



UNIVERSIDADE FEDERAL RURAL DO SEMI-ÁRIDO
PRÓ-REITORIA DE GRADUAÇÃO
DEPARTAMENTO DE CIÊNCIAS EXATAS, TECNOLÓGICAS E HUMANAS
CURSO DE LICENCIATURA EM COMPUTAÇÃO E INFORMÁTICA

MARIA LUIZA SOARES DA SILVA BISNETA

**APRENDER LINGUAGEM DE PROGRAMAÇÃO NO ENSINO FUNDAMENTAL COM A
FERRAMENTA *SCRATCH*: SENTIDOS CONSTRUÍDOS POR ESTUDANTES**

ANGICOS - RN

2019

MARIA LUIZA SOARES DA SILVA BISNETA

**APRENDER LINGUAGEM DE PROGRAMAÇÃO NO ENSINO FUNDAMENTAL COM A
FERRAMENTA SCRATCH: SENTIDOS CONSTRUÍDOS POR ESTUDANTES**

Monografia apresentada a Universidade Federal Rural do Semi-Árido como requisito para obtenção do título de Licenciatura plena em Computação e Informática.

Orientador: Prof. Drº: Samuel Oliveira de Azevedo

Co-orientador: Prof. Drª. Elaine Luciana Sobral Dantas

ANGICOS – RN

2019

© Todos os direitos estão reservados a Universidade Federal Rural do Semi-Árido. O conteúdo desta obra é de inteira responsabilidade do (a) autor (a), sendo o mesmo, passível de sanções administrativas ou penais, caso sejam infringidas as leis que regulamentam a Propriedade Intelectual, respectivamente, Patentes: Lei nº 9.279/1996 e Direitos Autorais: Lei nº 9.610/1998. O conteúdo desta obra tomar-se-á de domínio público após a data de defesa e homologação da sua respectiva ata. A mesma poderá servir de base literária para novas pesquisas, desde que a obra e seu (a) respectivo (a) autor (a) sejam devidamente citados e mencionados os seus créditos bibliográficos.

B621a BISNETA, MARIA LUIZA SOARES DA SILVA.
APRENDER LINGUAGEM DE PROGRAMAÇÃO NO ENSINO
FUNDAMENTAL COM A FERRAMENTA SCRATCH: SENTIDOS
CONSTRUÍDOS POR ESTUDANTES / MARIA
LUIZASOARES DA SILVA BISNETA. - 2019.
89 f. : il.

Orientador: Samuel Oliveira de Azevedo.
Coorientadora: Elaine Luciana Sobral Dantas.
Monografia (graduação) - Universidade Federal
Rural do Semi-árido, Curso de
Computação e Informática, 2019.

1. Sentidos das crianças. 2. Lógica de programação. 3. Scratch. I. Azevedo, Samuel Oliveira de, orient. II. Dantas, Elaine Luciana Sobral, co-orient. III. Título.

O serviço de Geração Automática de Ficha Catalográfica para Trabalhos de Conclusão de Curso (TCC's) foi desenvolvido pelo Instituto de Ciências Matemáticas e de Computação da Universidade de São Paulo (USP) e gentilmente cedido para o Sistema de Bibliotecas da Universidade Federal Rural do Semi-Árido (SISBI-UFERSA), sendo customizado pela Superintendência de Tecnologia da Informação e Comunicação (SUTIC) sob orientação dos bibliotecários da instituição para ser adaptado às necessidades dos alunos dos Cursos de Graduação e Programas de Pós-Graduação da Universidade.

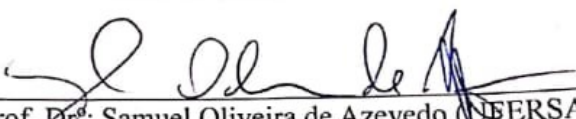
MARIA LUIZA SOARES DA SILVA BISNETA

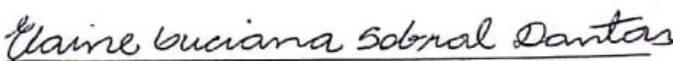
**APRENDER LINGUAGEM DE PROGRAMAÇÃO NO ENSINO FUNDAMENTAL COM A
FERRAMENTA *SCRATCH*: SENTIDOS CONSTRUÍDOS POR ESTUDANTES**

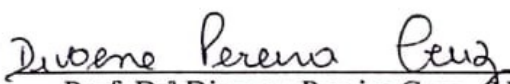
Monografia apresentada a Universidade Federal Rural do Semi-
Árido como requisito para obtenção do título de Licenciatura
plena em Computação e Informática.

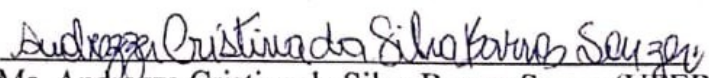
Defendida em: 16 / 08 / 2019.

BANCA EXAMINADORA


Prof. Dr^o: Samuel Oliveira de Azevedo (UFERSA)
Presidente


Prof. Dr^a. Elaine Luciana Sobral Dantas (UFERSA)
Vice-Presidente


Prof. Dr^a Divoene Pereira Cruz. (UFERSA)
Membro Examinador


Prof. Me. Andrezza Cristina da Silva Barros Souza (UFERSA)
Membro Examinador

Pretão, Scooby, Cristina e Totty (In Memoriam).

Naya e Hebie (presentes)

AGRADECIMENTOS

Agradeço primeiro a minha mãe Maria Soares e a minha vizinha Maria das Dores por ter cuidado de mim por todos esses anos, eu amo vocês.

Agradeço a minha irmã Alice por tudo que fez por mim até hoje, pelos incentivos, pelas brigas por me fazer refletir, por me apoiar, também quero agradecer a meu irmão Elias por me amar. Agradeço aos meus familiares em especial, minha tia mara, minha prima kathycia e Zé de lima pelo carinho e apoio.

Agradeço a minha prima Maria e meu primo Felipe, por me fazerem rir e por ser meus primos favoritos.

Quero agora agradecer as pessoas que me acompanharam de perto no mundo sofrido da Ufersa, meus amigos, os quatro primeiros nomes que vou citar quero deixar meu muito obrigado, se estou em pé e ainda aqui é porque tive vocês ao meu lado, me dando forças, apoiando minhas ideias criando junto comigo, chorando, rindo... E não tenho dúvidas do amor de vocês por mim, aqui deixo meu abraço e carinho aos meus migos, Paulinha, Erick, kátia e Alberto.

Agradeço e fico muito feliz de ter tido a honra de conhecer pessoas como Júlia meu amor de c&t, meus amores da LCI Joel, Dorinha, Rek, Daiane, Carla, Maria de Fátima, Jefferson, Thiago, Sinésio, Plínio, Maria Jales, Arthur, Alan e tantos outros. Meus colegas e amigos queridos de BSI Aldeci, Juan, Antônia, Tereza, Axel, Érica. Meus nenéns de pedagogia Joyce, Gabi, Paulo, Daniela, Welly, Alice, Michele, Laiara, em especial Zamy que começou comigo em LCI, e também agradeço de todo coração o carinho os abraços e os cheiros do meu anjo Serafim.

Agradeço as minhas amigas de infância Karol, Frankyllene, Andreza e Jardilla. Com toda certeza meu caráter e referência do que é ser amiga de verdade aprendi com vocês.

Quero deixar meu muito obrigada aos meus professores da graduação em especial, Maria do socorro, José Gildo, Sairo, Alex, Samuel e Elaine meus orientadores, Enai e Magnus. obrigada pelo conhecimento e ensinamentos.

E por último quero agradecer a mim, do passado, do agora e já me agradeço no futuro, obrigada por ter sido forte e parabéns.

Todo dia eu penso em desistir, e acabo
por desistir dessa ideia de desistir.

RESUMO

A computação está impulsionando a criação de empregos e a inovação em toda a nossa economia e sociedade. “Hoje, para serem cidadãos bem educados num mundo permeado de Tecnologia da Informação, todos devem possuir uma compreensão clara dos princípios e práticas da computação além do simples uso da TI” von Wangenheim, Nunes e Santos (2014, p. 116). Esse trabalho tem como objetivo, analisar sentidos construídos por alunos dos anos iniciais do ensino fundamental sobre suas aprendizagens em linguagem de programação com o uso da ferramenta *Scratch* no processo educacional. Para isso colocou-se como cenário, o projeto de extensão “*Scratch: Dê vida a sua imaginação*”, desenvolvido por alunos dos cursos de Computação, Campus da UFERSA em Angicos/RN, teve como perspectiva de proporcionar para alunos do ensino fundamental dos 4º e 5º anos do CEPA - Centro Educacional Doutor Pedro Amorim, o ensino dos conceitos básicos de computação relacionada ao ensino de lógica da programação, para alunos que nunca tiveram contato com o ensino de programação. O ensinar da lógica de programação transfere para a criança um pensamento mais organizado, impulsiona a sua criatividade e é o primeiro passo para aprender uma linguagem de programação, oferece a criança novas possibilidades de aprendizagem, pois a escola deve ensinar tudo que for real, atual dentro do contexto dos alunos e isto inclui o ensino da ciência da computação. A proposta metodológica da pesquisa-ação foi utilizado questionários, para obter o resultados quantitativos e qualitativos. Com a tabulação dos questionários, os gráficos evidenciam que o ensino da ciência da computação, dentro de programação de computadores com alunos do ensino fundamental pelas percepção e sentidos deles, deveria ser integralizado no currículo escolar, pois além da afetividade e o interesse que tiveram com o conteúdo e a ferramenta, mostram que eles acham importante ser estudado. Tendo em vista que essas tecnologias fazem parte do cotidiano dos alunos, dentro do contexto onde estão inseridos, a escola teria uma melhora em questão do desempenho dos alunos nas outras disciplinas, segundo as perspectivas dos alunos. Os sentidos construídos desses estudantes é uma amostra que no contexto urbano nordestino do interior as crianças já demonstram interesse em estudar lógica da programação. Crer-se que privar o conhecimento em assuntos de interesse dos estudantes é tirar as oportunidades de futuros profissionais desse campo.

Palavras-chave: Sentidos das crianças; Lógica de programação; *Scratch*.

ABSTRACT

Computing is driving job creation and innovation throughout our economy and society. “Today, to be well-educated citizens in an Information Technology-permeated world, everyone must have a clear understanding of computing principles and practices beyond the simple use of IT” von Wangenheim, Nunes & Santos (2014, p. 116). This paper aims to analyze meanings built by students of the early years of elementary school about their learning in programming language using the Scratch tool in the educational process. To this end, the “Scratch: Bring Your Imagination” extension project, developed by students from the Computer courses at the UFERSA Campus in Angicos / RN, had as its perspective the objective of providing for elementary students in the 4th grade. and 5th year of CEPA - Doctor Pedro Amorim Educational Center, the teaching of basic computing concepts related to the teaching of programming logic, for students who have never had contact with the teaching of programming. Teaching programming logic transfers a more organized thinking to the child, boosts creativity and is the first step to learning a programming language, offers the child new learning possibilities, because the school must teach everything that is real, current. within the context of the students and this includes the teaching of computer science. The methodological proposal of the field research was used questionnaires, to obtain the quantitative and qualitative results. With the tabulation of the questionnaires, the graphs show that the teaching of computer science, within the programming of computers with elementary school students by their perception and meanings, should be integrated into the school curriculum, because besides the affection and interest they had with the content and the tool show that they find it important to be studied. Given that these technologies are part of students 'daily life, within the context in which they are inserted, the school would have an improvement in student performance in other subjects, according to the students' perspectives. The constructed senses of these students is a sample that in the northeastern urban context of the interior children already show interest in studying the logic of programming. It is believed that depriving students of knowledge on subjects of interest is taking opportunities from future professionals in this field.

Keywords: Children's senses; Programming logic; Scratch.

LISTA DE FIGURAS

Figura 1	– Fluxograma do que é um <i>algoritmo</i>	34
Figura 2	– Computação Interdisciplinar.....	39
Figura 3	– Ambiente de programação <i>scratch</i> 2.0.....	41
Figura 4	– Desenho do aluno X.....	51
Figura 5	– Desenho do aluno Y.....	51
Figura 6	– Desenho do aluno Z.....	52
Figura 7	– Desenho do aluno XY.....	52
Figura 8	– Desenho do aluno XYZ.....	52

LISTA DE GRÁFICOS

Gráfico 1	– Percentual de alunos por Idade.	21
Gráfico 2	– Percentual de alunos por ano escolar.....	21
Gráfico 3	– Percentual de alunos por gênero.....	22
Gráfico 4	– Percentual de alunos por etnia.....	22
Gráfico 5	– Percentual de alunos por gênero e moradia.....	23
Gráfico 6	– Percentual de alunos por gênero inscritos.....	25
Gráfico 7	– Evasão do curso por Gênero.....	27
Gráfico 8	– Questão 1.....	47
Gráfico 9	– Questão 1-A: Resposta "Não".....	48
Gráfico 10	– Questão 1-A: Resposta "Sim".....	48
Gráfico 11	– Questão 1-B.....	49
Gráfico 12	– Questão 2.....	50
Gráfico 13	– Questão 4.....	53
Gráfico 14	– Questão 5.....	54
Gráfico 15	– Questão 6.....	54
Gráfico 16	– Questão 7.....	55
Gráfico 17	– Questão 7-A.....	56
Gráfico 18	– Questão 8.....	57
Gráfico 19	– Questão 9.....	58
Gráfico 20	– Questão 10.....	59
Gráfico 21	– Questão 11.....	59
Gráfico 22	– Questão 12.....	60
Gráfico 23	– Questão 12-A.....	61

LISTA DE QUADROS

Quadro 1	–	Imagens Primeiro dia de curso.....	28
Quadro 2	–	Imagens apresentação dos projetos.....	29

LISTA DE TABELAS

Tabela 1	–	Classificação; Conceito; e Paradigmas de linguagens.....	34
Tabela 2	–	Classificação de uso de ferramentas	38

LISTA DE ABREVIATURAS E SIGLAS

BNCC – Base Nacional Comum Curricular

BSI – Bacharelado Em Sistemas de Informação

CEPA - Centro Educacional Doutor Pedro Amorim

IFRN – Instituto Federal do Rio Grande do Norte

LCI – Licenciatura Em Computação e Informática

SBC - Sociedade Brasileira de Computação

SIGAA - Sistema Integrado de Gestão de Atividades Acadêmicas

TDIC - Tecnologias Digitais de Informação e Comunicação

LISTA DE SÍMBOLOS

@	Arroba
©	Copyright
®	Marca registrada
%	Porcentagem
\$	Cifrão

SUMÁRIO

1 INTRODUÇÃO.....	17
2 CARACTERIZAÇÃO DO UNIVERSO DA PESQUISA.....	20
2.1 RELATO DE EXPERIÊNCIA – O PROJETO NO CONTEXTO DO MINISTRANTE.....	23
2.2 PROJETO DE EXTENSÃO – <i>SCRATCH</i> : DÊ VIDA A SUA IMAGINAÇÃO.....	30
3 LÓGICA E LINGUAGEM DE PROGRAMAÇÃO.....	32
3.1 FERRAMENTA DE PROGRAMAÇÃO COMO MÉTODOS DE ENSINO.....	37
3.2 <i>SCRATCH</i>	40
3.3 COMPUTAÇÃO NO ENSINO FUNDAMENTAL.....	42
4 METODOLOGIA DA PESQUISA.....	46
5 RESULTADOS E DISCUSSÕES.....	47
6 CONSIDERAÇÕES FINAIS.....	62
6.1 TRABALHOS FUTUROS.....	63
REFERÊNCIAS.....	64
APÊNDICE A.....	67
ANEXO A.....	70
APÊNDICE B.....	74

1 INTRODUÇÃO

Entender os novos rumos que a sociedade está seguindo em tempos de globalização onde a tecnologia da informação e comunicação representa o principal eixo de mudança social e econômica dessa nova civilização, é também compreender que tudo muda com o tempo, e é preciso seguir com essa mudança, compreendendo suas causas, fenômenos que muitos chamam de evolução.

Nessa nova realidade,

Vivemos em tempos confusos, como muitas vezes é o caso em períodos de transição entre diferentes formas de sociedade. Isso acontece porque as categorias intelectuais que usamos para compreender o que acontece à nossa volta foram cunhadas em circunstâncias diferentes e dificilmente podem dar conta do que é novo referindo-se ao passado. (CASTELLS, 2010, P. 1).

Neste contexto e, considerando que as tecnologias educacionais ocupam o trabalho dos pesquisadores em como introduzir e da concretude dessas ferramentas na vida dos sujeitos, surge a importância de inclusão do ensino da computação no currículo da educação básica. Esse processo de inclusão acontece pelo fato de que precisamos educar nossos alunos e cidadãos para saber viver em uma sociedade digital em termos de trabalho, estudo e lazer. Porém não somente usar a tecnologia digital passivamente, mas também como criadores e multiplicadores dessas técnicas para um objetivo que beneficie sua própria formação e a sociedade como um todo.

Ao longo de estudos acadêmicos dentro da instituição superior, no curso de Licenciatura em Computação e Informática, entre vários temas estudados sobre educação, tecnologias e como conectar essas duas esferas através das tecnologias de informação e comunicação (TICs), várias dúvidas e interesses foram surgindo, e todos os estudos e disciplinas vistas durante o curso foram degraus para a formulação do conhecimento que molda este trabalho.

A primeira influência concreta com o tema deste trabalho foi através do conhecimento sobre algoritmos e programação de computadores, que trouxe uma visão crítica e questionamentos sobre porquê a computação não é ensinada na educação básica brasileira, já que as tecnologias digitais fazem parte das nossas vidas, temos cursos de nível superior nas instituições de ensino e espaço de trabalho no mercado brasileiro. Tomamos como exemplo, outros países como é o caso da “Austrália, que têm motivado as escolas a aumentarem o enfoque das aulas em áreas de ciência, tecnologia, engenharia e matemática desde o nível básico” (ROCHA, 2019). Fazendo também com que a programação de computadores entre no currículo como disciplina obrigatória para crianças acima de 10 anos.

Portanto, os estudos e iniciativas sobre as quais temos conhecimento fora e dentro da universidade fizeram com que surgisse a problemática que vai ser abordada neste projeto de pesquisa.

