Scratch	As duas, pois são bastante similares	Scratch	Scratch	1	4	1
MIT App Inventor	As duas, pois são bastante similares	MIT App Inventor	MIT App Inventor	4	1	1
Scratch	MIT App Inventor	Scratch	Scratch	1	3	0
MIT App Inventor	As duas, pois são bastante similares	MIT App Inventor	MIT App Inventor	4	1	1
Scratch	MIT App Inventor	MIT App Inventor	MIT App Inventor	3	1	0
MIT App Inventor	MIT App Inventor	MIT App Inventor	MIT App Inventor	4	0	0
MIT App Inventor	MIT App Inventor	MIT App Inventor	MIT App Inventor	4	0	0
MIT App Inventor	As duas, pois são bastante similares	Scratch	Scratch	2	3	1
MIT App Inventor	MIT App Inventor	MIT App Inventor	Scratch	3	1	0
Scratch	Scratch	Scratch	Scratch	0	4	0
Scratch	Scratch	Scratch	Scratch	0	4	0
Scratch	As duas, pois são bastante similares	Scratch	Scratch	1	4	1
MIT App Inventor	As duas, pois são bastante similares	Scratch	Scratch	2	3	1
Scratch	Scratch	Scratch	Scratch	0	4	0

MIT App Inventor	As duas, pois são bastante similares	MIT App Inventor	MIT App Inventor	4	1	1
Scratch	As duas, pois são bastante similares	Scratch	Scratch	1	4	1
Scratch	Scratch	Scratch	Scratch	0	4	0
MIT App Inventor	MIT App Inventor	Scratch	MIT App Inventor	3	1	0
Scratch	As duas, pois são bastante similares	Scratch	Scratch	1	4	1
MIT App Inventor	Scratch	MIT App Inventor	Scratch	2	2	0
Scratch	As duas, pois são bastante similares	Nenhuma	Scratch	1	3	0
Scratch	As duas, pois são bastante similares	Scratch	Scratch	1	4	1
Scratch	As duas, pois são bastante similares	Nenhuma	Scratch	1	3	0
Scratch	Scratch	Scratch	Scratch	0	4	0
Total				75	96	15
Total - intersecção				60	81	

Quadro 5 - Representação do processo feito para analisar os dados das quatro perguntas.

Fonte: próprio autor.

Utilizando os dados da última análise, buscou-se dividir as pessoas em grupos e avaliá-las de acordo com essa divisão. Foram utilizados os perfis quanto ao conhecimento e prática de programação: não teve contato com programação; contato inicial; intermediário/avançado.

Para a obtenção de um valor referencial, foi utilizado o seguinte método: analisou-se qual ferramenta foi preferida por aluno e por qual margem, isto é, a quantidade de respostas

positivas que uma ferramenta teve a mais que a outra para cada pergunta. Assim foram separadas as pessoas por grupo, obteve-se a somatória dessas margens para Scratch e MIT App Inventor de acordo com os perfis definidos. Dessas somatórias, foi feita a razão do valor maior pelo menor para cada ferramenta, a fim de obter um valor que, indicasse percentualmente, o grau que uma supera a outra (Tabela 3). É importante salientar que foram usadas nessas análises todos as pessoas da amostragem.

De forma simplificada, contava-se a quantidade de vezes que as ferramentas eram citadas para cada resposta das quatro perguntas, verificava qual foi mais citada e quantas vezes mais que a outra. Por fim, somava essas margens para um dos três grupos de perfis relacionados com o conhecimento e prática de programação. O processo é similar e foi inspirado na contagem de saldo de gols de clubes de futebol, a ideia era traduzir os dados em números que nos indicassem com precisão uma conclusão para a comparação entre as ferramentas e ainda apresentasse resultados para os perfis de forma mais específica, como os valores obtidos para cada grupo da amostragem.

Tabela 3 – Dados comparativos entre as ferramentas

Perfil	AIA	Scratch	Razão (maior/menor)	Porcentagem (%) = Razão - 1
Não teve contato	24	37	Scratch / AIA = 1,54	54
Contato Inicial	5	18	Scratch / AIA = 3,6	260
Intermediário /Avançado	14	9	AIA / Scratch = 1,55	55

Fonte: próprio autor.

Portanto, tem-se um valor que representa o resultado para a análise geral, considerando a amostragem em sua totalidade. E outra análise que divide os resultados entre os perfis de contato/prática de programação, permitindo uma identificação de qual ferramenta se sobressaiu, de forma mais específica.

Os dados apresentados na Tabela 3 indicam que para o perfil referente às pessoas que não tiveram contato com programação, o Scratch é superior ao AIA com a porcentagem de 54%. Quanto aos que tiveram contato inicial com a prática de programação, nota-se uma preferência 3 vezes maior pelo Scratch. Verifica-se uma preferência de 55% pelo AIA em relação ao Scratch, para o público intermediário/avançado.

4. CONCLUSÃO E TRABALHOS FUTUROS

4.1 Conclusão

Como já mencionado, estudos na área pedagógica fundamentam o lúdico como um meio de se promover o processo educacional. Vygotsky e Piaget introduzem a ideia de que jogos, ferramentas lúdicas e interações são bases do ensino-aprendizado. A aplicação deles como métodos de ensino podem trazer excelentes resultados. E a escolha da ferramenta mais adequada é primordial neste processo.

Os estudos apresentados corroboram a importância da lógica, devido estar presente tanto nas formações de ensino superior das áreas de exatas, como em tarefas presentes no cotidiano, como traçar uma rota para chegar em casa por exemplo. O lúdico presente nas TIC's vem como uma potencial possibilidade de incentivo ao desenvolvimento das habilidades lógicas. Qualquer função que um ser humano for desempenhar é fundamental que o mesmo planeje os passos de forma lógica e coerente para obter seu resultado esperado. O MIT App Inventor e Scratch representam novos meios de iniciar o ensino de programação e lógica com este forte componente, o lúdico, comprovadamente um grande motivador para o indivíduo adquirir conhecimento.