Nacional e internacionalmente já foram realizados estudos sobre a importância e os benefícios do ensino da computação e principalmente o ensino da lógica de programação para crianças. A partir desses estudos e de nossas preocupações, temos nos questionado sobre: O que pensam as crianças do ensino fundamental sobre aprender lógica de programação? Como esse aprendizado pode contribuir na educação dessas crianças? Diante disso, definimos como questão de pesquisa: **“Quais sentidos os estudantes da educação básica constroem acerca de suas aprendizagens sobre lógica da programação de computadores no contexto de utilização da ferramenta *Scratch*? ”**. Como a Licenciatura em Computação e Informática é uma área do conhecimento que envolve duas outras grandes áreas do conhecimento e áreas de atuação que são a pedagógica e a tecnológica, surge a necessidade da construção de trabalhos que proponham-se a realizar estudos que envolvam essas mesmas áreas do conhecimento, já que nossa sociedade está hoje em um contexto onde as TICs são parte do nosso cotidiano e que precisamos que elas sejam também parte do dia a dia das escolas, não somente como ferramentas de auxílios pessoais ou administrativas, mas que sejam usadas em um contexto educacional e com a intenção de usá-las como objeto de estudo em sala de aula. A partir desses estudos é salutar que possamos propor algumas metodologias e métodos de como incluir as TICs como parte do campo da educação, para que possamos criar em nossas crianças um pensamento computacional.

“O termo Pensamento Computacional, por sua vez, surge pela primeira vez com Wing (2006), ao afirmar que ele se constrói nos poderes e nos limites dos processos computacionais, sejam eles executados por um ser humano ou por uma máquina” Ramos, Teixeira (2016). Embora aparentemente possamos pensar como um termo mecanizado, o pensamento computacional na realidade, refere-se a métodos que através do computador estimulam, o pensamento lógico assim como a criatividade e a organização do pensamento de forma individual ou compartilhada, entre outros fatores e benefícios (WING, 2006)

O surgimento dessa proposta deu-se através da observação do contexto atual da nossa sociedade e os debates que ocorrem sobre as reformulações curriculares do ensino básico. Ramos e Teixeira (2016) refletem sobre o uso da computação e informática no ensino básico. Segundo eles:

A difusão das Tecnologias Digitais de Informação e Comunicação (TDIC) nas atividades humanas nessas últimas décadas, tem provocado discussões relacionadas à inserção dessas tecnologias na educação desde as etapas iniciais da educação básica. É possível registrar em muitas escolas a tímida inserção do ensino de Informática como componente curricular. Geralmente o trabalho desenvolvido na disciplina de Informática apresenta uma perspectiva meramente instrucionista, de forma que os estudantes são conduzidos a uma postura

passiva diante do computador, manipulando a máquina superficialmente, aprendendo as instruções básicas de editores de texto e/ou planilhas eletrônicas.

Pelas razões e reflexões expostas, esse trabalho se torna importante, considerando as tecnologias algo tão presente e natural para os jovens dessa década, considerando, também, as mudanças curriculares que estão ocorrendo em toda a educação básica brasileira. Sendo assim, é importante que tenhamos em nosso currículo básico a computação como ciência a ser estudada por crianças, adolescentes e jovens.

O projeto de extensão “*Scratch*: Dê vida a sua imaginação”, foi desenvolvido por licenciandos do curso de Licenciatura em Computação e Informática do Campus da UFERSA em Angicos/RN, teve como perspectiva proporcionar para alunos do ensino fundamental dos 4º ano o ensino dos conceitos básicos de computação relacionada ao ensino de lógica da programação. A utilização da ferramenta *scratch* se justifica, pois ela foi desenvolvida justamente com foco de ensino de lógica de programação para crianças na idade escolar, a ferramenta em si, é totalmente educativa e lúdica, proporcionando ensino mais fácil e mais interativo dos componentes básicos da computação, dentro do campo de ensino da programação.

Portanto este trabalho tem como objetivos analisar sentidos construídos por alunos dos anos iniciais do ensino fundamental sobre suas aprendizagens em linguagem de programação com o uso da ferramenta *Scratch* no processo educacional.

Onde iremos identificar conceitos e práticas adquiridos pelos alunos sobre a linguagem de programação a partir da utilização da ferramenta *Scratch* e analisar como esses aprendizados contribuem para a educação dos estudantes segundo suas perspectivas.

A metodologia de pesquisa utilizada será a pesquisa ação que ocorre um constante vaivém entre as fases, determinada pela dinâmica do grupo de pesquisadores em seu relacionamento com a situação pesquisada (GIL, 2002).

A primeira etapa da pesquisa foi constituída de revisão bibliográfica, com base em autores e pesquisadores da área da computação e educação, através de livros, apostilas e artigos retirados do banco de dados do *google* acadêmico e da biblioteca da UFERSA. Posteriormente entramos em campo para fazer a análise do perfil dos alunos, a partir da pesquisa de campo onde foi acompanhando as aulas do projeto, analisando o conteúdo e o método de ensino. Ao final foi aplicado um questionário com uma linguagem apropriada ao público alvo que são os alunos que participaram do projeto e tendo por base os objetivos específicos definidos cujo eixo central é identificar os sentidos das crianças sobre o ensino de lógica de programação.

A primeira etapa do trabalho trata da caracterização do nosso universo da pesquisa, onde foi identificado que o perfil do nosso universo são: meninos, pardos e negros com 9 anos de idade,

cursando o 4º ano do fundamental. Seguindo os demais capítulos onde se trata de toda a contextualização do tema, fala sobre o projeto de extensão como foi constituído através do seu documento de construção, e também foi feito sua descrição em forma de relato de experiência do executor do projeto, criando um olhar empírico do projeto. Depois foi feita uma documentação da ferramenta que está nos anexos, que foi utilizada para ensino da lógica da programação, mostrando toda sua interface e funcionalidades.

Teve-se uma reflexão sobre a computação no ensino fundamental, mostrando o contexto da computação no Brasil e refletindo o porque é viável que a computação seja uma área que precisa estar no sistema educacional desde a base de ensino, com enfoque no ensino fundamental tendo como exemplo de conteúdos a sugestão da SBC para a BNCC feita em 2016, e os sentidos da BNCC sobre a temática, onde foi feito um recorte do que diz em seus documentos na questão de linguagens e tecnologias.

Foi construído um capítulo teórico histórico sobre a lógica e linguagem de programação, para deixar os leitores que não são da área inteirados um pouco sobre a lógica da programação e suas linguagens como ciência.

No capítulo sobre ferramentas como métodos de ensino, fizemos uma contextualização histórica das ferramentas de programação mostramos algumas ferramentas divididas por escolaridade e onde tentou-se mostrar que a computação trata-se de uma área interdisciplinar que pode ser utilizada em várias áreas do conhecimento.

2 CARACTERIZAÇÃO DO UNIVERSO DA PESQUISA

A caracterização do perfil dos alunos torna-se importante para que a pesquisa possa ser toda situada no estudo das características desse grupo, levando em consideração todo o contexto social da amostragem do nosso universo de estudo, para que através dos resultados obtidos da pesquisa possamos criar hipóteses sobre esse contexto e assim fazer reflexões tendo uma observação desse grupo como exemplo, para que futuramente outras pesquisas possam fazer comparações desse contexto com outros e assim possibilitar a criação de uma teoria. Para Gil (2002, p. 53) a caracterização se torna importante pois, "no levantamento procura-se identificar as características dos componentes do universo pesquisado, possibilitando a caracterização precisa de seus segmentos."

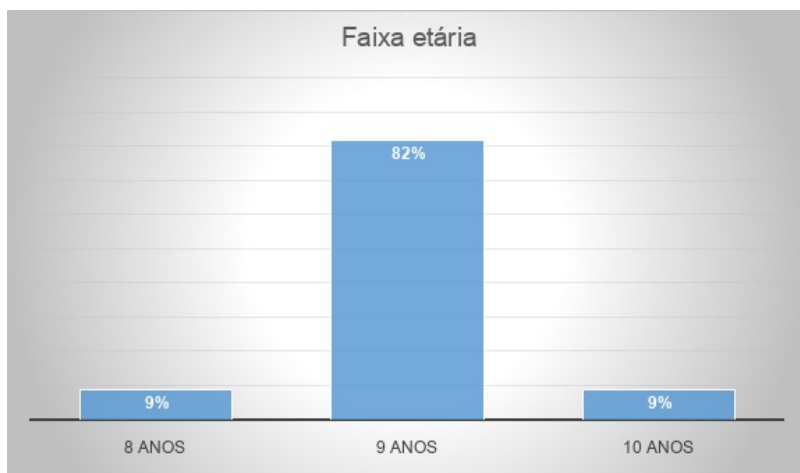
"A especificação dos objetivos é feita pela identificação de todos os dados a serem recolhidos e das hipóteses a serem testadas." Gil (2002, p. 112). O objetivo é traçar o perfil socioeconômico do universo da pesquisa.

Por tanto temos como objetivos específicos da caracterização:

1. Idade;
2. Ano escolar;
3. Etnia;
4. Gênero;
5. Moradia.

Os objetivos específicos tentam descrever, nos termos mais claros possíveis, exatamente o que será obtido num levantamento. Gil (2002, p. 112).

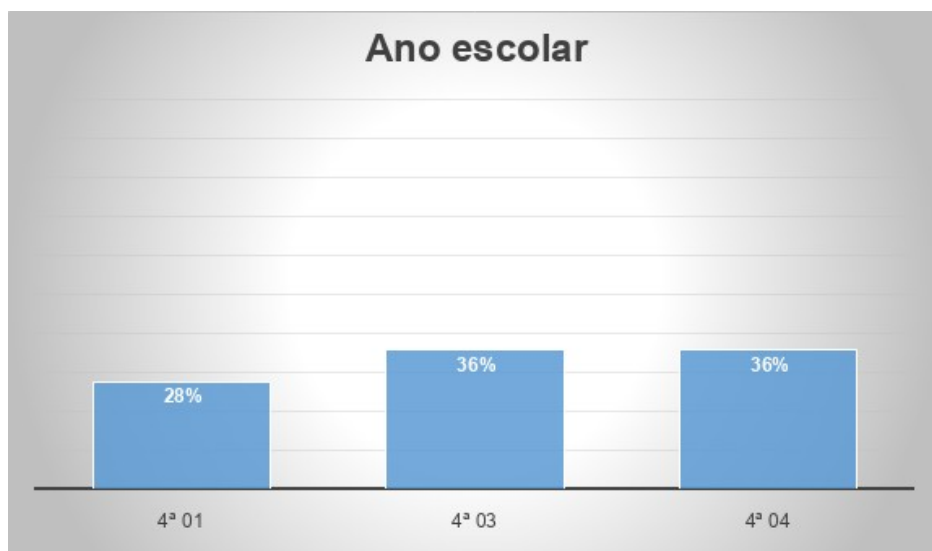
Gráfico 1 – Percentual de alunos por Idade



Fonte: da autora

O gráfico 1 mostra que os alunos têm idades regular entre 8 à 10 anos. Apropriados para os anos escolares que frequentam, sendo majoritário crianças com 9 anos. Isso mostra que os alunos da escola CEPA que participaram do projeto *Scratch* estão na idade apropriada e regular de ensino fundamental I.

Gráfico 2 – Percentual de alunos por ano escolar



Fonte: da autora

Conforme o gráfico 2, o universo da pesquisa obteve-se que 100% dos alunos cursam o 4º ano do ensino fundamental I.

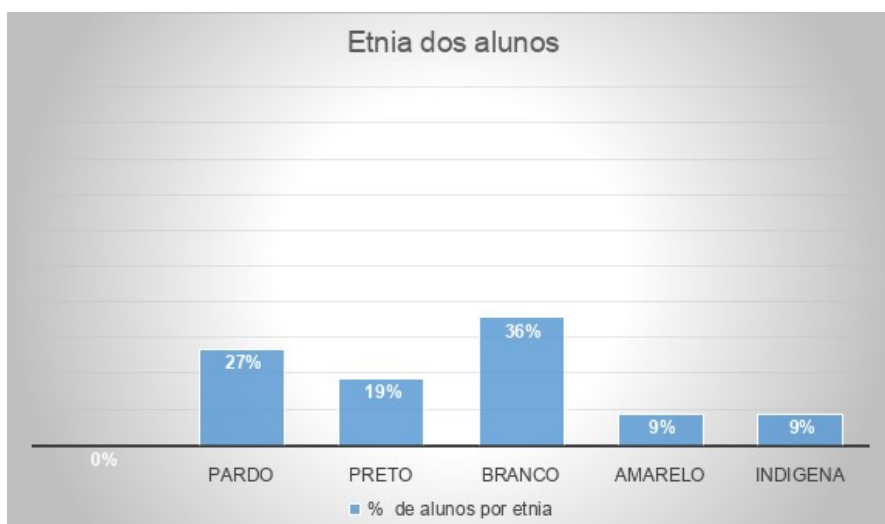
Gráfico 3 – Percentual de alunos por gênero



Fonte: da autora

O gráfico 3 mostra um maior número de alunos do gênero masculino que responderam o questionário isso implica dizer que os resultados das concepções diagnosticadas nessa pesquisa terá maior peso consideram os pensamentos e sentidos de alunos do sexo masculino, isso quer dizer que o olhar da pesquisa, os resultados, serão vistos sobre a ótica dos meninos.

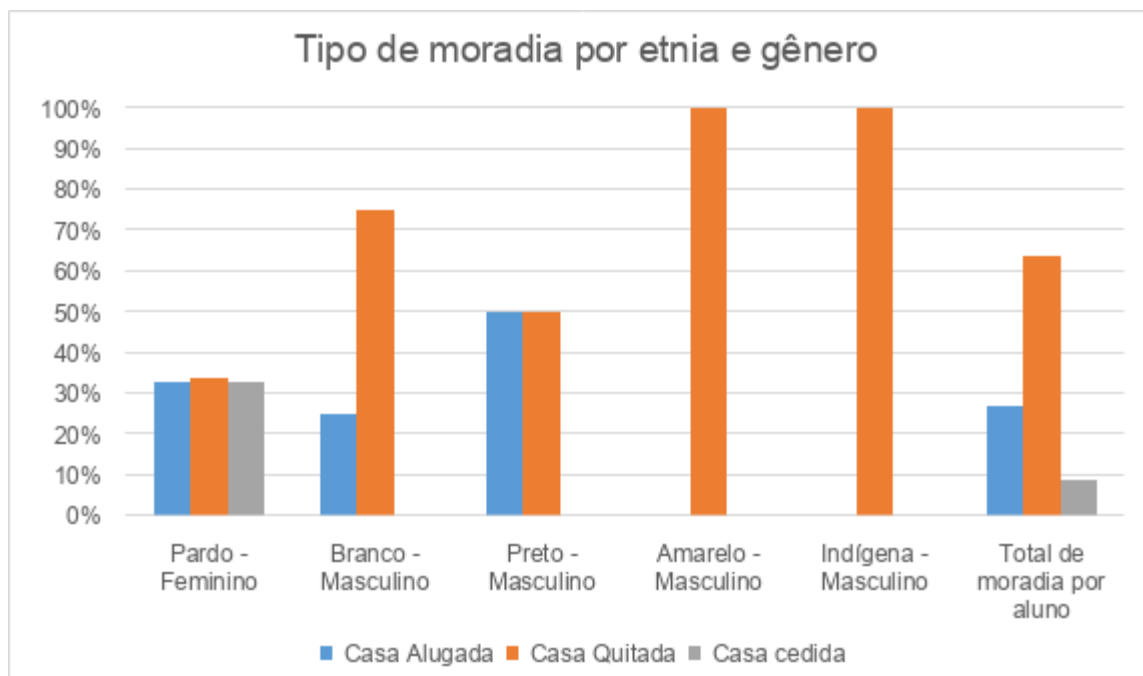
Gráfico 4 – Percentual de alunos por etnia



Fonte: da autora

O gráfico 4 apresenta que 36% dos alunos que participaram do curso *Scratch* se consideram brancos. Sendo pretos e pardos somam 46%.

Gráfico 5 – Percentual de alunos por gênero e moradia



Fonte: da autora

O gráfico acima, mostra que nosso público da pesquisa somam no total de aproximadamente 62% de alunos que moram em casa própria quitada.

Assim fazendo a análise dos gráficos podemos concluir que, o perfil do nosso universo são: meninos pardos e negros, com idade de 9 anos cursando o 4º ano do fundamental que moram em casa própria quitada.

2.1 RELATO DE EXPERIÊNCIA – O PROJETO NO CONTEXTO DO MINISTRANTE

Esta seção relata de forma empírica o desenvolvimento metodológico do projeto de extensão denominado *Scratch*: Dê vida a sua imaginação.

O projeto foi pautado na análise das práticas pedagógica e na reflexão de qual é o papel dos profissionais em computação no contexto da educação brasileira, fomentando assim um estudo aplicável de ensino da computação, dentro da área da lógica da programação para alunos do ensino básico, fazendo um recorte de métodos e metodologias para inserção no ensino fundamental I. Assim foi planejado a partir do contexto do 4º ano, a escolha dos assuntos e ferramentas adequadas com base na proposta da Sociedade Brasileira de Computação (SBC) de 2016.

Como o objetivo do projeto foi trabalhar a lógica da computação, e assim, o pensamento computacional nas crianças, através de pesquisas chegou-se a conclusão que a ferramenta que melhor desenvolveria esse objetivo era o *Scratch*, que tem um perfil e design voltado para crianças

das séries iniciais. Com a escolha da ferramenta e com o norte da proposta da SBC foi possível então dar início a escolha do ponto geográfico, a qual escola iria levar esse projeto.

O pré-requisito era que o ambiente escolar tivesse em funcionamento um laboratório de informática. A escolha da escola teve como principal ponto a infraestrutura de *hardware*, essa foi uma das questões da escolha do CEPA, além de ser um centro educacional que forneça ensino no nível fundamental I.

A partir do conhecimento estrutural da escola e já havendo feito toda parte documental do projeto com base no formulário disponibilizado no Sistema Integrado de Gestão de Atividades Acadêmicas - (SIGAA) da UFERSA, que pode ser encontrado o resumo documental do projeto. Disponível em: <https://sigaa.ufersa.edu.br/sigaa/link/public/extensao/visualizacaoAcaoExtensao/1498>. E para ter acesso a todo o documento do projeto é preciso acessar o SIGAA efetuar o *login*, e na barra principal do menu do sistema ir em "extensão".