O presente trabalho também tratou de equiparar duas ferramentas de programação em blocos, até então mais conhecidas pelos graduandos e indicadas por diversos professores como recurso para o ensino nas salas de aulas. Como demonstrado na tabela 3, o Scratch apresentou uma resposta bem superior ao MIT APP Inventor. O público da amostragem propiciou à primeira ferramenta uma preferência de 35%, estatisticamente uma proporção aproximada de sete para cinco.

Os valores finais (Tabela 3) indicam uma melhor resposta para o Scratch em dois dos três perfis, sendo que em um deles, o resultado aponta para uma diferença de aproximadamente quatro vezes. Fortalecendo a ideia de como essa ferramenta foi bem avaliada pelas pessoas com menor grau de conhecimento e prática. O MIT App Inventor foi na contramão disso, apresentou melhor resposta com o público mais intermediário/avançado..

Atribuiu-se ao MIT App Inventor a preferência maior por um público que tem compreensão de programação, pelos motivos de se tratar de uma plataforma com diversos recursos para dispositivo móvel, que exige certo grau de capacidade de abstração de dados para a construção de aplicações. Já os resultados do Scratch apontam a simplicidade da abordagem da ferramenta à prática de programação, sem grandes exigências prévias de conhecimento de estruturas de dados e captura de informações, ao contrário da outra ferramenta. Um ponto importante a se destacar é que o Scratch foi a primeira ferramenta a ser

apresentada nas oficinas, o que pode ter gerado uma melhor aceitação.

4.2 Trabalhos Futuros

Prosseguir com uma proposta de metodologia de ensino com ferramentas de programação em blocos, com execução de oficinas continuadas, de forma a ter-se mais tempo e um maior número de momentos voltados para a prática e estudos do conteúdo. A questão do tempo é importante para verificar com maior precisão a eficácia da proposta. Como parte da sugestão mencionada, sugere-se também a possibilidade de acompanhar durante um período letivo uma turma do primeiro ano de algoritmos a fim de verificar a evolução nas notas, ou uma turma de ensino médio que tenha disciplinas relacionadas. Fica como possibilidade um estudo que envolva as ordens, quantidade, formas, disposição, apreensão na inserção e assimilação das ferramentas selecionadas, visto que após aplicação do presente trabalho foi notada a dispersão de determinados alunos do conteúdo apresentado por consequência do tempo das oficinas, conseguindo assim maior participação nos inícios das atividades.

REFERENCIAL BIBLIOGRÁFICO

AIBA, C., ROIG, R. y BERNAL, C. (2011). **Retos de la educación especial en el mundo digital**. En A. Sánchez Palomino, C. Berna, J. M. Granados, R. Gutiérrez, A. Luque, M. N. Lázaro, y L. Ortiz (Eds.). *Educación Especial y Mundo Digital* (pp. 114-124). Almería: Ediciones Universidad de Almería.

ALONSO, C. M.; GALLEGO, D. J.; HONEY, P. **Los estilos de aprendizaje: procedimientos de diagnóstico y mejora**. Madrid: Mensajero, 2002.

AMARAL S. F., Barros. D. M. V. **Estilos de Aprendizagem no contexto educativo de uso das tecnologias digitais interativas**. 2007. Disponível em: <http://www.lantec.fe.unicamp.br/lantec/pt/tvdi_portugues/daniela.pdf>. Acesso em: 20 janeiro 2017.

AUSUBEL, David P., Novak, Joseph, D. & Hanesian, Helen (1978). **Educational psychology: a cognitive view**. 2ª ed. New York: Holt, Rinehart and Winston. 733p.

BATISTA, Nailson dos Anjos. **O Ensino da Matemática na Educação Infantil através das Atividades Lúdicas**. 2012. 29f. Grupo Educacional Uninter, Macapá.

BENIGNO, V., Bocconi, S. y Ott, M. (2007). **Inclusive education: helping teachers to choose ICT resources and to use them effectively**. *Elearning Paper*, 6. Recuperado de <http://www.itd.cnr.it/download/inclusive_education.pdf>. Acesso em: 15 outubro 2016.

BOIKO, V.A.T. e ZAMBERLAN, M.A.T. **A Perspectiva Sócio-construtivista na Psicologia e na Educação: o brincar na pré-escola**. Maringá: Psicologia em Estudo, vol.6, p. 51-58, 2001. Disponível em: <<http://www.scielo.br/pdf/pe/v6n1/v6n1a07.pdf>>. Acesso em: 10 outubro 2016.

BORGES, M. A. F. (2000). **Avaliação de uma Metodologia Alternativa para a Aprendizagem de Programação**. VIII Workshop de Educação em Computação – WEI 2000. Curitiba, PR, Brasil.

CACEBRO, J. y Llorente, M. C. (2006). **Capacidades tecnológicas de las TICs por los estudiantes**. *Enseñanza & Teaching: Revista Interuniversitaria de Didáctica*, 24, 159-175.

CASTRO, C. T., CASTRO Jr, A., MENESES, C., B., M. e RAUBER, M. (2003). **Utilizando Programação Funcional em Disciplinas Introdutórias de Computação**. In: XI Workshop de Educação em Computação – WEI, Campinas/SP.

CHADHA, K.; TURBAK, F. **Improving MIT App Inventor usability via conversion between blocks and text**, *Journal of Visual Languages & Computing*, USA: Lexington, 2014, Pages 1042-1043.

CHAVES DE CASTRO, T., CASTRO JÚNIOR, A., MENEZES, C., BOERES, M. e RAUBER, M. **Utilizando Programação Funcional em Disciplinas Introdutórias de Computação**. XI Workshop de Educação em Computação – WEI 2003. Campinas, SP, Brasil.

COSTA, H.A.X.; SANTOS, R.P.; WERNER, C.M.L. (2010) **“Uma Análise do Processo de Ensino e Aprendizagem de Engenharia de Software: Desafios e Soluções no Contexto Brasileiro”**, In: Proceedings of the XI International Conference on Engineering and Technology

Education, Ilhéus, BA, Brasil, 367-371.

DELGADO, C., XEXEO, J. A. M., SOUZA, I. F., CAMPOS, M., RAPKIEWICZ, C. E. “**Uma Abordagem Pedagógica para a Iniciação ao Estudo de Algoritmos**”. XII Workshop de Educação em Computação (WEI'2004). Salvador, BA, Brasil.

DEMO, P. **Revista Brasileira de Docência, Ensino e Pesquisa em Educação Física** – ISSN 2175-8093 – Vol. 1, n. 1, p.53-75, Agosto/2009 Disponível em: <<http://www.pucrs.br/famat/viali/doutorado/ptic/textos/80-388-1-PB.pdf>>. Acesso em: 17 abril 2017.

DERENE, G. How to Raise a Tech-Savvy Kid. **Popular Mechanics**, 2013. Disponível em: <<http://www.popularmechanics.com/technology/how-to/tips/how-to-raise-a-techsavvy-kid-15528124>>. Acesso em: 01 março 2016.

DUDA, R., SILVA, S. de C. R. da, ZONTINI, D. D. e GROSSI, L. (2015). **Elaboração de aplicativos para Android com uso do MIT App Inventor**: uma experiência no Instituto Federal do Paraná – Câmpus Irati. In Revista Brasileira de Ensino de Ciência e Tecnologia, v. 8, n. 2, p. 115-128.

FERRANDIN, M. & STEPHANI, S. **Ferramenta para o ensino de programação via Internet**. I Congresso Sul Catarinense de Computação: UNESC. Criciúma, 2005. Disponível em : <<http://infociencia.info/artigos/ARTc21.pdf>>. Acesso em 05 outubro 2016.

GODOY, A. S. **Introdução à pesquisa qualitativa e suas possibilidades**. In: Revista de Administração de Empresas. São Paulo: v.35, n.2, p. 57-63, abril 1995.

GREGOLIN, Vanderlei Rodrigues. **LINGUAGEM LOGO: EXPLORANDO CONCEITOS MATEMATICOS**, Unesp, São Paulo: Araraquara, 2015.

HEINEN E., MASCHIO E., MARCZAL., LEALDINO P. F. **RASPIBLOCOS: Ambiente de Programação Didático Baseado em Raspberry Pi e Blockly**. Anais do XXVI Simpósio Brasileiro de Informática na Educação (SBIE 2015).

HONEY, P.; MUMFORD, A. **The Manual of Learning Styles**. London: McGraw Hill, 1986.

JÚNIOR et al, Garibaldi da Silveira. **Análise da Ferramenta de Programação Visual Blockly como Recurso Educacional no Ensino de Programação**. 2012. Disponível em: <<http://ug.iffarroupilha.edu.br/sabtic/files/4.pdf>>. Acesso em: 25 março 2017.

KOLB, D. **Psicología de las Organizaciones**. Experiencias. Madrid: Prentice Hall, 1976.

MIT App Inventor Home Page. [http:// appinventor.mit.edu/explore/](http://appinventor.mit.edu/explore/). Acesso em fev. 2016.

MARCHIORI, Eugenio J.; BLANCO, Angela del; TORRENTE, Javier; ORTIZ, Iván Martinez; MANJÓN, Baltasar Fernández. **A visual language for the creation of narrative educational games**, *Journal of Visual Languages & Computing*, USA: Boston, 2011, Pages 443-452.

MARQUES, A., Silva, B. D., & MARQUES, N. (2011). **A Influência dos Videojogos no Rendimento Escolar dos Alunos: Uma Experiência no 2º e 3º Ciclo do Ensino Básico**. *Educação, Formação & Tecnologias*, 4(1), 17-27 [Online], disponível em <<http://eft.educom.pt>>.

Acesso em: 20 outubro 2016.

MARQUÉS, P. (2010). ***Nueva cultura, nuevas competencias para los ciudadanos. La alfabetización digital. Roles de los estudiantes hoy.*** Recuperado de <<http://peremarques.pangea.org/competen.html>>. Acesso em: 09 setembro 2016.

MINAYO, M.C. de S. (Org.) **Pesquisa social: teoria, método e criatividade**. 22 ed. Rio de Janeiro: Vozes, 2003.

MIT App Inventor - Massachusetts Institute of Technology. Disponível em: <<http://appinventor.mit.edu/explore/about-us.html>>. Acesso em: 10 dezembro 2016.

NÁVRAT, P. **A closer look at programming expertise: critical survey of some methodological issues**. Information and Software Technology. Volume 38, Issue 1, 1996, páginas 37-46.

NOBRE, I. A. M. N., MENEZES, C. S. (2002). **Suporte à Cooperação em um Ambiente de Aprendizagem para Programação (SAmbA)**. XIII Simpósio Brasileiro de Informática na Educação – SBIE 2002. São Leopoldo, RS, Brasil.

NEVES, M.F.; COELLO, J.M.A. (2006) **“OntoRevPro: Uma Ontologia sobre Revisão de Programas para o Aprendizado Colaborativo de Programação Java”**, In: Anais do XVII Simpósio Brasileiro de Informática na Educação, Brasília, DF, Brasil, 569-578.

PAPERT, Seymour. **A máquina das crianças: repensando a escola na era da informática**. 2008.

PASTERNAK, E. (2009). **Visual programming pedagogies and integrating current visual programming language features**. Master's thesis, Robotics Institute, Carnegie Mellon University.

PEREIRA, J.C.R. Jr, RAPKIEWICZ, C. (2004) **“O Processo de Ensino-Aprendizagem de Fundamentos de Programação: Uma Visão Crítica da Pesquisa no Brasil”**, WEIRJES.

PACHECO, M. G. Jr, FERREIRA, S. A. **O ESTUDO DO CONFLITO ENTRE ISRAEL E PALESTINA ATRAVÉS DE UMA PROPOSTA METODOLÓGICA INTERDISCIPLINAR COM O JOGO PEACEMAKER**. Instituto Federal de Goiás. Inhumas, 2016.

PIAGET, J. **A psicologia da criança**. Rio de Janeiro. Bortrand, Brasil, 1998.

RODRIGUES, M. C. **Como Ensinar Programação?**. Informática - Boletim Informativo Ano I nº 01, ULBRA. Canoas, RS, Brasil. 2002

ROSSO, A. e DANIELE, M. (2000). **Our method to teach algorithmic development**. ACM SIGCSE Bulletin, Volume 32, Issue 2, ACM Press, USA, pp 49-52.