Da fase construção até o final da execução do projeto, foi dividido por etapas. A primeira etapa foi a construção do documento contendo: resumo; justificativa; fundamentação teórica; metodologia; objetivos gerais; referências e resultados esperados.

A segunda etapa foi apresentação do projeto para a escola. Nessa apresentação foi feito o primeiro contato oficial, uma reunião expositiva com os professores e a direção da escola, essa reunião aconteceu com intuito de mostrar ao corpo docente da escola o projeto, e saber se havia interesse da escola em recebê-lo.

Com a seleção do CEPA confirmada, foi possível assim concluir a regulamentação do documento e oficializá-lo no sistema do SIGAA, ou seja, o projeto entrou em modo ativo.

Seguindo as outras etapas descritas no documento partiu-se para a seleção de alunos, para compor a equipe de membros e ministrantes do projeto.

A forma de seleção foi feita através de convites, por meio das redes sociais aos interessados e com perfil para o projeto, que eram alunos respectivamente dos cursos da LCI e BSI; após a seleção dos membros e ministrantes começou-se a fase de formação dos formadores, onde tivesse uma semana de formação com os discentes que iriam realizar as atividades na função de ministrantes.

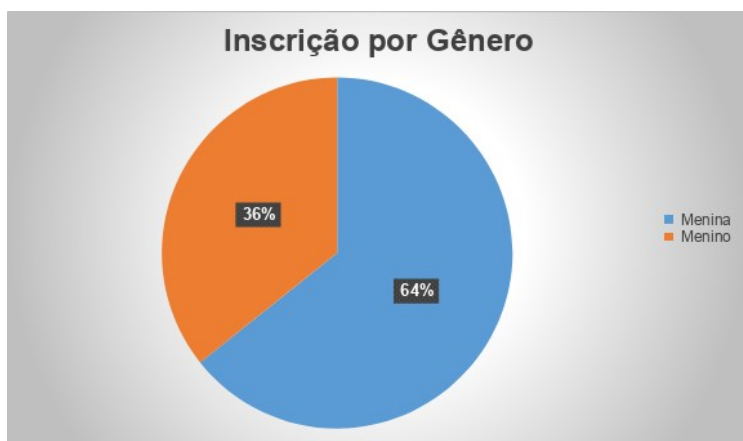
Na quarta etapa foi feito o diagnóstico dos alunos selecionados do CEPA, a partir desse diagnóstico, deu-se início ao planejamento, onde foi elaborado um plano geral de ensino, que cada equipe teria a autonomia de sua aplicabilidade e metodologias, dentro do previsto no plano geral.

No fim do planejamento, foi o início das construções dos materiais didáticos e preparação do calendário do curso, esse calendário foi entregue aos pais dos alunos, juntamente com o termo de autorização para participar do curso e uso de imagem, para fotografias, vídeos, sons e demais

documentos digitais visuais. No documento de autorização tinha um espaço para os pais ou responsáveis colocarem o número do *whatsapp*, usou-se esse aplicativo para obter um contato direto sobre possíveis contratempo, dúvidas ou qualquer informação, foi uma forma de interação que se achou importante ter com os responsáveis dos alunos.

As aulas tiveram início no dia 30 de junho de 2018 e término no dia 26 de setembro do mesmo ano. Com inscrição de 42 alunos passando assim a faixa dos 20 anos por turma.

Gráfico 6 – Percentual de alunos por gênero



Fonte: da autora

O gráfico 6 mostra o percentual de inscrição de alunos por gênero. É notável que o curso começou com um percentual com quase o dobro a mais de meninos em relação às meninas. Esse dado se torna importante, pois a partir deles podemos criar algumas hipóteses.

Hipótese um: Supondo que foi analisando todo o contexto da escola e com base nos dados de alunos com matrículas no CEPA. Fazendo um levantamento de todos os alunos matriculados e tabulando o percentual de alunos por gênero, supondo que obteve o resultado onde o percentual de meninos e meninas são equivalentes ou ainda que os resultados deram que existe um percentual maior de meninas, dentro da margem de erros, podemos dizer que: Os alunos do gênero com o percentual maior do gráfico 6 mostraram mais interesse em cursar, pela temática do curso.

Hipótese dois: Supondo que também foi analisando como na hipótese um, mas o resultado dessa análise chegou à conclusão de que os alunos do gênero masculino tem aproximadamente o dobro a mais no percentual de alunos do gênero feminino, pode-se concluir que a partir dos resultados do gráfico 6 houve um interesse igual em cursar a temática do curso.

As duas hipótese dentro do eixo, não levam em consideração outros fatores como por exemplo o contexto socioeconômico dos alunos.

Sobre os planejamentos das aulas: Entrando agora nas construções metodológicas onde o plano de aula geral foi produzido para dar um norte de como seria ministrada as aulas, e a cada dia

de aula se fazia um relatório e a partir das dificuldades, resultados que se observava nas aulas retornava-se a um novo planejamento. Por tanto o projeto teve um planejamento contínuo das aulas.

Sobre o corpo docente da escola: Ao decorrer das aulas foi perceptível que o corpo docente não iria participar dos planejamentos, a princípio o plano era ter o auxílio dos professores quanto aos assuntos que os alunos estavam estudando e montar um projeto no final do curso que envolvesse uma determinada disciplina com um conteúdo específico, para que o aluno pudesse ter uma experiência interdisciplinar com a computação e assim criar um projeto que ficasse exposto na escola e para posteriormente ser utilizado por outros estudantes, mas não houve essa colaboração, e continuou-se o projeto com as aulas práticas e conceituais da lógica da programação.

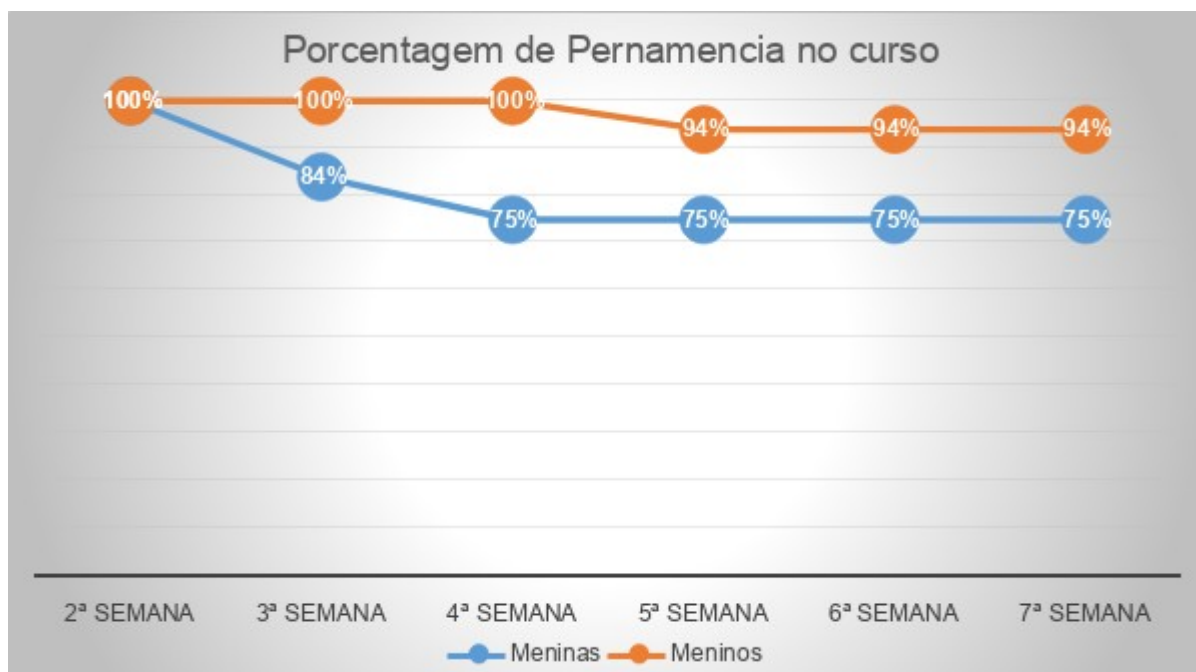
Sobre os alunos: Os alunos que continuaram até a formatura do curso eram bem empenhados, curiosos e tinham algo em comum que era a vontade de criar jogos e histórias com a ferramenta *scratch*, aprendiam rápido os comandos e sempre questionavam. Nesse aspecto era possível ter mais facilidade na hora de avaliar o contexto de cada aula e o rendimentos dos alunos.

Sobre a avaliação contínua: A cada termino do assunto estudado fazia-se um exercício prático do que havia-se estudado na aula, e era com base nesses exercícios que se fazia o diagnóstico avaliativo para o planejamento da próxima aula, outra forma que ajudava na avaliação constante era o *remember*, lembrar do conteúdo da aula anterior, nos exercícios também eram utilizado esse método.

O início das aulas eram sempre uma revisão de pergunta e respostas sobre todos os assuntos vistos, e como programação são aprendizagem cumulativa para avançar no conteúdo era preciso saber se os outros assuntos estavam de fato sendo aprendido.

Sobre a evasão dos alunos: No decorrer do curso houveram evasões o gráfico 7 mostra a porcentagem por gênero sobre uma linha temporal de dividida por semanas até o término do curso.

Gráfico 7 - Evasão do curso por Gênero.



Fonte: da autora

No gráfico 7 nas primeiras semanas 1ª até a 2ª semana, 100% dos alunos estavam frequentando as aulas regularmente. A partir da 3ª semana houve uma desistência com percentual de aproximadamente 16% das meninas, enquanto o número de meninos frequentando as aulas continuavam 100%, na 4ª semana ocorreu mais uma queda no percentual onde 9% do alunado feminino sofreu evasão.

Na 5ª semana não houve abandono do público feminino, porém cerca de 6% dos meninos desistiram do curso. Totalizando a porcentagem de meninos e meninas, assim somam uma evasão de aproximadamente 18%. Lembrando que o público de meninas no curso equivale a 36% do total de participantes e meninos 64% conforme visto no gráfico 6.

Os alunos do CEPA demonstram um perfil interativo com atividades extra curriculares, participam de projetos da comunidade e dentro da escola, como os ministrantes do projeto tinham contato direto com os responsáveis dos alunos, alguns deram suas justificativa como conflito de horários de projetos, foi uma das causas da evasão do curso, outros não se justificaram, podemos supor que a evasão se deu por outras problemáticas não temos como supor ou podemos supor que o alunado que evadiu não se identificou com o curso.

O gráfico mostra que 6% dos meninos evadiu o que é uma porcentagem bem baixa dentro do contexto que somam a maioria do público do curso, e 25% das meninas evadiu um número expressivo.

Sobre as aulas: As aulas ocorreram todas no laboratório de informática do CEPA, o laboratório foi cedido pelo IFRN – Instituto Federal do Rio Grande do Norte, como a escola recebe vários projetos e ela é aberta para a comunidade no fim das aulas dos horários matutino e vespertino, dentro dela existe uma extensão do Ifrn projeto de inclusão digital com idosos.

As aulas tiveram início dia 30 de julho de 2018 nos horários manhã e tarde. A primeira aula foi o da conhecimento e apresentação, todos os ministrantes se apresentaram e, logo após as crianças. Depois de feita as apresentações e o que cada um esperava do curso, deu-se início ao conteúdo, primeiramente foi explicado como seria realizado o projeto, seu objetivo e seus esperados resultados, feito isso os conteúdos da primeira aula foram: história das máquinas computacionais onde foi mostrado através de imagens e contado como teve início os computadores, até os dias atuais, ocorreu bastante interações e questionamentos sobre, o porquê? o pra que? e como? Eram perguntas constantes.

Foi ensinado a ligar o computador e entrar no ambiente do *scratch*, onde no ambiente foi ensinado: reconhecer os signos do ambiente, fazendo com que o aluno saiba para que serve cada ícone na tela do programa, abrir e fechar as abas, criando e modificando os atores do programa assim como o cenário no mundo do ator. As aulas que se seguiram foi-se aprendendo toda a lógica, o que era um algoritmo e tudo em conjunto ao sintaxe da linguagem do *scratch*.

Quadro 1 – Imagens Primeiro dia de curso





Fonte: da autora

Quadro 2 - Imagens apresentação dos projetos

Último dia do curso - Apresentação dos Projetos



Fonte: da autora

No quadro 2 temos as imagens do último dia do curso, onde os alunos trouxeram as ideias discutidas nas aulas passadas que antecederam a última aula, construíram seus projetos e apresentaram suas ideias, tivemos desde jogos, animações com conversação de personagens e quizz.

A apresentação consistia em falar sobre o que seu projeto fazia, porquê da sua escolha e mostrar os comando usados.

2.2 PROJETO DE EXTENSÃO – *SCRATCH*: DÊ VIDA A SUA IMAGINAÇÃO

Seção ficará a par da organização e objetivos do projeto de extensão, relatando cada fase do projeto até a sua perspectiva dos resultados esperados.

O projeto teve como perspectiva proporcionar a formação de conceitos básicos da computação relacionados à sua prática e programação, desenvolvimento da organização do pensamento computacional de forma interdisciplinar ao conteúdo do currículo cotidiano de alunos do ensino fundamental do Centro Educacional Doutor Pedro Amorim, localizada no município de Açu. Por não exigir o conhecimento prévio de outras linguagens, o *scratch* é ideal para pessoas que estão começando a programar. Além disso, foi desenvolvido para ajudar crianças e adolescentes acima de oito anos, no aprendizado de conceitos matemáticos e computacionais.

Com o *scratch* é possível criar histórias animadas, jogos e outros programas interativos que podem contribuir no desenvolvimento da aprendizagem em todas as áreas do conhecimento. O projeto, teve uma carga horária de 256 horas, distribuídas no período de 30 de junho a 26 de setembro de 2018, foi realizado de forma interativa, integrando Escola parceira e Universidade.

A escola foi convidada não apenas a participar do projeto, mas a exercer uma ação proativa, planejando, acompanhando, avaliando e sugerindo alternativas que visem aperfeiçoar o projeto, quando for o caso.

A metodologia da proposta teve como princípio a prática vinculada ao desenvolvimento do pensamento computacional. A proposta era a interdisciplinaridade do conteúdo dado em sala de aula vinculada com o ensino da programação através da ferramenta usada, desta forma seria necessária uma forte interação com a escola e principalmente com os professores.

O projeto tinha formulado a formação de duas turmas de 20 alunos cada, onde foi ministrada duas aulas semanais de 45 minutos no contra turno dos alunos. Para tanto, foi dividido o processo nas seguintes fases:

1. Seleção por meio de convite a discentes do curso de LCI e BSI para participar do projeto;
2. Apresentação do projeto à escola e formalização da parceria;
3. Formação dos formadores;
4. Divulgação na escola e inscrição dos interessados;
5. Início do curso para os alunos inscritos na escola;
6. Seminário de avaliação do projeto;
7. Exposição dos projetos elaborados durante o curso;
8. Elaboração de relatórios;

9. Produção de trabalhos acadêmicos, produção científica, artigos e resumos científicos;
10. Seminário de avaliação final do projeto.

Ao final do projeto queria-se que as crianças que participaram do curso, conseguissem minimamente compreender o que é um algoritmo e usar os passos básicos de algoritmos para a resolução de problemas ao projetar soluções. Descrever e analisar uma sequência de instruções a ser seguidas. Colaborar com colegas em grupo para criar, desenvolver, publicar e apresentar projetos, utilizando *Scratch*. Usar os passos básicos da engenharia de software (análise, projeto, implementação e teste) para criar o projeto de um jogo com tema de história utilizando conceitos básicos de programação (variáveis, operadores, estruturas condicionais e estruturas de repetição).

3. LÓGICA DE PROGRAMAÇÃO E LINGUAGEM DE PROGRAMAÇÃO

Nesta seção trataremos sobre alguns conceitos necessários a compreensão do nosso objeto de estudo. Destacamos o conceito de lógica, lógica da programação, veremos a sua estrutura e os paradigmas, tendo em vista de como se constitui um algoritmo, e também adentrar na história das linguagens de programação; o que se trata e ver exemplos de linguagens e qual a usabilidade de algumas delas.

A lógica é um campo de estudo da Filosofia que se dedica a entender as relações linguísticas que tornam uma proposição válida ou inválida no interior de um argumento. (PORFÍRIO, 2018). Existem diversos tipos de lógica, dentre elas podemos ter a lógica filosófica, a matemática e a computacional que é uma derivação da lógica da matemática.

A palavra lógica vem do grego *logiké*, e seu significado é “arte de raciocinar” (EDITORA, 2003). Segundo o Dicionário Aurélio

“a lógica é um conjunto de estudos que visam a determinar os processos intelectuais que são condição geral do conhecimento verdadeiro. Isso significa que o campo da lógica está inserido na coerência de raciocínio de ideias, seja na maneira individual de raciocinar de cada pessoa ou um grupo, onde a lógica de uma criança é distinta da lógica de um adulto que por sua vez é diferente de uma pessoa com transtornos mentais e assim sucessivamente.” Editora (2010).

Junior (2009) aborda que a lógica trata da correção do pensamento, com isso podemos dizer que ela ensina a colocar ordem nos nossos pensamentos e diferencia porque pensamos dessa forma e não de outra forma.

Entrando no campo específico da Lógica de programação, que é a técnica de desenvolver sequências lógicas para atingir um determinado objetivo. Podemos entender com Souza et al. (2012) que lógica de programação é a organização e planejamento de instruções, em um algoritmo, com o objetivo de tornar viável a implementação de um programa ou software.

Sendo assim, pensa-se com uma boa lógica de programação existe um conjunto de fatores a serem empregados, tais como: a organização; criatividade; perseverança; padronização e otimização. Todas essas características são relevantes para a construção dos algoritmos. (FORBELLONE; EBERSPÄCHER, 2005).

“A palavra “algoritmo” é derivada do nome Mohammed ibn Musa Al-Khowarizmique, que foi um matemático, astrólogo, astrônomo e autor persa. Ele fez parte de um centro acadêmico conhecido como a Casa da Sabedoria, em Bagdá, por volta de 800 d.C. Seus trabalhos introduziram o cálculo hindu aos árabes e, a partir daí, ao resto da Europa.” (SOUSA; JÚNIOR; FORMIGA, 2014).