SANTOS, R., XEXÉO, J., RAPKIEWICZ, C., CAMPOS, M. e MACHADO, M. (2003). **Inovando a dimensão estratégica do ensino de Ciência da Computação**. Anais do WEI 2003, Brasil.

SANTOS, R. P. e COSTA, H. A. X. (2006) **Análise de metodologias e ambientes de ensino para algoritmos, estruturas de dados e programação aos iniciantes em computação e informática**. In: Infocomp Journal of Computer Science, vol.5, n.1, p.41-50.

SCHOTS, M.; SANTOS, R.P.; MENDONÇA, A.P.; WERNER, C.M.L. (2009) **“Elaboração de um Survey para a Caracterização do Cenário de Educação em Engenharia de Software no Brasil”**, In: Anais do II Fórum de Educação em Engenharia de Software, XXIII Simpósio Brasileiro de Engenharia de Software, Fortaleza, CE, Brasil, 57-60.

SCRATCH. **ABOUT Scratch** (Scratch Documentation Site). Disponível em: <http://info.scratch.mit.edu/About_Scratch>. Acesso em: 20 novembro 2016.

SILVA, C.E.M. (2007) **“A Importância dos Jogos Eletrônicos como Elemento de Apoio à Produção, Resgate e Valorização Cultural”**, In: Anais do XII Congresso de Ciências da Comunicação na Região Sudeste, Juiz de Fora, MG, Brasil.

SILVA, P. L. E. (2015) **“Análise do Scratch para ensino-aprendizagem de lógica de programação: estudo de caso no curso técnico em informática do Instituto Federal de Educação, Ciência e Tecnologia de Goiás, Campus Inhumas”**, Inhumas, GO, 2015.

VANDEVELDE, C., SALDIEN, J., CIOCCI, C., and VANDERBORGHT, B. (2013). **Overview of technologies for building robots in the classroom**. In International Conference on Robotics in Education, pages 122–130.

VYGOTSKY, Levi. **A formação social da mente**. São Paulo. Martins Fontes, 1989.

ZIEDE M. K. L., SILVA E. T. , PEGORARO L. , CANALLE E. M., SILVA A. O. M., CARVALHO A. F. W. **TECNOLOGIAS DIGITAIS NA EDUCAÇÃO BÁSICA: desafios e possibilidades**. 2016. Disponível em: <<http://seer.ufrgs.br/index.php/renote/article/view/70692>>. Acesso em 15 de setembro de 2017.

APÊNDICE

Apêndice A – Questionário

Questionário sobre o MIT APP INVENTOR E SCRATCH

O objetivo desse questionário é relatar suas primeiras experiências em contato com as plataformas de desenvolvimento em busca de verificar se foi possível aprender a programar por blocos em um curto espaço de tempo. Por favor sejam o mais sincero possível nas respostas. Agradecemos! (Há vidas em jogo! Huehue)

***Obrigatório**

1. Nome do participante *

2. Email *

3. Idade

4. Série

5. Você é a favor da utilização de ambientes visuais e lúdicos para ensino da programação de computadores?

Marcar apenas uma oval.

☐ Sim

☐ Não

☐ Acredito que não acrescentaria ao método de ensino

6. Já teve uma boa experiência com o uso de tecnologias para ensino em sala de aula?

7. Qual seu nível de conhecimento em programação?

Marcar apenas uma oval.

☐ Ainda não tenho

☐ Apenas contato inicial

☐ Intermediário

☐ Avançado

2017-6-26

Questionário sobre o MIT APP INVENTOR E SCRATCH

8. Como você enxergava a programação para aplicativos móveis antes da oficina?

9. Como você enxerga a programação para aplicativos móveis agora? Depois da oficina?

10. Qual foi sua maior dificuldade ao ter o primeiro contato com a ferramenta?

Marque todas que se aplicam.

- ☐ Falta de conhecimento no método utilizado (blocos)
- ☐ Falta de conhecimento teórico sobre a ferramenta
- ☐ Dificuldade com raciocínio lógico
- ☐ Desconhecimento de algoritmo

11. Foi encontrado alguma outra dificuldade além da marcada anteriormente?

12. Antes desta oficina, você já teve experiência com alguma ferramenta parecida?

Marcar apenas uma oval.

- ☐ Sim
- ☐ Não

13. Qual sua sugestão para que a oficina fique mais interessante e que nos ajude a despertar o interesse do uso das ferramentas em horas vagas?

2017-6-26

Questionário sobre o MIT APP INVENTOR E SCRATCH

14. Em qual área teve contato com a programação em blocos (método utilizado nas duas ferramentas trabalhadas)?

Marcar apenas uma oval.

- ☐ Exatas
- ☐ Humanas
- ☐ Biológicas
- ☐ Não tive experiência

15. Você conseguiria desenvolver os aplicativos? Se não, o que você não conseguiu entender ou utilizar nas ferramentas?

16. Você conhecia o MIT APP INVENTOR?

Marcar apenas uma oval.

- ☐ Sim e já usei
- ☐ Já ouvi falar, mas nunca tinha utilizado
- ☐ Não conhecia

17. Você conhecia o Scratch?

Marcar apenas uma oval.

- ☐ Sim e já usei
- ☐ Já ouvi falar, mas nunca tinha utilizado
- ☐ Não conhecia

18. Qual o motivo que te levou a participar dessa oficina?

Marcar apenas uma oval.

- ☐ Interesse pelo tema da oficina
- ☐ Interesse por tecnologia utilizada como ferramenta na educação
- ☐ Interesse na questão do uso do dispositivo móvel como forma de aprendizado
- ☐ Outros

19. Como está seu rendimento em sala de aula? Você se considera um aluno:

Marcar apenas uma oval.

- ☐ Excelente
- ☐ Muito bom
- ☐ Regular
- ☐ Fraco
- ☐ Péssimo

2017-6-26

Questionário sobre o MIT APP INVENTOR E SCRATCH

20. Você gostou mais de qual das ferramentas apresentadas?*Marcar apenas uma oval.*

- ☐ MIT APP INVENTOR
- ☐ Scratch
- ☐ Nenhuma

21. De acordo com a resposta da pergunta anterior. Quais elementos justificam sua escolha na preferência de uma ferramenta em detrimento da outra? Há elementos que você julga serem superiores na ferramenta de sua escolha?