Podemos definir algoritmos como um conjunto finito de regras que oferece uma sequência de operações para resolver um tipo de problema específica, onde não pode haver ambiguidade. (SOUZA ET AL, 2012).

Para compreendermos melhor o que é um algoritmo, mostrando na prática com um exemplo de algoritmo que é corriqueiro do nosso dia a dia.

Algoritmo: Batata frita

Início

1. Pegar as batatas
2. Descascar as batatas
3. Cortar as batatas em tiras
4. Pegar uma panela
5. Colocar óleo na panela
6. Acender o fogo do fogão
7. Esperar o óleo esquentar
8. Colocar as batatas para fritar
9. Aguardar o tempo ideal até que as batatas estejam fritas
10. Remover as batatas fritas e servir

Fim

Outro tipo de algoritmo é o fluxograma, Segundo Silva (2012)

Fluxograma é um tipo de diagrama, e pode ser entendido como uma representação esquemática de um processo, muitas vezes feito através de gráficos que ilustram de forma descomplicada a transição de informações entre os elementos que o compõem, ou seja, fluxograma é um gráfico que demonstra a sequência operacional do desenvolvimento de um processo, o qual caracteriza: o trabalho que está sendo realizado, o tempo necessário para sua realização, a distância percorrida pelos documentos, quem está realizando o trabalho e como ele flui entre os participantes deste processo.

Mas especificamente, fluxogramas podem:

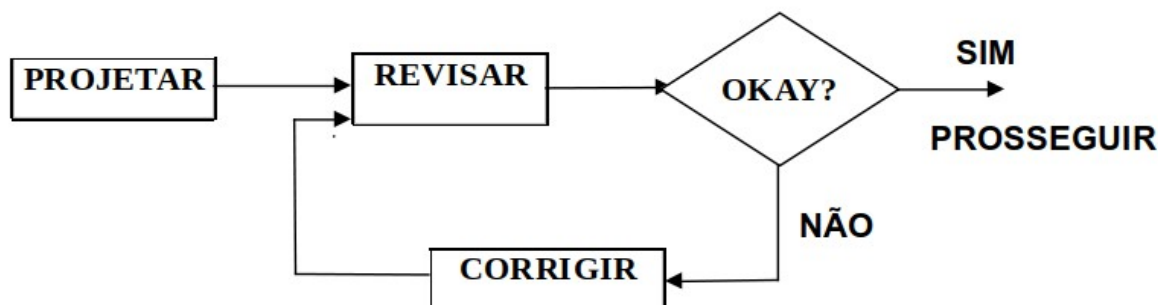
- Demonstrar a forma como o código é organizado.
- Visualizar a execução do código dentro de um programa.

- Mostrar a estrutura de um site ou aplicativo.
- Entender como usuários navegam um site ou programa.

O fluxograma pode também ser definido como o gráfico em que se representa o percurso ou caminho percorrido para alcançar um objetivo. (SILVA, 2012).

Fluxogramas são fáceis de entender pois são algoritmo visuais simplistas do código ou tarefa. Como mostra o fluxograma genérico na figura 1.

Figura 1 - Fluxograma do que é um algoritmo



Fonte: Autoria própria.

Existem diversas formas de representação de um algoritmo como por exemplo: formas gráficas, forma textual e como é o caso dos algoritmos acima pseudocódigo e fluxograma.

Para compreender melhor os algoritmos é preciso estudar seus paradigmas, na ciência da computação chamamos de “Paradigmas de programação” onde cada linguagem de programação é caracterizada por um ou mais paradigmas. As que possuem mais de um paradigma, são as chamadas multi paradigmas.

“Um paradigma de programação está relacionado com a forma de pensar do programador na construção de soluções para os problemas com os quais se depara. Programar seguindo um determinado paradigma de programação significa representar soluções a partir de uma forma particular de raciocinar na elaboração dos algoritmos.” Sousa, Júnior e Formiga (2014).

O paradigma de uma linguagem de programação é a sua identidade. Corresponde a um conjunto de características que, juntas, definem como ela opera e resolve os problemas. Tedesco (2016).

Assim como existem muitas linguagens de programação, também existem muitos paradigmas de programação, nesse texto vamos destacar alguns dos paradigmas. São elas: Funcional; Lógico; Declarativo; Imperativo; Orientado a objetos.

Cada qual tem sua aplicação e pode ser utilizado dependendo da necessidade do software. É comum aplicações de linguagens multi paradigmas utilizarem mais de um. Tedesco (2016).

Tabela 1 – Classificação, conceito e Paradigmas de linguagens.

Paradigma	Conceito	Linguagens de Programação
Imperativo	Ele é focado na mudança de estados de variáveis. (sequência, atribuição, estado). Tedesco (2016)	Basic, Pascal, C, Ada, Fortran, Cobol, Assembly, C++, PHP, Perl, C#, Ruby...
Funcional	O foco desse paradigma está na avaliação de funções. Como na matemática quando temos, por exemplo, uma função $f(x)$. (função, aplicação, avaliação). Tedesco (2016)	Lisp, Haskell, Erlang, Scheme, F# (da Microsoft), Elixir, Mathematica...
Lógico	Utilizado em aplicações de inteligência artificial. Esse paradigma chega no resultado esperado a partir de avaliações lógico-matemáticas. (relação, dedução). Tedesco (2016)	Prolog.
Orientado a Objetos	Na orientação a objetos utilizamos uma lógica bem próxima do mundo real, lidando com objetos, estruturas que já conhecemos e sobre as quais possuímos uma grande compreensão. (objeto, estado, mensagem). Tedesco (2016)	C++, Java, C#, Eiffel, Smalltalk, Python, PHP, Ruby...
Declarativo	O paradigma declarativo é baseado no lógico e funcional. Linguagens declarativas descrevem o que fazem e não exatamente como suas instruções funcionam. Tedesco (2016)	HTML, XML, XSLT, XAML.

Fonte: da autora

Neste ponto, pode-se compreender os conceitos de algoritmo, paradigma de programação, para poder compreender o conceito de linguagem de programação. Porém antes tem-se que ser entendido o conceito de linguagem. Segundo a linguística, trata-se de qualquer meio sistemático de comunicar ideias ou sentimentos através de signos convencionais, sonoros, gráficos, gestuais, onde toda linguagem precisa ter regras sintáticas e semânticas.

(Peirce, 1977) apud Koslosky (1999) , mostra outra teoria mais aprofundamento que falam: A linguagem é um sistema simbólico e o homem é único animal capaz de criar símbolos, isto é, signos arbitrários em relação ao objeto que representam, e por isso mesmo, convencionais, ou seja,

dependentes da aceitação social. Na medida que este laço entre representação e objeto representado é arbitrário, ele é uma construção da razão. (Peirce, 1977).

Na computação, também surgiu a necessidade de se criar um método padronizado para expressar instruções sobre as quais um computador vai atuar, como elas serão armazenadas ou transmitidas e quais ações devem ser tomadas sob várias circunstâncias. (PEREIRA, CAMILA, SANTIAGO, 2009).

Linguagem de programação, segundo o Dicionário online de português é o conjunto de regras que, criadas artificialmente, servem para dar instruções padronizadas a um computador, permitindo que os programadores codifiquem suas intenções mais facilmente.

Ou seja, essa linguagem usada pelo homem para desempenhar comunicação com o computador, pois o computador não reconhece a linguagem natural do ser humano. Por tanto essa linguagem é usada para obter uma comunicação com as máquinas computacionais e assim manipulá-las e criando projetos que possam facilitar a vida humana.

Hoje em dia há milhares de linguagens de programação, mas nem sempre foi assim, a história fala que os primeiros computadores digitais surgiram no final dos anos 1940 no início dos anos 1950, com isso houve a necessidade de criação de linguagens que facilitasse a vida do programador, já que as máquinas eram apenas usadas para fazer cálculos e totalmente manuais, diferente dos dias atuais que são totalmente automatizadas.

Para que a relação homem versus máquina pudesse evoluir, foi necessário a criação de uma linguagem que tornasse possível a comunicação com ela, ou seja tornasse a máquina operacional. A partir do desenvolvimento da linguagem de programação é que o ser humano passou a obter grandes resultados com a máquina.

Tucker nos demonstra que:

As primeiras linguagens de programação foram as linguagens de máquina e as linguagens *assembly* dos primeiros computadores, começando na década de 1940. Centenas de linguagens de programação e dialetos foram desenvolvidos desde então. A maioria teve tempo de vida e utilidade limitados, enquanto algumas tiveram amplo sucesso em um ou mais domínios de aplicações. Muitas desempenharam um papel importante na influência sobre o projeto de futuras linguagens” Tucker (2009, p.6).

A linguagem *assembly* ou ASL, é uma linguagem de baixo nível de programação, se denomina linguagem de baixo nível as linguagens que se distanciam da linguagem natural do ser humano. Tucker (2009), descreve que uma linguagem de alto nível é aquela que se aproxima mais com nossa linguagem humana.

A *assembly* usa comandos de programação para se comunicar com a interface do *hardware* do computador, sabemos que o computador só compreende números binários no caso 0 e 1, pois dentro do hardware do computador só existe dois estados de energia que se passa, o impulso de energia 1, ou quando não há energia representado pelo 0, então temos que: 0 - desligado e 1- ligado.

A linguagem assembly usa comandos estruturados em vez de números e permite que os programadores consigam ler mais facilmente o código. Embora o código assembly seja mais fácil de ler do que o código binário, a linguagem assembly é uma linguagem difícil. (MARTINS, 2017).

A linguagem de programação *Assembly* pode ser considerada como uma família de linguagens já que cada arquitetura de processador possui a sua. Freire (2009).

Apesar do substancial declínio da popularidade da linguagem, na atualidade, ela ainda é usada em diversas áreas, destacando os sistemas embarcados, *boot loaders*, *drivers* de dispositivos, emuladores, ou seja, em áreas cujo desempenho do programa precisa ser máximo, devido ao fato de ser a linguagem legível por humanos capaz de produzir os menores e mais rápidos programas. Freire (2009).

3.1 FERRAMENTA DE PROGRAMAÇÃO COMO MÉTODOS DE ENSINO

As ferramentas de programação começaram a ser desenvolvidas juntamente com o desenvolvimento dos computadores digitais nos anos 1950.

Ferramenta de programação e ambiente de programação são duas coisas diferentes, as ferramentas estão inseridas no ambiente de programação, portanto, são *softwares*, programas, aplicativos ou plataformas que permitem o programador desenvolver projetos em uma determinada linguagem de programação.

Geralmente essas ferramentas são destinadas a uma linguagem específica, porém existem ferramentas editores que se adequam a qualquer linguagem. São como editores de textos, porém o texto que irá utilizar não são linguagem natural e sim linhas de código onde essas linhas são conhecidas como algoritmos e se tornaram programas.

Um ambiente de programação é a coleção de ferramentas usadas no desenvolvimento de software. Essa coleção pode consistir em apenas um sistema de arquivos, um editor de textos, um ligador e um compilador. (SEBESTA, 2018, p.29). Mas também pode-se incluir uma grande coleção de ferramentas integradas.

Logo, as características de uma linguagem de programação não são a única medida da capacidade de desenvolvimento de um sistema. (SEBESTA, 2018, p.29).

O mais antigo ambiente de programação é o UNIX, ele foi criado a partir de um sistema operacional nos anos 1970. Como ambiente ele tinha muitas ferramentas para suporte de produção e manutenção de software, sem contar que o UNIX vinha com uma variedade de linguagens de programação.

O UNIX era uma interface uniforme entre suas ferramentas. Isso o tornava mais difícil de aprender e usar. (SEBESTA, 2018, p.29).

Na atualidade existem várias ferramentas tanto para construção de projetos destinados aos desenvolvedores quanto para o ensino de programação. Abaixo um levantamento de algumas ferramentas de ensino da programação separado por categorias de Grau escolar, são elas:

Tabela 2 – Classificação de uso de ferramentas

Ferramentas Para o Ensino da Programação			
Ferramentas	Ensino Fundamental I	Ensino Fundamental II	Ensino Médio
<i>Scratch</i> Foi projetado especialmente para idades entre 8 e 16 anos, mas é usado por pessoas de todas as faixas etárias. About... (2016)	X	X	X
<i>Algorithm City</i> O jogo é um jogo de codificação divertido e inovador para ensinar às crianças entre 6 a 12 anos, as noções básicas de programação e algoritmo. Os conceitos básicos de codificação, como sequenciamento de comandos, funções e loops, orientando o personagem coletando ouro e resolvendo os níveis. Musteren (2018)	X	X	
<i>Lightbot: Code Hour</i> Lightbot é um jogo de programação; ele usa a mecânica do jogo que estão firmemente enraizadas em conceitos de programação. Lightbot permite aos jogadores ganhar uma compreensão prática dos conceitos básicos como sequenciamento de instruções, procedimentos e loops, apenas guiando um robô para acender telhas e resolver níveis. Indicado para crianças de 6 a 12 anos. Llc (2016)	X	X	
<i>KTurtle</i> O KTurtle é um ambiente educativo de programação que usa o TurtleScript, uma linguagem de	X	X	

programação vagamente baseada e inspirada pelo Logo. O objetivo do K Turtle é fazer a programação tão fácil e acessível quanto possível. Isto torna o K Turtle adequado para ensinar às crianças as bases da Matemática, Geometria e... programação. Breijs (2018)			
---	--	--	--

Fonte: da autora

Um dos benefícios para as crianças e adolescentes que vão aprender conceitos de programação é o desenvolvimento do pensamento lógico e matemático. Como também o ato de desenvolver software utilizando linguagem computacional exige esse tipo de conhecimento, e é algo hoje em dia natural só observa o crescimento nessas áreas. Segundo Antonio Junior (2019).

Um dos objetivos da introdução dos computadores na vida das crianças é que esta tecnologia estimule suas mentes e potencialize seu desenvolvimento intelectual, paralelamente ao seu desenvolvimento psicossocial, uma vez que sua coordenação motora está se estabelecendo concomitantemente a seus gostos e relações sociais. Antonio Junior (2019).

No âmbito universitário necessitamos construir, definir e avaliar práticas interdisciplinares que forneçam aos nossos alunos experiências de aprendizagem que qualifiquem sua formação profissional. Lopes e Zuffo (2017).

No ensino básico a interdisciplinaridade também faz parte dentro da construção do currículo, a BNCC – Base Nacional Comum Curricular no texto do “pacto interfederativo” fala que: “decidir sobre formas de organização interdisciplinar dos componentes curriculares e fortalecer a competência pedagógica das equipes escolares para adotar estratégias mais dinâmicas, interativas e colaborativas em relação à gestão do ensino e da aprendizagem;” BNCC (2019).

É preciso buscar meios e estratégias que possibilitem essa interdisciplinaridade dentro dos estabelecimentos escolares.

Figura 2 – Computação Interdisciplinar



Fonte: da autora

Segundo Sacristan (1998) apud Silva (2002), a interdisciplinaridade, tão falada, mas tão pouco praticada, não apenas leva à formação mais completa do estudante e do cidadão, preparando-o para interagir com as informações esparsas e construir seu conhecimento, como também permite uma compreensão mais profunda das próprias questões especializadas.

A interdisciplinaridade não existe por si só; ela depende da interpretação, aplicação e avaliação feitas na prática cotidiana de cada professor. Lopes e Zuffo (2017).

O ensino da abstração, do reconhecimento de padrões, da decomposição de problemas, da detecção de erros e do sistema de símbolos e representações colabora para a aprendizagem de alunos de maneira interdisciplinar. Brackmann e Cassola (2017).

A Ciência da Informação, é a interdisciplinaridade, associada à área desde sua origem, tornando-se uma marca de sua evolução e amplamente reconhecida por sua comunidade. Bicalho e Oliveira (2011).

Pode-se dizer que a computação é uma ciência que pode ser utilizada como método bem apropriado para a aplicação da interdisciplinaridade, pois nela corrobora, hoje, principalmente como o maior uso para se ter acesso e compartilhamento de conhecimento das outras ciências.

3.2 SCRATCH

O *Scratch* foi desenvolvido pelo Instituto Tecnológico de Massachussets (MIT, experiente no desenvolvimento de ferramentas educativas para crianças na idade escolar) e pelo grupo KIDS da Universidade de Califórnia, Los Angeles.

Scratch é uma nova linguagem de programação, que permite a criação de histórias, animações, jogos e outras produções. Linguagem de programação é a linguagem usada para dar instruções ao computador, ou seja, é com a linguagem de programação que conseguimos fazer o computador seguir nossas ordens.

A programação em *Scratch* é feita a partir de comandos de blocos lógicos que devem ser agrupados como peças de Lego. Imagine um grande quebra-cabeças com várias possibilidades e resultados diferentes, a linguagem de programação *Scratch* é exatamente isso, a diferença é que no quebra-cabeças só há uma possibilidade de resultados, no *scratch* há infinitas possibilidades.

O *Scratch* foi criado com foco nas crianças para desenvolverem e aprenderem de maneira fácil e divertida a lógica da programação de computadores. Com o *scratch* crianças e pessoas de qualquer outra faixa etária sem nenhum conhecimento em computação podem aprender a utilizar a linguagem e a ferramenta, por se utilizar de uma interface gráfica (a interface gráfica é a parte

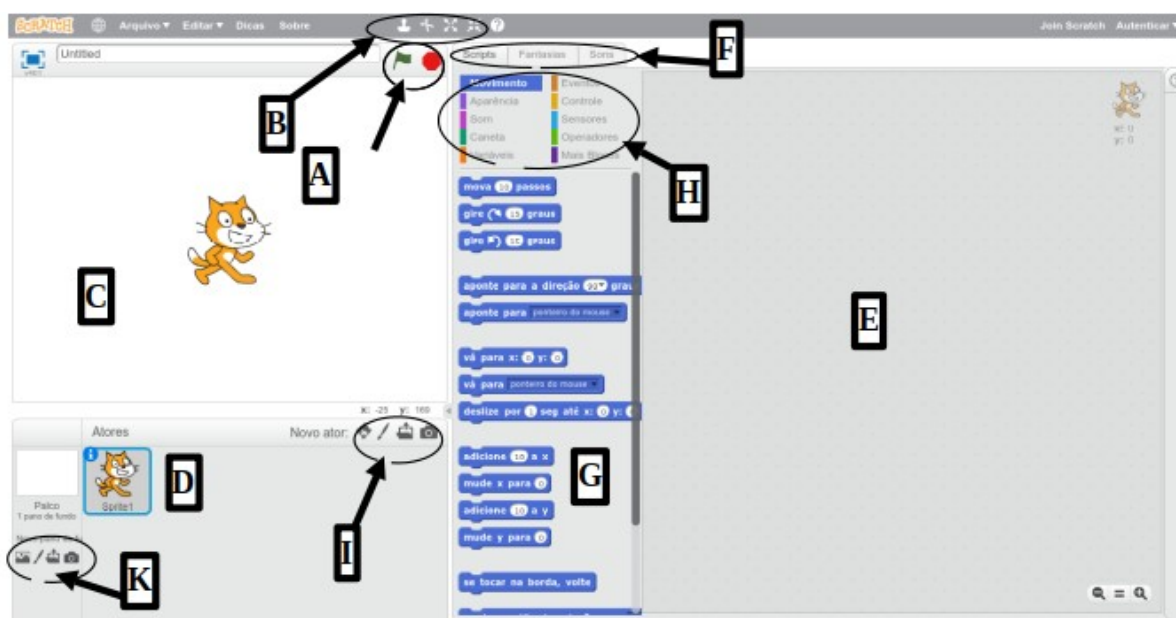
visual, no caso do *Scratch* são os blocos de comandos) que permite que programas sejam construídos com blocos encaixados.