22. Considere que um professor pretende utilizar uma das ferramentas para introduzir seus alunos em programação. Qual das ferramentas, você julga ser mais apta para isso?*Marcar apenas uma oval.*

- ☐ MIT App Inventor
- ☐ Scratch
- ☐ As duas, pois são bastante similares

23. No exemplo para cálculo do IMC qual ferramenta você considera mais fácil o manuseio e criação do programa?*Marcar apenas uma oval.*

- ☐ MIT App Inventor
- ☐ Scratch
- ☐ Nenhuma

24. Sobre o App Inventor. Ser uma ferramenta que produz aplicativos é um grande diferencial em relação ao Scratch? Como ferramenta de ensino de programação, isso tem grande relevância?

25. Em qual das ferramentas você compreendeu melhor os métodos de inicialização das variáveis?*Marcar apenas uma oval.*

- ☐ MIT App Inventor
- ☐ Scratch
- ☐ Nenhuma das duas

2017-6-26

Questionário sobre o MIT APP INVENTOR E SCRATCH

26. Considerando que uma variável x deva estar em um intervalo de valores maior que 3 e menor que 20.

Marcar apenas uma oval.

☐ if $((x < 3) \text{ and } (x > 20))$

☐ if $((x > 3) \text{ and } (x < 20))$

☐ if $((x > 3) \text{ or } (x < 20))$

27. Que nota você dá para a oficina?

Marcar apenas uma oval.

☐ 1

☐ 2

☐ 3

☐ 4

☐ 5

Powered by

 Google Forms

15	1º ano	Sim	Sim
15	1º ano	Sim	Sim
15	1º ano	Sim	Sim
15	1º ano	Sim	Sim
16	1º ano	Sim	Sim
15	1º ano	Sim	Sim
14	1º ano	Sim	Sim
16	1º ano	Sim	Sim
15	1º ano	Sim	Sim
15	1º ano	Sim	Sim, para pesquisas
15	1º ano	Sim	Não
14	1º ano	Sim	Sim
15	1º ano	Sim	Sim
17	1º ano	Sim	Sim
16	1º ano	Sim	Não
15	1º ano	Sim	Sim
15	1º ano	Sim	Não
15	1º ano	Sim	Sim
14	1º ano	Sim	Sim
13	1º ano	Sim	Nunca
14	1º ano	Sim	Sim
14	1º ano	Sim	Não

Questão 7	Questão 8
Intermediário	Complexa, pois a preparação do ambiente de desenvolvimento Android é cansativa

Intermediário	Bem complexo.
Intermediário	Sei que em outras ferramentas a programação é um pouco mais complexa, e via a programação para aplicativos móveis um pouco mais complicada
Intermediário	Pequenas coisas possuíam grau de programação um pouco mais avançadas
Apenas contato inicial	Muito mais complexa.
Intermediário	Complexa
Avançado	Enxergava como sendo uma prática alternativa aos métodos de programação convencionais, podendo ser um novo caminho, porém com menos adeptos devido a certo receio de se desviar do padrão.
Ainda não tenho	Pensava que programação para aplicativos móveis era mais fácil do que para um site, mas sua complexidade não é inferior a mesma
Ainda não tenho	Eu achava interessante, mas agora, depois da palestra, vi que é muito mais importante do que eu pensava
Ainda não tenho	Simples
Ainda não tenho	
Intermediário	Muito legal
Ainda não tenho	Muito complicada
Apenas contato inicial	Super difícil
Intermediário	Uma coisa boa
Apenas contato inicial	Complicada
Apenas contato	Eu pensava que era difícil de fazer programação

inicial	
Apenas contato inicial	Eu não tinha conhecimento
Apenas contato inicial	Eu nao sabia nada antes
Apenas contato inicial	Não tinha o conhecimento da programação de aplicativos móveis antes da oficina
Apenas contato inicial	Eu não tinha ideia de como funcionava a programação
Apenas contato inicial	Bom
Intermediário	Não sei, mas a oficina me ajudou bastante
Ainda não tenho	Achava que era mais fácil de ser feita
Apenas contato inicial	Boa
Ainda não tenho	Complicado
Apenas contato inicial	Da Mesma forma
Ainda não tenho	Algo simples sem conhecer a programação
Apenas contato inicial	Difícil e complicada, pois não havia muitos conhecimentos sobre este assunto
Apenas contato inicial	Como algo muito difícil

Apenas contato inicial	Complicada
Apenas contato inicial	Difícil
Apenas contato inicial	Eu acreditava ser muito complicada a criação de qualquer aplicativo móvel antes da oficina, mas percebi que não é bem assim
Apenas contato inicial	Muito interessante
Intermediário	Algo complicado e demorado
Apenas contato inicial	Não tinha a menor ideia
Apenas contato inicial	Legal e meio complicado
Apenas contato inicial	Interessante
Apenas contato inicial	Difícil

Questão 9	Questão 10
Para aplicações simples, atende muito bem as expectativas	Falta de conhecimento no método utilizado (blocos)
Que existe maneiras mais fáceis para criação de aplicativos móveis.	Falta de conhecimento teórico sobre a ferramenta
Agora sei que o desenvolvimento de aplicações básicas, que não envolvem um nível avançado de conhecimento em programação pode ser	Falta de conhecimento no método utilizado

desenvolvida de maneira fácil usando o MITT ou SCRATCH.	(blocos)
Após a oficina foi notório que a utilização desses aplicativos são de grande valia, pois podem transformar tarefas árduas em tarefas mais simples	Falta de conhecimento no método utilizado (blocos)
Com muita facilidade. Não imaginava que poderia fazer com tanta facilidade	Falta de conhecimento no método utilizado (blocos)
De outro prisma, com uma aceitação maior	Desconhecimento de algoritmo
A oficina deixou bem claro que é possível manter os métodos convencionais de programação porém para uma plataforma mais utilizada corriqueiramente	
Achei muito interessante. Da vontade até de estudar e criar um	Falta de conhecimento teórico sobre a ferramenta
Pra mim foi uma descoberta, porque eu vi o quanto pode ajudar pessoas, ate mesmo leigas como eu	Falta de conhecimento no método utilizado (blocos)
Planejado	Falta de conhecimento teórico sobre a ferramenta
	Falta de conhecimento no método utilizado (blocos)
Muito legal	Desconhecimento de algoritmo
Mais simples e interessante	Falta de conhecimento no método utilizado (blocos)
A facilidade	Falta de conhecimento teórico sobre a ferramenta
Muito legal	Falta de conhecimento teórico sobre a ferramenta
Mais simples e interessante	Falta de conhecimento no método utilizado (blocos)