Tratando-se de uma ferramenta livre e gratuita, para baixar o *Scratch* no computador ou notebook pode acessar no site disponível em: <http://scratch.mit.edu/download>. E após preencher um formulário você poderá escolher a versão para o download, ou utilizá-lo online através do site oficial.

Segundo o scratch.mit.edu, o *scratch* é uma comunidade de aprendizagem criativa presente em mais de 150 países com mais do que 11 milhões de usuários e mais de 3 milhões de projetos compartilhados. No *Scratch* existe uma comunidade muito pequena de falantes de língua portuguesa em comparação à quantidade total, mas os poucos que se encontram programando e compartilhando esta linguagem de programação, já possuem um conhecimento avançado na ferramenta.

No ano de 2016, o *Scratch Foundation* anunciou uma parceria com a *Google* para a nova geração da ferramenta (versão 3.0) no ano de 2018 podíamos ter acesso a versão para testes, mas foi no ano de 2019 que houve seu lançamento substituindo a versão anterior 2.0 que ainda pode ser utilizada offline assim como 1.0, baixando e instalando no computador. Esta nova versão é baseada em tecnologia web nativa (HTML 5) e conta com uma mudança na sua gramática

Figura 3- Ambiente de programação *scratch* 2.0



Fonte: scratch.mit.edu

A) Botões para iniciar (bandeira verde) e parar o *script* (círculo vermelho).

B) Botões para editar o ator selecionado no palco (clonar, apagar, aumentar e diminuir).

- C) Palco onde os atores são colocados e onde é possível ver o resultado da Programação criada. O ator inicial que aparece no palco é o gato o símbolo e a logo do *scratch*.
- D) Área dos objetos usados na animação. Objeto em edição fica selecionado.
- E) Área de edição e conexão de *scripts*.
- F) Abas com opções para a área de *script*, para fantasias e para sons.
- G) Blocos de comandos.
- H) Categorias de comandos: Movimento, Aparência, Som, Caneta, Variáveis, Eventos, Controle, Sensores, Operadores.
- I) Importar atores da biblioteca, editar ator, importar atores do computador e criar ator através da *webcam*.
- K) Importar pano de fundo da biblioteca, editar pano de fundo, importar pano de fundo do computador e criar pano de fundo através da *webcam*.

3.3 COMPUTAÇÃO NO ENSINO FUNDAMENTAL

O rápido crescimento em tecnologia computacional que vem ocorrendo pelo mundo, cria-se com ele a necessidade de ensinar nas escolas desde a educação básica as teorias conceituais e fundamentais da Ciência da computação como forma de melhorar o aprendizado e possibilitar o uso mais eficiente dessas tecnologias em benefício da sociedade como um todo.

Desenvolver na Educação Básica projetos de programação de criação de jogos digitais é uma porta de entrada para começar a inserção do ensino da ciência da computação, cursos de computação já existem no Brasil desde 1969 na graduação, mas os de Licenciatura em informática e Licenciatura em computação são recentes no país, com média de 10 anos. Como outra ciência que aprendemos na escola o Brasil está muitos passos atrás, países bem desenvolvidos já utilizam do estudo dessa ciência desde cedo nas escolas.

"A computação está impulsionando a criação de empregos e a inovação em toda a nossa economia e sociedade. Hoje, para serem cidadãos bem educados num mundo permeado de Tecnologia da Informação, todos devem possuir uma compreensão clara dos princípios e práticas da computação além do simples uso da TI." von Wangenheim, Nunes e Santos (2014, p. 116).

Segundo (VON WANGENHEIM; NUNES; SANTOS, 2014) é importante que se compreenda a computação como ciência, pois isso permitirá que os alunos sejam consumidores educados de tecnologia, assim como também criadores inovadores, sendo capazes de projetar sistemas de computadores para melhorar a qualidade de vida de todas as pessoas.

É também na compreensão de como funciona o mundo digital que se compreende o que é algo mecânico e nos torna mais humanos, e é uma habilidade importante em um mundo digital,

compreender como as coisas funcionam minimamente no mundo digital é algo relevante nos dias atuais.

Segundo pesquisas passamos em média quase cinco horas e 12 minutos no mundo virtual por dia e no Brasil cerca de 100 milhões de pessoas estão conectados em alguma ou mais redes sociais, isso gera lucro. “O ensino da computação nas escolas é essencial para criar essa ampla compreensão pública, o que também auxiliará o atendimento às necessidades de crescimento da força de trabalho”. von Wangenheim, Nunes e Santos (2014, p. 116).

O desafio de ensinar computação no Brasil é o fato da ciência da computação principalmente no campo da programação tem toda sua raiz e métodos estrangeiros, o ensino introdutório se torna bem mais complicado na graduação, por isso que essa ciência deve começar da base do ensino para que haja desde o início uma afetividade com a ciência o conhecimento.

O CSTA K-12 Computer Science Standards (CSTA, 2011) destaca a importância do desenvolvimento de habilidades computacionais na educação básica que promovam a capacidade de resolução de problemas, para que possam auxiliar outras ciências, e motivar os alunos. (RAMOS; TEIXEIRA, 2016).

A comissão de Educação da SBC, com a colaboração da Comissão Especial de Informática na Educação, elaboraram em março de 2016, uma sugestão de conteúdos para a BNCC - Base Nacional Comum Curricular.

Para o ensino fundamental eles dividiram da seguinte forma:

Ensino fundamental – 1ª ao 3ª

- (a) Desenvolver pensamento computacional - Identificar e executar sequência de passos de uma tarefa. Criar passos para solução de problemas.
- (b) Reconhecer e representar dados - Classificar objetos de diferentes formas de acordo com suas características.
- (c) Desenvolver fluência tecnológica - Identificar a presença da computação no cotidiano das pessoas. Usar Desenvolver fluência tecnológica aplicativos que envolvam pensamento computacional de acordo com a faixa etária.

Ensino fundamental – 4ª ao 6ª

- (a) Compreender a noção de algoritmo - Representar algoritmos através de Compreender a noção de algoritmo imagens e palavras. Criar algoritmos para movimentar algum objeto seguindo instruções.
- (b) Identificar tipos de dados - Descrever movimentos de objetos reais utilizando diagramas. Identificar textos, números, sons imagens, como sendo dados.
- (c) Aprofundar o desenvolvimento da fluência tecnológica - Distinguir objetos eletrônicos de não eletrônicos. Distinguir diferentes interfaces de dispositivos computacionais.

Porém essa sugestão não foi incluída nos documentos da BNCC. Quando se ler os documentos da Base Nacional Comum Curricular, é tratado com ênfase às questões referentes à inovação e tecnologia, na competência número 4 do ensino fundamental, e estipula o uso de diferentes linguagens. Antes de mencionar as competências, no “4.1 a área de linguagens” segundo documento da BNCC: “As linguagens, antes articuladas, passam a ter *status* próprios de objetos de conhecimento escolar. O importante, assim, é que os estudantes se apropriem das especificidades de cada linguagem, sem perder a visão do todo no qual elas estão inseridas.” BNCC (2019).

Com base no que está escrito no documento, podemos incluir também as linguagens de programação como uma linguagem de objeto de estudo dentro desse currículo, Papert (1985) já fazia esses questionamentos embasado, segundo ele:

"Programar significa, nada mais, nada menos, comunicar-se com o computador numa linguagem que tanto ele quanto o homem podem "entender". E aprender línguas é uma das coisas que as crianças fazem bem. Toda criança normalmente aprende a falar. Por que então não deveria aprender a "falar" com um computador ?" Papert (1985).

Então a BNCC entende como competências específicas de linguagens para o ensino fundamental, na competência 3 fala:

“Utilizar diferentes linguagens – verbal (oral ou visual-motora, como Libras, e escrita), corporal, visual, sonora e digital –, para se expressar e partilhar informações, experiências, ideias e sentimentos em diferentes contextos e produzir sentidos que levem ao diálogo, à resolução de conflitos e à cooperação.” BNCC (2019).

A abrangência do texto torna o currículo bem flexível, não havendo restrições para a inserção do ensino de linguagens de programação no currículo do básico fundamental. No geral podemos ver que “Embora as diretrizes da BNCC ultrapassem os aspectos tecnológicos, o documento reconhece que essas ferramentas podem ser grandes aliadas no desenvolvimento de habilidades cognitivas e socioemocionais com as crianças.” Mello (2018).

Fica claro na competência específica 6, linguagens para o ensino fundamental:

Compreender e utilizar tecnologias digitais de informação e comunicação de forma crítica, significativa, reflexiva e ética nas diversas práticas sociais (incluindo as escolares), para se comunicar por meio das diferentes linguagens e mídias, produzir conhecimentos, resolver problemas e desenvolver projetos autorais e coletivos. BNCC (2019).

Pode-se dizer que os documentos da BNCC apesar de não especificar como objeto de estudo, ou dentro dos conteúdos da base como ciência, os textos nos dá a liberdade e moldar esse currículo e ir inserindo esses conteúdos nas aprendizagem dos alunos.

4. METODOLOGIA DA PESQUISA

Pode-se definir pesquisa como o procedimento racional e sistemático que tem como objetivo proporcionar respostas aos problemas que são propostos. Gil (2002, p.17).

A metodologia de pesquisa utilizada será a pesquisa-ação no qual é exigido o envolvimento ativo e participativo do pesquisador, das pessoas e/ou grupos envolvidos no problema (GIL, 2002, p. 55).

A pesquisa é sempre desenvolvida através do concurso concentração dos conhecimentos permitidos e a utilização cuidadosa de métodos, técnicas e outros processos científicos. O fato é que a pesquisa desenvolve-se ao longo do processo que envolvem inúmeras fases, desde a adequada formulação do problema até a satisfatória apresentação dos resultados. (GIL, 2002).

Para a realização dessa pesquisa-ação foi optado como instrumento metodológico, por meio de questionários pois, “Analisando-se [...], pode-se verificar que o questionário constitui o meio mais rápido e barato de obtenção de informações, além de não exigir treinamento de pessoal e garantir o anonimato”. Gil (2002, p.115).

Segue agora as concepções utilizadas nos questionários aplicados com o objeto de estudo deste trabalho.

Com a caracterização dos alunos e perfil com maior peso nas questões já definidos. As questões são divididas da seguinte forma: questões 1, 2 e 3 são referentes a memória dos alunos, o que eles lembram do curso os assuntos estudados, é uma forma de avaliação para saber se o conhecimento continua lá, “pois o conhecimento é aquilo que fica quando o esquecimento fez o seu trabalho” Ariano Suassuna. Tudo o que foi lembrado por eles, é o que realmente foi aprendido. As questões procuravam também saber se havia utilização ferramenta.

As questões 4, 5 e 6 tratam da afetividade que as crianças tiveram e ainda tem com o curso, como um todo, desde os professores e colegas até os assuntos estudados.

As questões 7 e 8 procuram saber se o que eles estudaram foi significativo na vida escolar deles, havendo ou não melhora em algum aspecto.

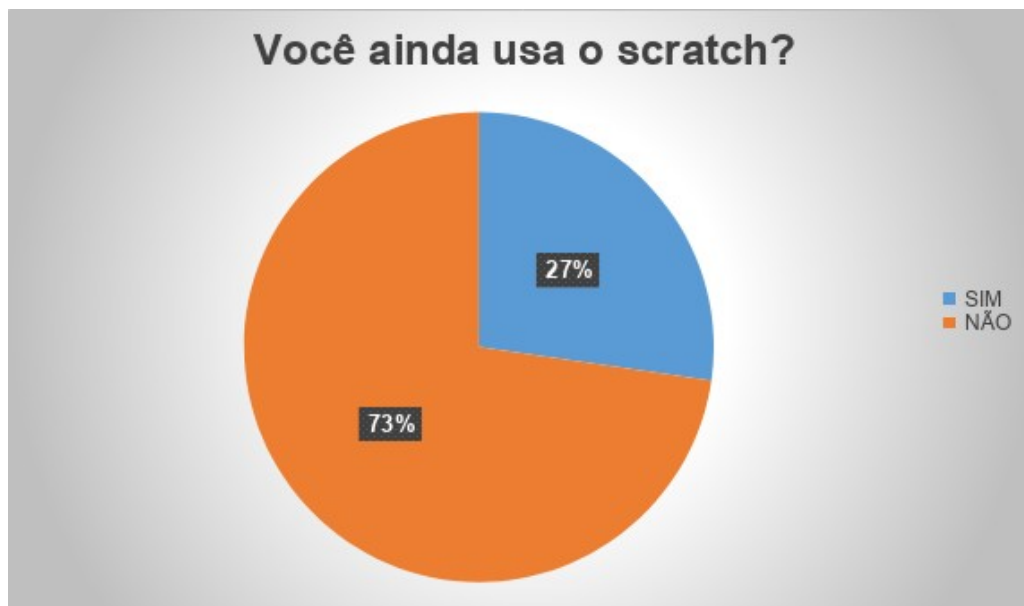
E por fim as questões 9, 10, 11 e 12 busca compreender os sentidos (o que pensam sobre) dos alunos sobre o ensino aprender linguagem de programação.

5 RESULTADOS E DISCUSSÕES

Seção irá abordar os resultados obtidos através do questionário aplicado, analisado de forma empírica feita pela autora desse trabalho. Os resultados são quantitativos e qualitativos, em forma de gráficos para melhor compreensão dos resultados.

A primeira questão tenta descobrir se os alunos utilizam a ferramenta do curso.

Gráfico 8 - Questão 1



Fonte: da autora

O gráfico 8 mostra que 73% dos alunos responderam que não, então foi questionado o porquê de não utilizar a ferramenta. Enquanto 27% afirma que ainda utilizam a ferramenta e também foi questionado.

Ainda na primeira questão, lhes foi questionado o porque de utilizar ou não a ferramenta, com isso obteve-se saída de dois gráficos. Um gráfico com as crianças que responderam “sim” e outro com as que responderam “não”.

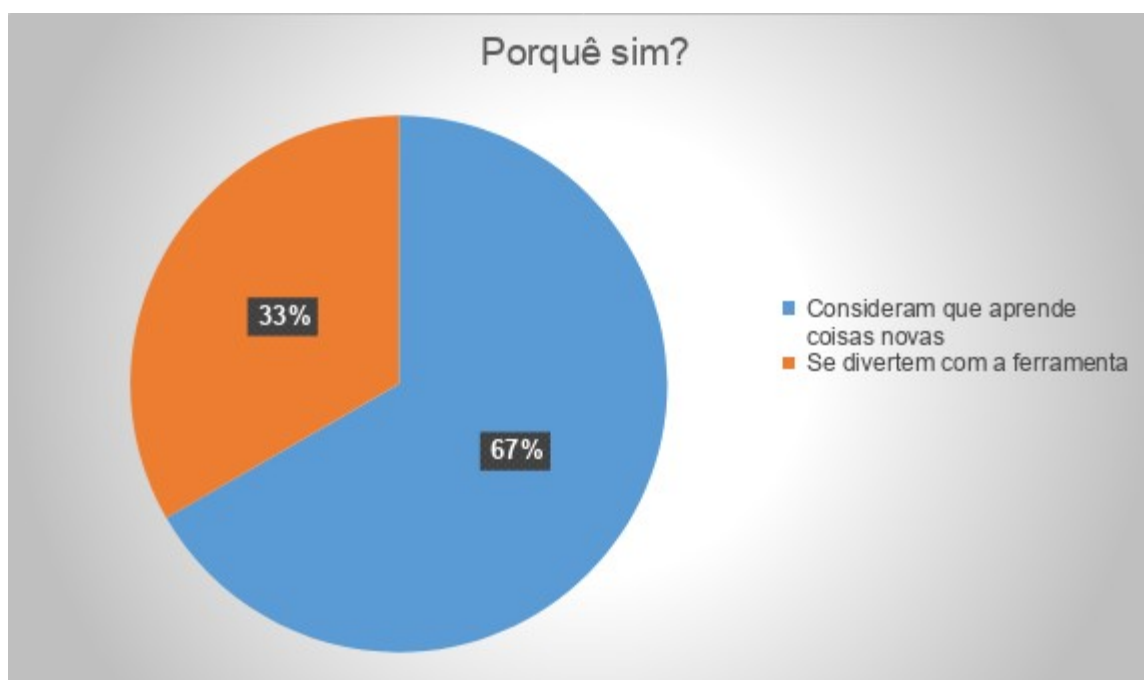
Gráfico 9 - Questão 1-A: Resposta "Não"



Fonte: da autora

O gráfico 9 representa que aproximadamente 75% dos alunos não utilizam a ferramenta *scratch*, pois não tem infraestrutura de *hardware* ou quando têm, não possuem ainda a prática de fazer *downloads* ou pesquisarem corretamente pelo *software* em seus navegadores, *desktop* ou *mobile*. O aluno X disse “porquê pesquisei e não achei, achei o errado”, O aluno Y fala que “porquê não tenho computador” e o aluno Z afirma que “porquê não tenho tempo”.