Mais facil de fazer	Falta de conhecimento teórico sobre a ferramenta
Interessante	Falta de conhecimento no método utilizado (blocos)
É boa forma de aprender	Falta de conhecimento teórico sobre a ferramenta
Agora eu enxergo de outro modo melhor	Falta de conhecimento no método utilizado (blocos)
Agora tenho conhecimento	Falta de conhecimento teórico sobre a ferramenta
Melhor ainda	Falta de conhecimento teórico sobre a ferramenta
acho que e bem útil	Falta de conhecimento no método utilizado (blocos)
Tem muitos detalhes, mas é bastante interessante	Falta de conhecimento no método utilizado (blocos)
Ótima	Falta de conhecimento no método utilizado (blocos)
Mais simples	Falta de conhecimento no método utilizado (blocos)
da mesma forma tbm	Desconhecimento de algoritmo
Que precisa ser bem feito e com qualidade	Desconhecimento de algoritmo
Bem, ao final desta aula de programaçãoo incrivelmente cheia de sabedoria, vejo que é uma ótima matéria para se poder aprender sempre mais	Falta de conhecimento no método utilizado (blocos)

Algo interessante	Falta de conhecimento no método utilizado (blocos)
Complicada	Falta de conhecimento no método utilizado (blocos)
interessante e divertido	Desconhecimento de algoritmo
Eu vejo como algo que todos temos capacidade de fazer	Falta de conhecimento no método utilizado (blocos)
Ainda acho interessante	
Usando estas ferramentas apresentadas em sala, algo fácil e rápido	
Um pouco complicado	Falta de conhecimento teórico sobre a ferramenta
Melhor ainda e mais fácil	
interessante	
Um pouco complicada	Falta de conhecimento no método utilizado (blocos)

Questão 11	Questão 12	Questão 13	Questão 14
Não	Não	Talvez a implementação de jogos pode ajudar.	Exatas
Não	Sim	Essa oficina tem que ir ao contatos de outros alunos mostrando como é fácil e divertido programar	Exatas
Não	Não		Não tive experiência
Não	Não	A iniciativa de desafios, propondo aos usuários que criem seus aplicativos em prol de suas necessidades, e os melhores após serem criados serão divulgados.	Exatas

Não	Não	Eu já achei ela muito interessante. Não tenho idéia como fazer ela mais interessante ainda	Não tive experiência
Não	Sim	Criar desafios interagindo com os amigos	Exatas
	Sim	Usar/mostrar mais exemplos que se enquadrem no cotidiano do público alvo da oficina o objetivo de despertar o interesse dos mesmos será contemplado	Exatas
Não	Não	Pedir para os participantes levarem celulares para interagir com o programa criado	Não tive experiência
Não	Não	Não teho sugestões, para mim não faltou nada	Não tive experiência
sim	Sim	Programacao de aplicativos da moda	Exatas
			Exatas
Ainda não	Sim	Tá legal assim	Exatas
Não	Não	Não sei	Exatas
Não	Sim	Pra mim nao precisa mudar nada	Não tive experiência
Não	Sim		Não tive experiência
Não	Não	Não, necessita mudar nada	Humanas
Achei um pouco complicado no começo	Sim	Continuar do jeito que está	Não tive experiência
Não	Não		Não tive experiência
Não	Sim		Exatas
Não	Não	Não tenho sugestões	Não tive experiência
Não	Sim		Não tive experiência
Não	Não	Haver mais oficinas	Não tive experiência

Não	Não	Já está ótima.	Exatas
Não	Não	Mostrar sempre novidades	Não tive experiência
Não	Sim	A Internet não acabar	Não tive experiência
Não	Não	Não sei	Exatas
Não	Não	O uso de novas ferramentas	Exatas
Não	Não	Apenas continuar como está	Não tive experiência
Sim, havia alguns outros componentes na aula que me trouxeram dificuldades, por exemplo o idioma inglês.	Não		Não tive experiência
Não	Não	Não tenho sugestão	Exatas
Não	Não	Mais aplicativos mobile para programação	Não tive experiência
Não	Não	Uma programação mais complexa	Não tive experiência
Não	Não	A oficina já interessante o suficiente para chamar a atenção de muitos	Não tive experiência
Não	Não	Para mim está ótimo	Não tive experiência
Encaixar os blocos	Não	Mais aplicativos semelhantes	Não tive experiência
Não	Não	Não tenho sugestão	Não tive experiência
	Não	Uma rede de internet que funcione em todos os computadores	Não tive experiência
Não	Não	Para mim está ótimo	Não tive experiência
Sim	Não		Não tive experiência

Questão 15	Questão 16	Questão 17
Sim	Já ouvi falar, mas nunca tinha utilizado	Já ouvi falar, mas nunca tinha utilizado
Depois da oficina sim.	Sim e já usei	Já ouvi falar, mas nunca tinha utilizado
Sim, conseguiria desenvolver.	Já ouvi falar, mas nunca tinha utilizado	Já ouvi falar, mas nunca tinha utilizado
Sim	Já ouvi falar, mas nunca tinha utilizado	Já ouvi falar, mas nunca tinha utilizado
Conseguiria	Não conhecia	Não conhecia
Sim	Já ouvi falar, mas nunca tinha utilizado	Já ouvi falar, mas nunca tinha utilizado
Sim, a ferramenta é bem didática.	Não conhecia	Sim e já usei
Sim, a montagem do aplicativo em blocos creio eu que conseguiria montar	Não conhecia	Não conhecia
Acredito que sim	Não conhecia	Não conhecia
Não, nada	Já ouvi falar, mas nunca tinha utilizado	Não conhecia
Sim	Não conhecia	Não conhecia
Mais ou menos	Já ouvi falar, mas nunca tinha utilizado	Já ouvi falar, mas nunca tinha utilizado
Não , eu não entendi como pensar em uma sequência correta para os blocos	Não conhecia	Não conhecia
Que posso criar varios aplicativos	Não conhecia	Já ouvi falar, mas nunca tinha