Gráfico 10 - Questão 1-A: Resposta "Sim"

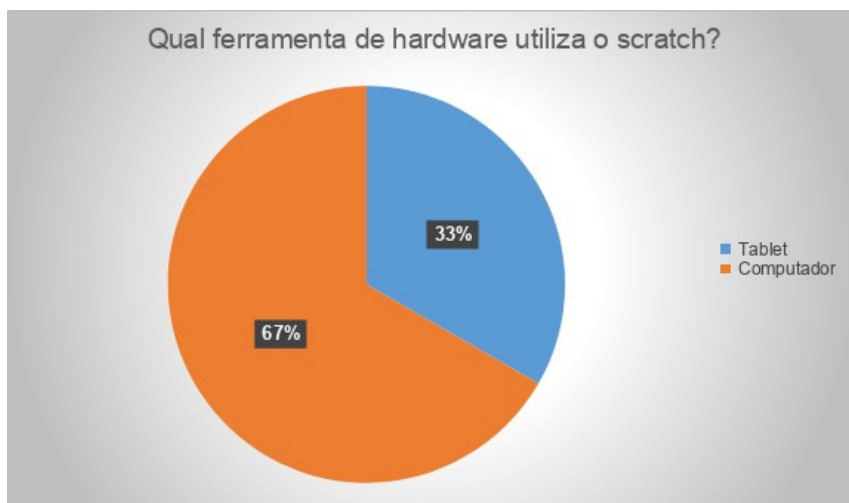


Fonte: da autora

Os 27% que responderam que utiliza mostra além de possuírem infraestrutura de hardware conseguem manusear, utilizar, pesquisar e fazer *download* do *software*. Além de mostrar curiosidade e considerar que a ferramenta é algo educativo e divertido. O aluno XX diz “porquê eu gosto de aprender e porquê aprendo muito com ele”, o aluno ZZ fala que “porquê acho bom divertido”.

Ainda na questão 1 buscou-se saber das crianças que responderam que usa a ferramenta, teve-se o seguinte resultado.

Gráfico 11 - Questão 1-B

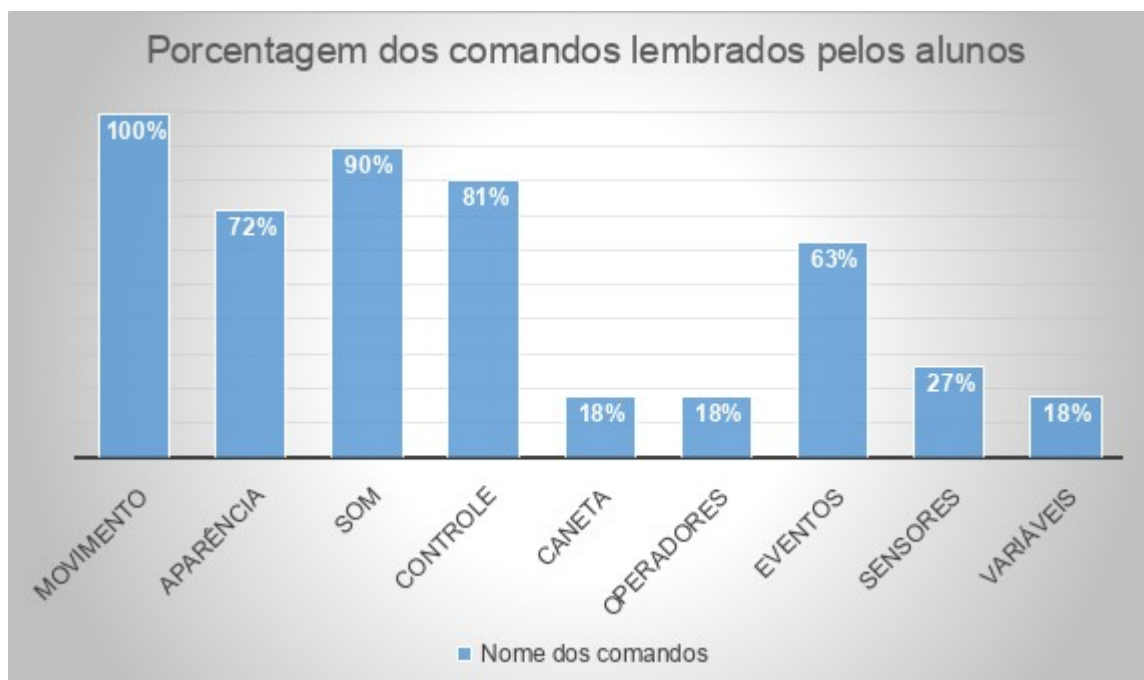


Fonte: da autora

Esse gráfico 11 é referente aos alunos que utilizam a ferramenta *scratch* em suas casas, 67% possuem computadores e utilizam por ele, enquanto 33% usa a ferramenta através de tablet, no questionário 0% citaram outras infraestrutura de *hardware*.

Veremos agora o resultado e a análise da questão 2, onde tratou-se tabular os assuntos do curso que mais as crianças recordam, significativamente é o que mais eles gostaram de aprender e de manipular dentro do ambiente do *scratch*.

Gráfico 12 - Questão 2



Fonte: da autora

O gráfico 12 mostra que 100% dos alunos lembram do comando “movimento”, e analisando o conjunto do gráfico fica claro que os comandos que envolvem mais a parte de *front-end* que são chamados os desenvolvedores da interface de um programa, a parte visual de um projeto de programação de computadores foram os comandos com maiores porcentagens. São eles os comandos “aparência” com 72%, “som” com 90%, “eventos” 63% e movimento com os 100%.

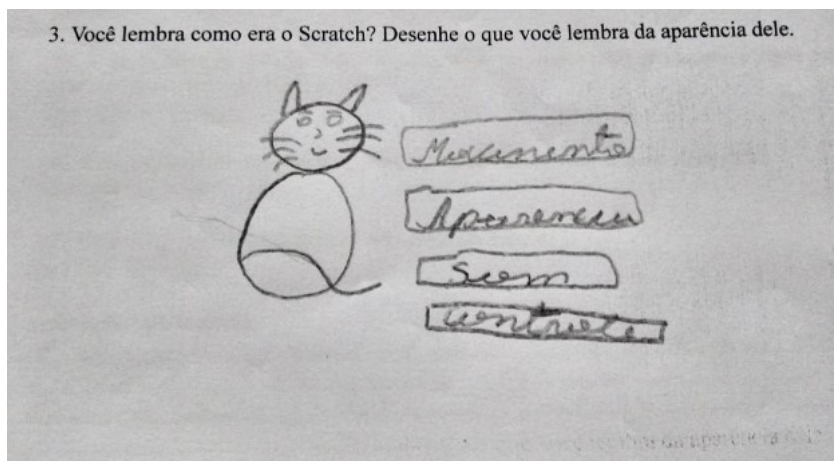
Já dos comandos que envolve a parte lógica, os *back-end* assim chamados desenvolvedores que trabalham com a parte da camada lógica de um programa, a parte não visual, como é o caso dos comandos “variáveis” 18%, “sensores” 27% e “operadores” 18%, no caso do comando operadores foi bem surpreendente a baixa porcentagem, pois é um comando da lógica matemática que os alunos veem nas escolas, ou seja não é uma novidade ou algo que os alunos nunca tenham visto, a expressão desse resultado pode levar a várias hipóteses. Em contraponto a baixa porcentagem do comando operadores, tivemos uma alta de 81% dos alunos que lembraram do comando “controle” um dos comandos lógicos de repetição e de grande importância para a construção de projetos dentro do *scratch*.

O gráfico 12 também mostra que todos os assuntos estudados no curso foram lembrados pelos alunos, mesmo depois de meses após o término do curso.

Na questão (3) Foi pedido para que os alunos desenhasssem o que eles lembravam do ambiente e aparência da ferramenta *Scratch*.

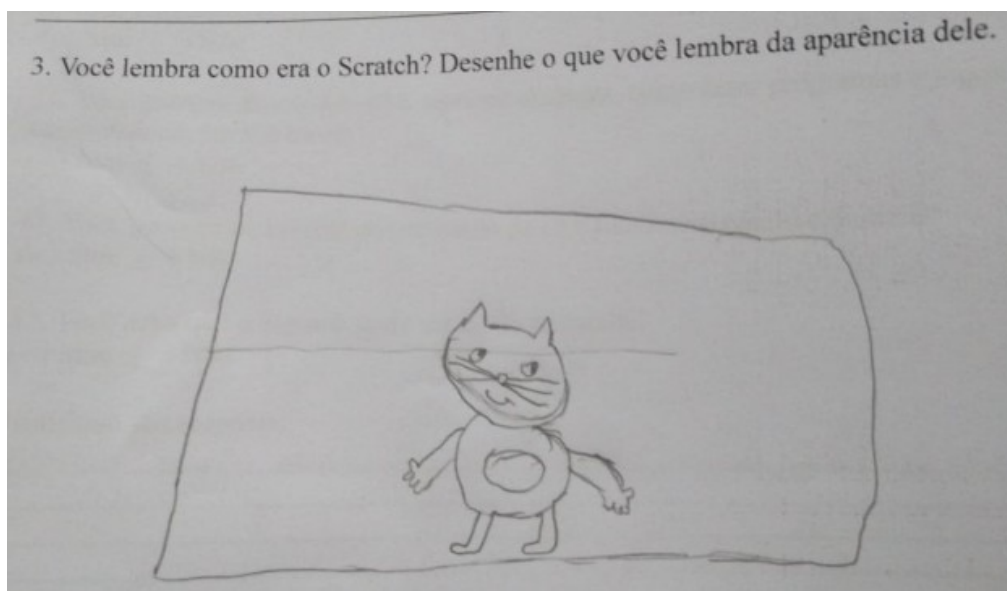
O desenho infantil exprime a afetividade antes mesmo do conhecimento, é para a criança uma expressão de mundo. O desenho representa sua relação com o outro, com o meio, tem uma realidade própria, um ato de significar o quê se apresenta à sua percepção (ANDRADE, 2005).

Figura 4 - Desenho do aluno X.



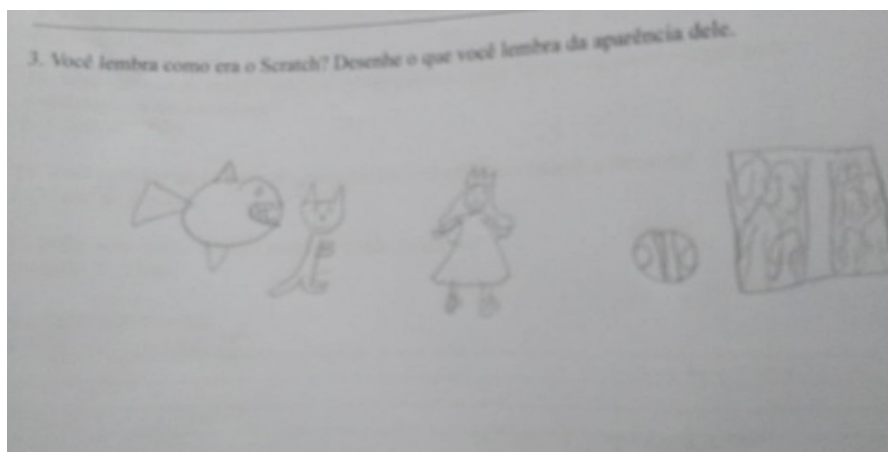
Fonte: da autora

Figura 5 - Desenho do aluno Y.



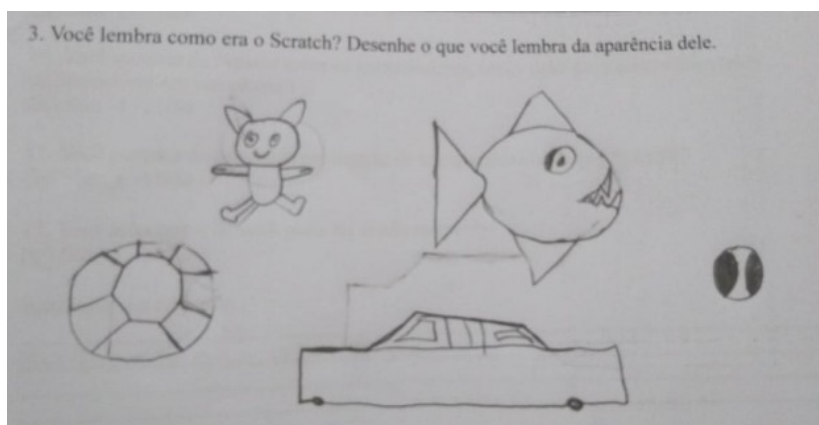
Fonte: da autora

Figura 6 - Desenho do aluno Z.



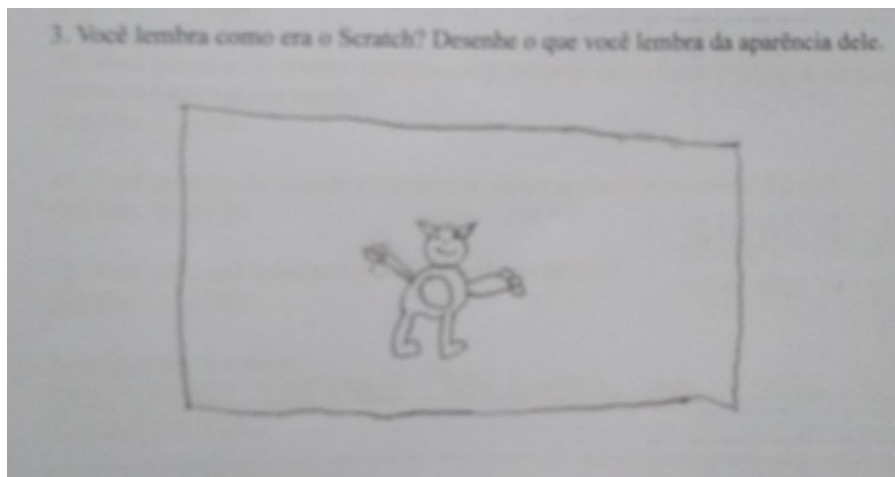
Fonte: da autora

Figura 7 - Desenho do aluno XY.



Fonte: da autora

Figura 8 - Desenho do aluno XYZ.



Fonte: da autora

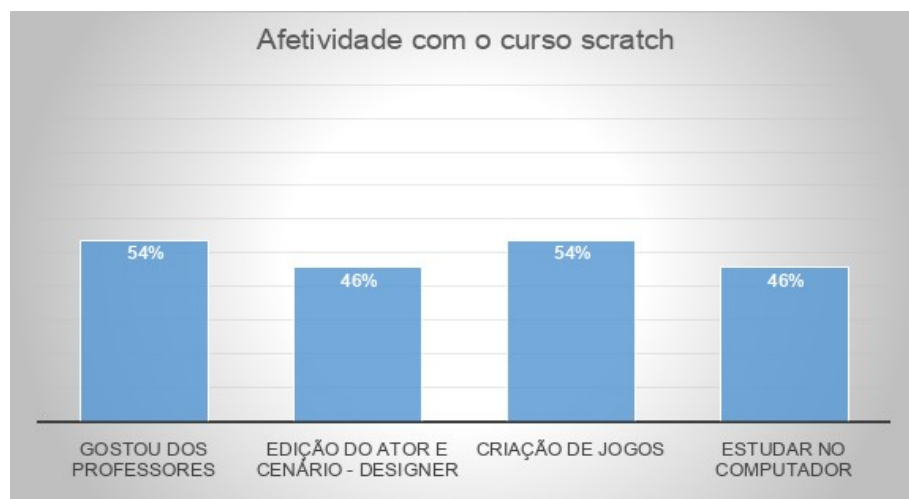
Os desenhos das figuras (4), (5), (6), (7) e (8), notar-se e vai ficando claro conforme cada análise das questões que os alunos têm uma forte memória visual, onde os desenhos seguem essa mesma tendência.

O que ficou em suas memórias ou o que eles acharam mais importante ser lembrado foi, os atores que eles programaram, os cenários, a tela do computador com os personagens dentro, e o gato que é o mascote e personagem principal da ferramenta *scratch*, foi lembrado também os blocos de comandos colocados em ordem um abaixo do outro que são a parte da interface do *scratch* mais coloridas.

O desenho infantil tem relação com o mundo visível e com o outro, com a realidade percebida. {...} através do desenho, a criança tenta explicar o mundo exprime o seu pensamento. (ANDRADE, 2005).

Agora a análise da questão 4 que procurou saber dos alunos na parte sensível, emotiva sobre o curso em todo seu contexto, aluno-professor-aluno-conteúdo.

Gráfico 13 - Questão 4



Fonte: da autora

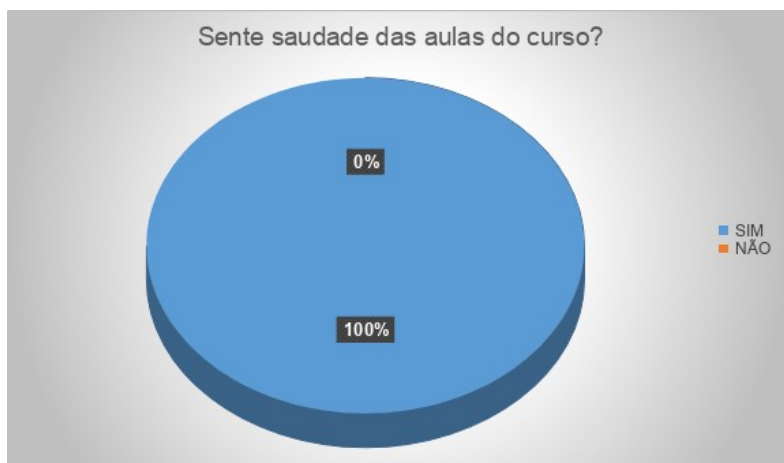
O gráfico 13 reflete o resultado do questionamento afetivo que os alunos tiveram com o curso, lhe foi questionado subjetivamente o que mais lhe agradou no curso. Pela tabulação de respostas chegamos a 4 afirmações de interesse das maiorias.

De 100% dos alunos 54% respondeu que: o que mais gostou do curso foi dos professores, um dos alunos falou “Gostei dos professores porque eles ensinam muito bem”, aluno X. De 100% dos alunos 46% disseram ter gostado de trabalhar com a edição dos atores na criação de cenários, “Gostei dos professores e de pintar o boneco”, aluno Y. De 100% dos alunos 54% diz ter gostado da parte de criação de jogos e de jogar seus jogos, aluno YY diz “fazer jogos e mexer no

computador”. E 46% dos 100% dos alunos afirmam que o que mais gostou no curso do *scratch* foi estudar no computador, aluno XY diz “como controlar e mexer o computador”. Nenhum dos alunos deixaram de responder essa questão.