		utilizado
Sim	Não conhecia	Não conhecia
Não	Já ouvi falar, mas nunca tinha utilizado	Já ouvi falar, mas nunca tinha utilizado
Conseguiria	Não conhecia	Não conhecia
Conseguiria	Não conhecia	Não conhecia
Sim	Sim e já usei	Já ouvi falar, mas nunca tinha utilizado
Talvez com algum tempo eu conseguiria desenvolver	Não conhecia	Não conhecia
Sim	Não conhecia	Não conhecia
Não	Já ouvi falar, mas nunca tinha utilizado	Já ouvi falar, mas nunca tinha utilizado
Conseguiria	Não conhecia	Não conhecia
Eu acho que eu conseguiria	Não conhecia	Não conhecia
Sim conseguiria	Já ouvi falar, mas nunca tinha utilizado	Já ouvi falar, mas nunca tinha utilizado
Não, não consegui juntar os blocos	Não conhecia	Não conhecia
Sim	Não conhecia	Não conhecia
Talvez sim. De primeiro momento compreendi sua função	Não conhecia	Não conhecia
Bem, poderia sim desenvolver, mas hoje não seria com tamanha facilidade até porque não temos um conhecimento tão grande sobre programação de aplicativos	Não conhecia	Não conhecia
Sim	Não conhecia	Não conhecia
Conseguiria	Não conhecia	Não conhecia
Eu conseguiria	Não conhecia	Não conhecia
Sim	Não conhecia	Não conhecia

Sim	Não conhecia	Já ouvi falar, mas nunca tinha utilizado
Sim	Não conhecia	Não conhecia
Sim	Não conhecia	Não conhecia
Só depois que eu tivesse essa aula	Não conhecia	Não conhecia
Conseguiria	Não conhecia	Não conhecia
Não , muito complexo	Não conhecia	Já ouvi falar, mas nunca tinha utilizado

Questão 18	Questão 19	Questão 20
Interesse por tecnologia utilizada como ferramenta na educação	Muito bom	MIT APP INVENTOR
Interesse pelo tema da oficina	Regular	MIT APP INVENTOR
Interesse pelo tema da oficina	Regular	MIT APP INVENTOR
Interesse pelo tema da oficina	Regular	MIT APP INVENTOR
Interesse pelo tema da oficina	Fraco	MIT APP INVENTOR
Interesse na questão do uso do dispositivo móvel como forma de aprendizado	Regular	MIT APP INVENTOR
Outros	Regular	MIT APP INVENTOR
Interesse por tecnologia utilizada como ferramenta na educação	Muito bom	MIT APP INVENTOR
Interesse na questão do uso do dispositivo móvel como forma de aprendizado	Excelente	Scratch
Interesse pelo tema da oficina	Muito bom	Scratch

Interesse pelo tema da oficina	Muito bom	Scratch
Interesse pelo tema da oficina	Regular	Scratch
Outros	Regular	Scratch
Interesse na questão do uso do dispositivo móvel como forma de aprendizado	Muito bom	MIT APP INVENTOR
Interesse pelo tema da oficina	Regular	Scratch
Interesse por tecnologia utilizada como ferramenta na educação	Excelente	Scratch
Interesse na questão do uso do dispositivo móvel como forma de aprendizado	Muito bom	MIT APP INVENTOR
Outros	Regular	Scratch
Interesse na questão do uso do dispositivo móvel como forma de aprendizado	Excelente	MIT APP INVENTOR
Interesse pelo tema da oficina	Péssimo	Scratch
Interesse na questão do uso do dispositivo móvel como forma de aprendizado	Muito bom	MIT APP INVENTOR
Interesse na questão do uso do dispositivo móvel como forma de aprendizado	Regular	MIT APP INVENTOR
Interesse por tecnologia utilizada como ferramenta na educação	Muito bom	MIT APP INVENTOR
Interesse na questão do uso do dispositivo móvel como forma de aprendizado	Muito bom	MIT APP INVENTOR
Interesse por tecnologia utilizada como ferramenta na educação	Muito bom	Scratch
Outros	Regular	Scratch
Interesse pelo tema da oficina	Muito bom	Scratch
Interesse por tecnologia utilizada como ferramenta na educação	Regular	MIT APP INVENTOR
Outros	Excelente	Scratch
Interesse pelo tema da oficina	Muito bom	MIT APP INVENTOR
Interesse pelo tema da oficina	Regular	Scratch

Interesse por tecnologia utilizada como ferramenta na educação	Regular	Scratch
Interesse pelo tema da oficina	Excelente	MIT APP INVENTOR
Interesse pelo tema da oficina	Excelente	Scratch
Interesse pelo tema da oficina	Excelente	MIT APP INVENTOR
Interesse por tecnologia utilizada como ferramenta na educação	Muito bom	Scratch
Interesse pelo tema da oficina	Excelente	Scratch
Interesse pelo tema da oficina	Muito bom	Scratch
Interesse pelo tema da oficina	Regular	Scratch

Questão 21	Questão 22	Questão 23
A possibilidade de desenvolver um aplicativo tornou a App Inventor mais interessante.	As duas, pois são bastante similares	MIT App Inventor
O MIT APP INVENTOR é mais dinâmico e abrangente.	MIT App Inventor	MIT App Inventor
Depois do desenvolvimento de sua aplicação ele te fornece um aplicativo que pode ser baixado e instalado em seu smartphone Android.	MIT App Inventor	MIT App Inventor
Me identifiquei mais ao MIT APP INVENTOR, ambos são excelentes.	MIT App Inventor	MIT App Inventor
Creio que o MIT APP INVENTOR é mais interessante. Creio q ele tem uma maior facilidade.	As duas, pois são bastante similares	MIT App Inventor
O melhor manuseio e a abrangência de manipulação	Scratch	Scratch
Ao meu ver o MIT APP INVENTOR é uma ferramenta mais completa.	As duas, pois são bastante similares	MIT App Inventor
Apresenta o aplicativo com uma forma que já to acostumada no meu celular	As duas, pois são bastante	MIT App Inventor

	similares	
A questão gráfica com desenhos, acredito tornar mais fácil o aprendizado para crianças ou a uma pessoa leiga como eu	Scratch	Scratch
Por que ele é mais legal	Scratch	Scratch
	Scratch	MIT App Inventor
Ele é mais fácil	Scratch	Scratch
É mais simples	As duas, pois são bastante similares	Scratch
Uma melhor interface grafica	MIT App Inventor	Scratch
Mais	Scratch	Scratch
Preferi, pois é mais interessante	As duas, pois são bastante similares	Scratch
Eu achei melhor por que pode usar em dispositivos moveis	As duas, pois são bastante similares	MIT App Inventor
Gostei pois é mais interativo, devido a cores, etc.	MIT App Inventor	Scratch
sim	As duas, pois são bastante similares	MIT App Inventor
	MIT App Inventor	MIT App Inventor
É uma ferramenta mais esclarecida	MIT App Inventor	MIT App Inventor
O MIT é muito bom	MIT App Inventor	MIT App Inventor
Porque ela tem mais opções de uso	As duas, pois são bastante similares	Scratch
Parece ser mais fácil de ser manuseada	MIT App Inventor	MIT App Inventor

Eu gostei mais do Scratch	Scratch	Scratch
Ele é mais simples	Scratch	Scratch
A facilidade do programa e ser um programa desktop	As duas, pois são bastante similares	Scratch
Sim o MIT APP tem oportunidade para teste do aplicativo feito	As duas, pois são bastante similares	Scratch
Uma interface de tamanha facilidade maior mostrou ser também superior a sua concorrente e também adicionais como o som que o gato emite pode chamar a atenção de usuários diversos	Scratch	Scratch
Funcionalidade e praticidade	As duas, pois são bastante similares	MIT App Inventor
Interface gráfica (gatinho)	As duas, pois são bastante similares	Scratch
É mais divertido, sim a interface	Scratch	Scratch
Nele é necessária maior atenção e raciocínio lógico e possui uma interface mais 'real'	MIT App Inventor	Scratch
Mostra mais divertimento.	As duas, pois são bastante similares	Scratch
Capacidade de produzir o App	Scratch	MIT App Inventor
Não sei, apenas me chamamos mais a atenção	As duas, pois são bastante similares	Nenhuma
Por causa do gatinho	As duas, pois são bastante similares	Scratch
O gatinho	As duas, pois são bastante similares	Nenhuma
Simplicidade	Scratch	Scratch

Questão 24	Questão 25	Questão 26	Questão 27
Sim. Talvez nem tanto, mas depende do que for desenvolver	Scratch	if ((x > 3) and (x <20))	4
Sim.	Scratch	if ((x > 3) and (x <20))	5
Sim, na minha opinião o App inventor se sobressai por isso, além de poder programar em blocos ele ainda fornece um aplicativo que pode ser instalado em um smartphone.	MIT App Inventor	if ((x > 3) and (x <20))	5
Sim. Sim.	MIT App Inventor	if ((x > 3) and (x <20))	5
Sim. Tem mais interação com os dispositivos móveis.	Scratch	if ((x > 3) and (x <20))	5
Sim	MIT App Inventor	if ((x > 3) and (x <20))	5
Com certeza, fazer uso de uma ferramenta que como resultado produz quase que automaticamente um aplicativo móvel é um grande diferencial	MIT App Inventor	if ((x > 3) and (x <20))	5
Sim, porque se torna mais acessível	MIT App Inventor	if ((x < 3) and (x >20))	5
Não vejo uma diferença enorme	Scratch		5
	Scratch	if ((x > 3) and (x <20))	5
	Scratch	if ((x > 3) or (x <20))	5
Sim	Scratch	if ((x > 3) and (x <20))	5
Não	Scratch	if ((x > 3) and (x <20))	4
Sim, grande	MIT App Inventor	if ((x > 3) and (x <20))	5

Sim	Scratch	if ((x > 3) and (x <20))	5
Sim	Scratch	if ((x > 3) and (x <20))	5
sim	MIT App Inventor	if ((x > 3) and (x <20))	5
Não tem muita relevância	Scratch	if ((x > 3) and (x <20))	5
	MIT App Inventor	if ((x < 3) and (x >20))	5
Sim	MIT App Inventor	if ((x > 3) and (x <20))	5
Sim	MIT App Inventor	if ((x < 3) and (x >20))	5
Sim	MIT App Inventor	if ((x > 3) and (x <20))	5
Não sei	Scratch	if ((x > 3) and (x <20))	5
Sim	Scratch	if ((x > 3) and (x <20))	5
	Scratch	if ((x > 3) and (x <20))	5
Sim	Scratch	if ((x > 3) and (x <20))	5
	Scratch	if ((x < 3) and (x >20))	5
Não. Releva um pouco pela característica de teste	Scratch	if ((x > 3) and (x <20))	5
Seu maior diferencial, mostrado na aula foi simplesmente sua facilidade para ser baixado e instalado para utilização no android	Scratch	if ((x > 3) and (x <20))	5
Acredito que sim	MIT App Inventor	if ((x > 3) and (x <20))	5
Sim. Sim	Scratch	if ((x > 3) and (x <20))	5

		(x <20))	
Não pois eles tem a mesma lógica de programação que serviria para os dois	Scratch	if ((x < 3) and (x >20))	5
Sim	MIT App Inventor	if ((x > 3) and (x <20))	5
Tem uma grande importância	Scratch	if ((x > 3) and (x <20))	5
Sim. Sim	Scratch	if ((x > 3) and (x <20))	5
Sim, pois abre espaço para criações maiores	Scratch	if ((x > 3) and (x <20))	5
Sim	Scratch	if ((x > 3) and (x <20))	5
Tem sim uma grande importancia	Scratch	if ((x > 3) and (x <20))	5
Sim. Sim	Scratch	if ((x > 3) and (x <20))	5

Importante: As questões 1 e 2 foram omitidas, pois revelam a identidade dos participantes.