Ainda nas questões de cunho afetivo, a questão 5 busca saber se as crianças sentem saudade, do curso.

Gráfico 14 - Questão 5

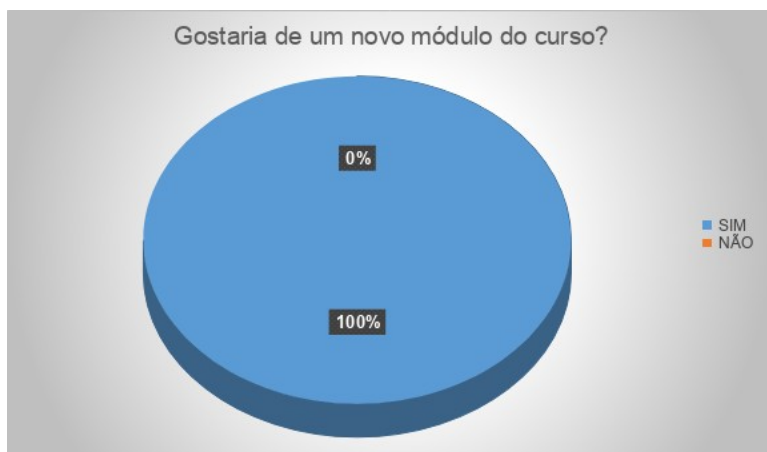


Fonte: da autora

Todos os alunos responderam essa questão e 100% dizem que sente saudade do curso, isso mostra que o curso foi satisfatório para eles. Saber da parte afetiva das crianças quanto a questão do ensino é importante, pois gostar de aprender algo deixa o ensino mais fácil e os resultados esperados pelos educadores se tornam mais concretos e o desenvolvimento da criança flui melhor.

A questão 6 procura saber sobre os interesses das crianças em aprender mais sobre linguagem de programação junto ao *scratch* potencializando o pensamento computacional.

Gráfico 15 - Questão 6



Fonte: da autora

O gráfico 15 tem a porcentagem de 100% refletindo que os alunos querem ampliar seus conhecimentos na temática do curso, pode-se perceber que os sentidos desses alunos já apontam para uma direção, ou seja que suas percepção quanto a linguagem de programação é positiva, mostrando sua curiosidade em aprender sobre.

A questão 7 entra em questões de aprendizagem interdisciplinar, onde buscou-se compreender dos alunos, se os estudos sobre a linguagem de programação *scratch* trouxe resultados positivos em outras disciplinas e no comportamento geral do aluno.

Gráfico 16 - Questão 7

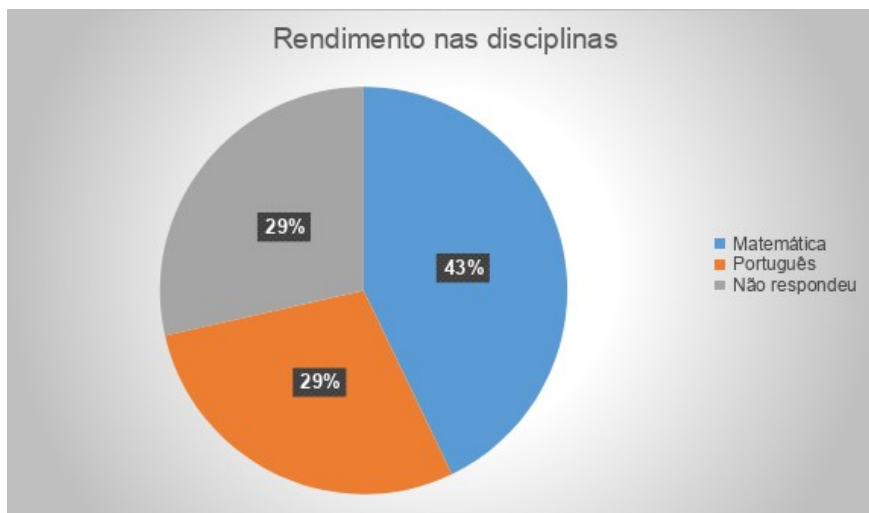


Fonte: da autora

O gráfico 16 mostra que 91% dos alunos disseram que “melhoraram muito”, enquanto 9% disse que “melhorou um pouco”, e ninguém ficou indiferente ao curso “não faz diferença”. Mostrando que em seus sentidos acredita que algo mudou nos aspectos escolares.

Ainda na questão 7 buscou saber se houve uma melhora em alguma disciplina específica.

Gráfico 17 - Questão 7-A



Fonte: da autora

Os alunos que responderam só citaram as disciplinas de matemática com maioria de 43% e Português com 29%. Os outros 29% não citaram nenhuma disciplina apesar do gráfico anterior mostrar que 91% dos alunos disseram que seu rendimento escolar melhorou, porém não especificaram as ou a disciplina. Vale salientar que existem outros aspectos de melhora de rendimento, por exemplo organização, trabalho em equipe são requisitos sociais importantes, que se pode obter ensinando uma linguagem de programação dentro dos muros do raciocínio lógico. O aluno ZY falou que “melhorei muitas nas disciplinas da matéria de matemática”

A questão 8 procura justificativas dos alunos quanto aos que responderam positivamente que o curso trouxe questões positivas em sua vida escolar, então nessa questão buscou-se saber como foi que ocorreu essa contribuição, qual metodologia os ajudou no seu aprendizado na escola.

Gráfico 18 - Questão 8



Fonte: da autora

Analisando o gráfico 18 notamos que 36% dos alunos não souberam responder a questão, isso equivale a aproximadamente metade dos alunos que responderam. Podemos identificar duas hipóteses: falta de interpretação, saber identificar o que realmente a questão queria saber. Segunda hipótese: a questão foi elaborada fora do contexto a nível de interpretação dos alunos, onde é possível que tenha-se usando palavras que confundem ou não deixam claro o que realmente queria que fosse compreendido.

Os outros 63% que são a soma das porcentagens dos outros itens citados pelos alunos, mostram que o objetivo da ferramenta foi minimamente alcançada, aprender a programação faz de você um usuário melhor do computador, como por exemplo: conhecimentos de funções básicas no teclado e aprendizagem para manusear aplicações da *web* e *desktop*, como também aprender coisas novas e despertar a curiosidade do aluno é uma dos pontos de ensino de programação, e saber trabalhar em equipe, tendo respeito nas opiniões dos colegas e prestar muito atenção no trabalho que está realizando. Aluno XYY falou “Eu aprendi que deve respeitar com os professores e que devemos ter responsável em sala de aula e colegas”, aluno ZXX falou “aprendi a mexer no computador mais”, aluno YYZ diz “ficar atento”, já o aluno ZYZ relata “eu faço jogos na aula de computação”.

A questão 9 busca compreender e identificar nos sentidos dos alunos se é importante colocar o computador, as tecnologias digitais como objeto de estudo no contexto do fundamental.

Gráfico 19 - Questão 9

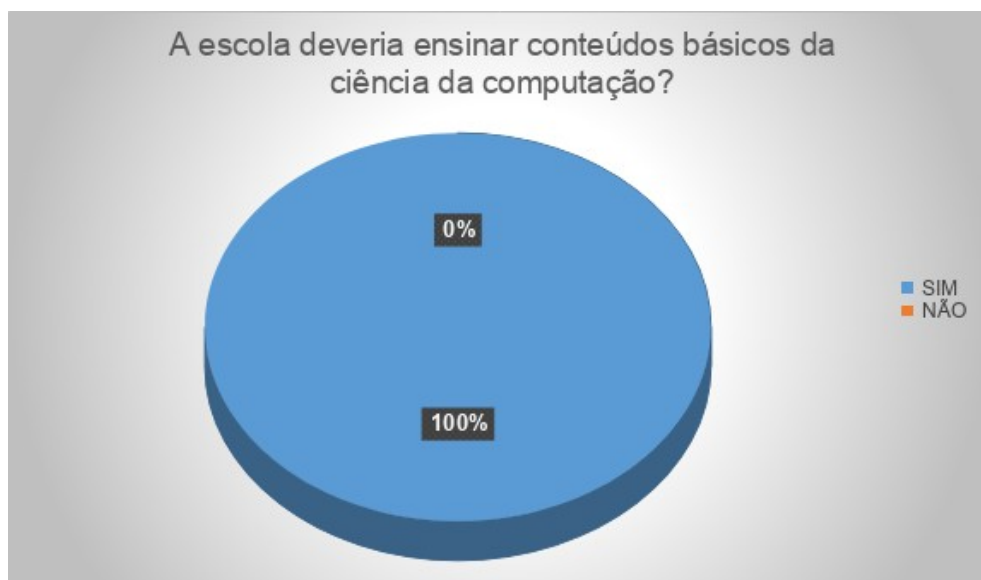


Fonte: da autora

No gráfico 19 todos os alunos responderam, e 100% disseram que acham importante estudar sobre dos computadores. Em suas opiniões acreditam que devemos ter as TICs não só apenas como instrumentos de ensino metodológico mas como objeto de estudo.

A questão 10 permite que as crianças opinem sobre o papel das escolas em compor do seus currículos, acerca dos estudos que envolvam a ciência da computação. Aqui buscou-se saber se os alunos veem a computação como ciência e ter uma ideia de suas percepções sobre o que demos estudar, ou seja, queremos entender o que pensa esses alunos sobre o que a escola deve ou não ensinar, ver a compreensão deles acerca de seus currículos.

Gráfico 20 - Questão 10



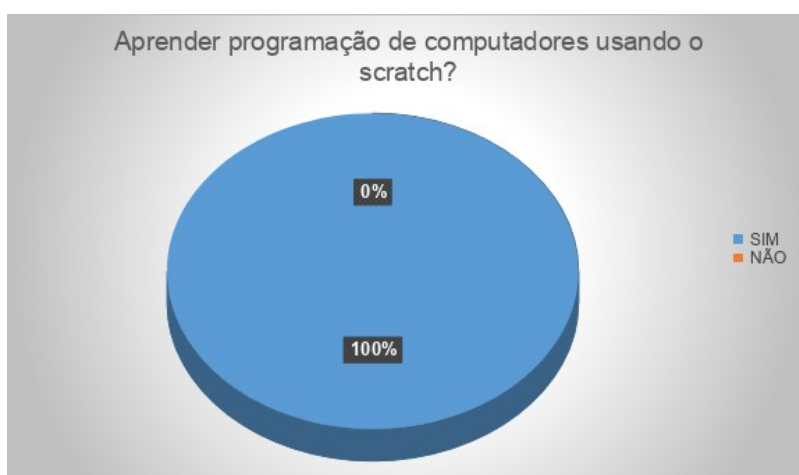
Fonte: da autora

O gráfico 20 representa que as crianças acreditam que a escola deveria ensinar sobre conteúdos da ciência da computação.

Em suma os gráficos 19 e 20 fazem uma leitura sobre os pensamentos das crianças acerca do ensino da ciência da computação no ensino básico. Isso mostra uma visão das crianças sobre o mundo moderno, e sabemos que os nascidos a partir do ano 2000 convivem com a tecnologia, passivamente e ativamente desde sempre, nas diversas áreas da sociedade, e muitos deles sentem falta desse aprendizado sobre tecnologias, computação nas escolas.

A questão 11 procura entender os gostos e as vontades das crianças em aprender programar usando o *scratch*.

Gráfico 21 - Questão 11



Fonte: da autora

O gráfico 21 da ênfase mostrando a afetividade que os alunos tiveram com a ferramenta *Scratch* usado no curso. Esse gráfico mostra que a para eles a ferramenta foi satisfatória para aprender a programar, e que a linguagem *scratch* foi de fácil compreensão, claro tento em vista que os alunos não tiveram contato com outras linguagens de programação, nem outras ferramentas, porém isso é claro, pois o que se quer saber é o que de positivo ou negativo teve a ferramenta em suas opiniões, foi isso que a questão tentou moldar.

A questão 12 busca saber se é possível que a ferramenta possa ser utilizada na escola pelo seus professores.

Gráfico 22 - Questão 12



Fonte: da autora

Os 9% dos alunos que disseram que o *scratch* não pode ser usado, responderam que isso não é possível, O aluno XX disse “não vejo o *scratch* em nenhuma disciplina, mas se tivessem gostaria”. Os 9% do gráfico 22 acreditam que podemos usar o *scratch* para aprender programação, como aponta o gráfico 21, porém para usar na escola eles acham que não seria possível, pelo fato possivelmente a escola não estimular a inserção das tecnologias de programação em sala de aula.

A questão 12 ainda deu-se o trabalho de perguntar aos que responderam, positivamente que o *scratch* deveria ser utilizado na escola. Esse questão busca entender em seus pensamentos o que leva eles a acreditar que a ferramenta poderia esta no cotidiano escolar, em suas aula.

Gráfico 23 - Questão 12-A



Fonte: da autora

No gráfico 23 é correspondente a justificativa dos alunos com a porcentagem de 91% do gráfico 22. Quando foram feitas as tabulações entre as respostas 40% mencionaram que deve se dar a oportunidade de ensino para outras pessoas como diz o aluno XYX “porquê mais pessoas conhece-se mais o scratch” outro aluno afirma que “porquê as pessoas que entrar na escola ai mexer o computador”. E dos 100% dos alunos, também 40% disseram que a ferramenta ajuda a melhorar o desempenho escolar, o aluno ZZZ fala “eu acho que deve ter sim na escola porquê ajuda muito com as atividades”. E 20% não souberam ou não respondeu.

A partir das análises de todos os gráficos, pode-se evidenciar que o ensino da ciência da computação, dentro do ensino de programação com alunos do ensino fundamental do 4º ano, pelos sentidos deles, deveria ser integralizado no currículo escolar, pois além da afetividade e o interesse que tiveram com o conteúdo e a ferramenta, mostra que eles acham importante ser estudado essas tecnologias, que fazem parte do cotidiano deles direta e indiretamente onde estão inseridos e que a escola teria uma melhora na questão do desempenho dos alunos em outras disciplinas, segundo suas perspectivas. Pode-se dizer que os sentidos construídos dos alunos acerca de aprender linguagem de programação com a utilização da ferramenta *scratch* foi positiva e assertiva quanto aos autores que defendem o ensino dessa ciência no ensino básico brasileiro.

Entende-se que por se tratar de algo novo e digital não visto nas aulas tradicionais mas visto no seu cotidiano fora da escola, houve esse grande interesse de ser estudado. Vale também ressaltar que as metodologias junto a esses conteúdos são a base fundamental, se a metodologia empregada não tivesse um resultado positivo nas visões das crianças, a ferramenta juntamente com o conteúdo iria ser julgado de uma forma equivalente a metodologia, então poder-se-ia ter obtido resultado diferente e não positivo quanto a aceitação dos estudos em ciência da computação.

6 CONSIDERAÇÕES FINAIS

O objetivo deste trabalho foi realizar uma análise acerca dos sentidos dos alunos do ensino fundamental sobre a temática do ensino da ciência da computação, com enfoque em lógica da programação e linguagem de programação usando a ferramenta *Scratch*. Como as escolas do ensino público brasileiro ainda não tem em seus currículos o ensino da computação, foi interessante saber de crianças que já tivessem experiência com a temática, por isso o universo da pesquisa utilizado para analisar os sentidos foram os alunos do CEPA, que participaram do projeto de extensão da UFRSA/Angicos.

No capítulo da pesquisa de campo, foram feitas as análises dos gráficos referentes aos questionários que foram aplicados aos alunos que participaram do projeto *scratch* dê vida a sua imaginação.

A conclusão sobre as análises feitas em cima dos dados quantitativos e qualitativos é que : demonstrado que quando o ensino da lógica da programação é inserido no contexto de uma criança cria-se nela uma percepção da realidade onde ela caracteriza esse estudo como sendo algo importante para a vida dela, mostrando interesse e afetividade no ensino dessa nova área que os estabelecimentos escolares público brasileiro não as oferecem. Os sentidos construídos desses estudantes é uma amostra que no contexto urbano nordestino do interior as crianças já demonstram interesse em estudar lógica da programação. Crer-se que privar o conhecimento em assuntos de interesse dos estudantes é tirar as oportunidades de futuros profissionais deste campo, se a computação já faz parte da nossa vida, se já usamos ela para trabalho e lazer, isso significa que é de responsabilidade governamental que se tenha no currículo da educação básica o ensino da ciência da computação.

As pesquisas mostram que a área da computação está em constante crescimento e que futuramente sobrarão vagas de trabalho para essas áreas, pois não se têm ainda uma quantidade expressiva de intelectuais e força de trabalho para ocupar as vagas ociosas no mercado.

Um ponto muito positivo no documento da BNCC é a amplitude do currículo, a flexibilidade de ensino, onde apesar de não termos em suas escritas algo específico para o ensino da computação, o documento deixa claro que o currículo pode ser moldado pelas instituições escolares, podendo aplicar esse ciência como nova forma de ensino, ou extra curriculares.

Portanto o que falta para que ocorra aplicações desses estudos nas escolas? Crê-se que a maior dificuldade seja a infraestrutura de aparatos tecnológicos digitais nas escolas públicas, formação de professores, vagas de empregos para professores de computação nas escola públicas, questões que podem ser debatidos em outros textos.

6.1 TRABALHOS FUTUROS

Este trabalho oferece algumas criações de outros trabalhos envolvendo o desenvolvimento de metodologia de ensino da programação com uso do *scratch* a partir do diagnóstico feito. Possibilidades:

- **Uso da ferramenta *scratch* em uma área do conhecimento**
- **O papel do professor da computação no ensino básico**
- **O papel da escola em moldar um currículo real, atual para seus alunos**

REFERÊNCIAS

ABOUT Scratch. Disponível em: <<https://scratch.mit.edu/about/>>. Acesso em: 15 maio 2019.

ANDRADE, Luci Carlos de. **O DESENHO COMO EXPRESSÃO NO APRENDIZADO INFANTIL: CAMINHOS E POSSIBILIDADES**. 2005. 126 f. Dissertação (Mestrado) - Curso de Educação, Universidade Federal de Mato Grosso do Sul, Campo Grande, 2005. Disponível em: <<http://repositorio.cbc.ufms.br:8080/jspui/bitstream/123456789/761/1/Luci%20Carlos%20de%20Andrade.pdf>>. Acesso em: 25 dez. 2018.

ANTONIO JUNIOR, Wagner. **A influência da informática no desenvolvimento infantil**. 2019. Disponível em: <<https://educador.brasilecola.uol.com.br/trabalho-docente/influencia-informatica-desenvolvimento-infantil.htm>>. Acesso em: 10 jun. 2019.

BASE Nacional Comum Curricular: **COMPETÊNCIAS GERAIS DA EDUCAÇÃO BÁSICA**. **COMPETÊNCIAS GERAIS DA EDUCAÇÃO BÁSICA**. Disponível em: <http://basenacionalcomum.mec.gov.br/images/BNCC_EI_EF_110518_versaofinal_site.pdf>. Acesso em: 29 jul. 2019.

BRACKMANN, Christian Puhlmann; CASSOLA, Nathália. **DESENVOLVIMENTO DO PENSAMENTO COMPUTACIONAL ATRAVÉS DE ATIVIDADES DESPLUGADAS NA EDUCAÇÃO BÁSICA**. 2017. Disponível em: <<http://www.ufrgs.br/secom/ciencia/o-pensamento-computacional-no-ensino-fundamental/>>. Acesso em: 06 abr. 2019.

BREIJS, Cies; MAHFOUF, Anne-Marie; PIACENTINI, Mauricio. Manual do Kturtle. **Acesso em**, v. 16, 2018.

BICALHO, Lucinéia; OLIVEIRA, Marlene de. A teoria e a prática da interdisciplinaridade em Ciência da Informação. **Perspectivas em Ciência da Informação**, Belo Horizonte, v. 16, n. 3, p.47-74, set. 2011. Trimestral. FapUNIFESP (SciELO). <http://dx.doi.org/10.1590/s1413-99362011000300004>. Disponível em: <<http://www.scielo.br/pdf/pci/v16n3/04.pdf>>. Acesso em: 05 nov. 2018.

CASTELLS, Manuel. Prefácio à edição de 2010 de A sociedade em rede. In: CASTELLS, Manuel. **Sociedade em Rede: A era da informação: Economia, sociedade e cultura**. 6. ed. São Paulo: Paz e Terra, 2010. p. 1-30. Tradução de: Marcelo Lino.

EDITORA, Porto. **Lógica in Dicionário infopédia da Língua Portuguesa**. 2003. Disponível em: <<https://www.infopedia.pt/dicionarios/lingua-portuguesa/l%C3%B3gica>>. Acesso em: 09 nov. 2018.

EDITORA, Positivo. **Dicionário Aurélio de Língua Portuguesa**. 5. ed. Rio de Janeiro: Positivo, 2010.

FORBELLONE, André Luiz Villar; EBERSPÄCHER, Henri Frederico. **Lógica de Programação: A construção de algoritmos e estruturas de dados**. 3. ed. São Paulo: Pearson Prentice Hall, 2005.

FREIRE, Victor. **Ainda vale a pena aprender Assembly?** 2009. Disponível em: <<http://www.dsc.ufcg.edu.br/~pet/jornal/outubro2009/materias/carreira.html>>. Acesso em: 02 nov. 2018.

GIL, A. C. **Como Elaborar Projetos de Pesquisa**. 5. ed. São Paulo: Atlas, 2010.

JUNIOR, Carlos Isaias Dos Santos. **Introdução a Lógica de Programação**. Clube de Autores, 2009.

LLC, Spritebox. **Lightbot : Code Hour**. 2016. Disponível em: <<https://play.google.com/store/apps/details?id=com.lightbot.lightbothoc>>. Acesso em: 09 nov. 2018.

LOPES, Roseli de Deus; ZUFFO, Marcelo Knörich. Experiências interdisciplinares em tecnologias imersivas e interativa. **Revista da Sociedade Brasileira de Computação**, Porto Alegre/rs, v. 2, n. 31, p.47-66, Disponível em: <http://www.sbc.org.br/images/flippingbook/computacaobrasil/computa_31/Comp_Brasil_02_2016.pdf>. Acesso em: 22 dez. 2018.

KOSLOSKY, Marco Antônio Neiva. **APRENDIZAGEM BASEADA EM CASOS UM AMBIENTE PARA ENSINO DE LÓGICA DE PROGRAMAÇÃO**. 1999. 113 f. Dissertação (Mestrado) - Curso de Engenharia de Produção., Universidade Federal de Santa Catarina, Florianópolis, 1999. Disponível em: <<https://repositorio.ufsc.br/bitstream/handle/123456789/81012/146842.pdf?sequence=1>>. Acesso em: 28 jan. 2019.

MESTAL. **Programação para crianças**. 2018. Disponível em: <<https://play.google.com/store/apps/details?id=com.mestal.programmingforkids>>. Acesso em: 10 fev. 2019.

MARTINS, Ricardo. **Linguagem assembly**: Conceito de Linguagem Assembly. 2017. Disponível em: <<http://knoow.net/ciencinformtec/informática/linguagem-assembly/>>. Acesso em: 11 set. 2018.

MELLO, Guiomar de. **Diretrizes da BNCC norteiam uso da tecnologia na Educação Básica**. 2018. Redação INOVEDUC. Disponível em: <<https://www.bettbrasileduc.com.br/bettblog/diretrizes-da-bncc-norteiam-uso-da-tecnologia-na-educa%C3%A7%C3%A3o-b%C3%A1sica>>. Acesso em: 23 fev. 2019.

MUSTEREN. **Algorithm City**. Disponível em: <https://play.google.com/store/apps/details?id=air.MusterenGames.AlgorithmCityPro&hl=pt_BR>. Acesso em: 09 nov. 2018.

PAPERT, Seymour. Logo: Computadores e educação. **Brasiliense**, São Paulo, p.1-128, 1985. Disponível em: <<https://www.dropbox.com/s/kjioae2jlyctlib/Logo%20-%20Computadores%20e%20educacao%20v2.pdf?dl=0>>. Acesso em: 18 jun. 2019.

PEREIRA, Miriã; CAMILA, Priscilla; SANTIAGO, Suzy. **Sintaxe e Semântica de Linguagens de Programação**. 2009. Disponível em: <<http://sibetim.blogspot.com/2009/03/sintaxe-e-semantica-de-linguagens-de.html>>. Acesso em: 12 Set. 2018.

RAMOS, Fellipe Oliveira; TEIXEIRA, Lilian da Silva. O Scratch como ferramenta pedagógica para o desenvolvimento do pensamento computacional sob a perspectiva da aprendizagem significativa. In: RAABE, André Luís Alice et al. **Educação Criativa**: Multiplicando experiências para a aprendizagem. 4. ed. Recife: Pipa Comunicação, 2016. Cap. 11. p. 349-356.

ROCHA, Julci. **STEM: Um novo olhar para o ensino de ciências**. 2019. Disponível em: <<https://www.geekie.com.br/blog/stem-ensino-ciencias-matematica/>>. Acesso em: 23 jul. 2019.

SBC, Sociedade Brasileira de Computação (2016). Computação na Base Nacional Comum Curricular. Elaborada pela comissão de Educação da SBC em colaboração com a Comissão Especial de Informática na Educação e membros da SBC.

SILVA, Meg. **FLUXOGRAMA**. 2012. Disponível em: <<http://files.meg-silva9.webnode.com/200000004-66861677f7/FLUXOGRAMA.pdf>>. Acesso em: 26 out. 2018.

SILVA, Geraldo Lúcio da. **O APRENDER A APRENDER NA EDUCAÇÃO TECNOLÓGICA**. 2002. 132 f. Dissertação (Mestrado) - Curso de Engenharia de Produção, Universidade Federal de Santa Catarina, Florianópolis, 2002. Disponível em:

<<https://repositorio.ufsc.br/xmlui/bitstream/handle/123456789/83735/185261.pdf?sequence=1&isAllowed=y>>. Acesso em: 12 nov. 2018.

SOUSA, Bruno Jeferson de; DIAS JÚNIOR, José Jorge Lima; FORMIGA, Andrei de Araújo. **Introdução a Programação: Paradigmas de Programação**. João Pessoa: Editora da Ufpb, 2014.

SOUZA, Marcos A. Furlan de et al. **Algoritmos e Lógica de Programação: Um Texto Introdutório para Engenharia**. 2. ed. São Paulo: Cengage Learning, 2012.

SEBESTA, Robert W. **Conceitos de Linguagens de Programação-11**. Bookman Editora, 2018.

TUCKER, Allen; NOONAN, Robert. **Linguagens de Programação-: Princípios e Paradigmas**. AMGH Editora, 2009.

TEDESCO, Kennedy. **Linguagens e paradigmas de programação**. 2016. Disponível em: <<https://www.treinaweb.com.br/blog/linguagens-e-paradigmas-de-programacao/>>. Acesso em: 04 out. 2018.

VON WANGENHEIM, Christiane Gresse; NUNES, Vinícius Rodrigues; SANTOS, Giovane Daniel dos. Ensino de Computação com SCRATCH no Ensino Fundamental – Um Estudo de Caso. *Revista Brasileira de Informática na Educação, Cidade*, v. 22, n. 3, p.115-115, nov. 2014.

WING, Jeannette. Computational thinking. **Communications Of The Acm**, [s.l.], v. 49, n. 3, p.33-53, 1 mar. 2006. Association for Computing Machinery (ACM).
<http://dx.doi.org/10.1145/1118178.1118215>.

APÊNDICE A – ROTEIRO DE ENTREVISTA

QUESTIONÁRIO

Idade: _____ **Série** _____

Etnia:

() Branco () Preto () Pardo () Amarelo () Indígena

Gênero:

() Feminino () Masculino

Moradia:

() Casa Própria quitada () Casa Própria ainda pagando () Casa alugada

() Casa cedida

1. Você ainda usa o Scratch em casa?

() Sim () Não

Porquê?

Se a resposta anterior for sim, você acessa no seu computador, celular (smartphone) ou tablet?

2. Você lembra de todos os assuntos vistos no curso? Marque com um X aqueles que você lembra:

- () Movimento
- () Aparência
- () Som
- () Controle
- () Comando Caneta
- () Operadores
- () Eventos
- () Sensores
- () Variáveis

Escolha dois (2) assuntos estudados e escreva o que você sabe sobre eles:

3. Você lembra como era o Scratch? Desenhe o que você lembra da aparência dele.

4. O que mais você gostou do curso do Scratch?

5. Você sente saudades das aulas?

() Sim () Não

6. Gostaria que tivesse mais aulas de Scratch?

() Sim () Não

7. Depois do curso você acha que melhorou as suas dificuldades nas aulas ou em alguma das disciplinas e/ou conteúdos que estuda na escola?

() Melhorou um pouco () Melhorou Muito () Não fez diferença

Qual(is) disciplina(s) e/ou conteúdo(s)? _____

8. O que você aprendeu com a ferramenta Scratch que contribuiu para seu aprendizado na escola?

9. Você acha importante estudar sobre os computadores?

(☐) Sim (☐) Não

10. Você gostaria de estudar sobre os computadores, como fazer programas e jogos para computadores em sua escola?

(☐) Sim (☐) Não

11. Você gostaria de estudar programação de computadores usando o Scratch?

(☐) Sim (☐) Não

12. Você acha que o Scratch pode ser usado na escola?

(☐) Sim (☐) Não

Justifique sua resposta:

ANEXO A – Plano de ensino Scratch: dê vida a sua imaginação

Scratch: dê vida a sua imaginação				
O projeto Scratch: dê vida a sua imaginação, tem como perspectiva proporcionar formação de conceitos básicos da computação relacionados à sua prática e programação, desenvolvimento da organização do pensamento computacional de forma interdisciplinar ao conteúdo do currículo cotidiano de alunos do ensino fundamental da E. M. Pedro Amorim, localizada no município de Açú.				
Objetivos de aprendizagem				
Compreender o que é um algoritmo e usar os passos básicos de algoritmos para a resolução de problemas ao projetar soluções. Descrever e analisar uma sequência de instruções a ser seguidas. Colaborar com colegas em grupo para criar, desenvolver, publicar e apresentar projetos, utilizando Scratch. Usar os passos básicos da engenharia de software (análise, projeto, implementação e teste) para criar o projeto de um jogo com tema de história utilizando conceitos básicos de programação (variáveis, operadores, estruturas condicionais e estruturas de repetição).				
Conteúdo				
Conhecer conceitos de computação e programação. Conhecer o Scratch (ambiente e linguagem de programação). Analisar, projetar, implementar, testar e compartilhar um jogo no Scratch, junto com o(a) instrutor(a), ilustrando um tema de história. Analisar, projetar, implementar, testar, compartilhar e apresentar um jogo no Scratch, em grupo.				
Pré requisitos				
Hardware: Computador com monitor, teclado e mouse – no máximo 3 alunos por computador. Projetor multimídia – utilizado para apresentação de slides. Acesso à internet – utilizado para criação, compartilhamento, apresentação de jogos exemplos no site do Scratch. Opcional: Caixa de som/microfone – utilizado para apresentação de vídeos com som/criação de áudios. Software: Navegador Firefox ou Chrome atualizado – utilizado para acessar o ambiente Scratch on-line. Caso não esteja atualizado podem ocorrer problemas de acesso ao ambiente. Sistema Operacional: Linux Educacional (versão 4 ou acima), Linux (qualquer distribuição atualizada de 32 bits) Windows (XP ou acima) e Mac OS (versão 10 ou acima). Opcional: Adobe Air e Scratch off-line instalado nos computadores – utilizado para acessar o ambiente Scratch off-line. Outros: Contas no ambiente Scratch on-line para cada grupo – criadas pelo professor. Giz/Caneta e apagador – utilizado para escrever no quadro.				
Sequenciamento				
Tempo	Tópico	Estratégia	Recursos didáticos	Avaliação
1ª aula				

	Apresentação do projeto.	Apresentação	Slides	
	O que é computação? O que é um algoritmo, para que serve? O que é linguagem de programação?	Apresentação	Slides	
	O que é Scratch? Interface.	Apresentação	Slides	
	Divisão de grupos.	Formação de Lista de grupos grupos pelos anos (máximo 3 pessoas)	Lista de grupos	
	Distribuição de material	-	Slides	
	Introdução ao Scratch	-	Slides	
2ª Aula				
	Explorando a interface: Movimento	Apresentação passo a passo.	Slides	
3ª Aula				
	Explorando a interface: Aparência	Apresentação passo a passo.	Slides	
4ª Aula				
	Explorando a interface: Som	Apresentação passo a passo.	Slides	
5ª Aula				
	Explorando a interface: Comando Controle	Apresentação passo a passo.	Slides	
6ª Aula				

	Explorando a interface: Comando Caneta	Apresentação passo a passo.	Slides	
7ª Aula				
	Explorando a interface: Comando Variáveis	Apresentação passo a passo.	Slides	
8ª Aula				
	Explorando a interface: Comando Eventos	Apresentação passo a passo.	Slides	
9ª Aula				
	Explorando a interface: Comando Sensores	Apresentação passo a passo.	Slides	
10ª Aula				
	Explorando a interface: Comando operadores	Apresentação passo a passo.	Slides	
11ª Aula				
	Criação do jogo: Munda da Imaginação	Apresentação passo a passo.	Slides	
12ª Aula				
	Início dos projetos dos alunos: Criar um projeto (inserir atores do jogo, planos de fundo, inicialização de atores).	-	Comunicação oral	
13ª Aula				
	Continuação do projeto: programar função de atores/pano	-	Comunicação oral	

	de fundo.			
14ª Aula				
	Continuação do projeto: programar função de atores/pano de fundo.	-	Comunicação oral	
15ª Aula				
	Apresentação do projeto pelos alunos.	-	Slides, comunicação oral	
Bibliografia				
1. Iniciativa Computação na Escola. Disponível em: <http://www.computacaonaescola.ufsc.br>. Acesso em: mar. 2016. 2. CSTA. K-12 Computer Science Standards. The CSTA Standards Task Force. CSTA K–12 Computer Science Standards – Revised 2011, ACM, New York/USA, 2011. (Resumo em Português: http://www.computacaonaescola.ufsc.br/wp-content/uploads/2013/09/CurriculoACMIEEE-resumido-PORT_v10.pdf) 3. MIT. Scratch, 2016 Disponível em <http://scratch.mit.edu> Acesso em: mar. 2016 4. PCN, Parâmetros Curriculares Nacionais, Terceiro e Quarto ciclos do Ensino Fundamental. MEC. Disponível em: <http://portal.mec.gov.br/seb/arquivos/pdf/introducao.pdf>. Acesso em: mar. 2016				

APÊNDICE B

Os comandos do *Scratch*

Para utilizar os comandos é necessário arrastar e agrupá-los.

Os blocos de comando eventos e movimento

Os eventos são primeiros comandos a serem usados, todos os comandos são importantes, porém os eventos têm em sua lógica a executar os outros scripts. É ele que dá o pontapé inicial, podemos chamar eles inicializadores.

Quadro 1 – Scripts de eventos

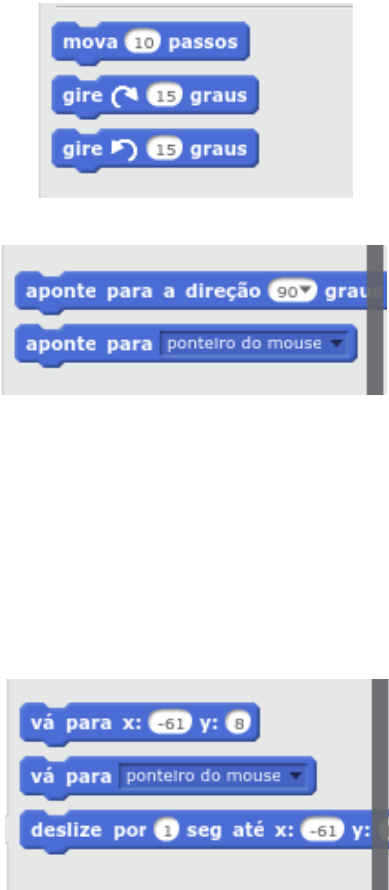
Descrição dos Blocos de comando	
Imagem dos blocos de comandos	Nome e descrição dos blocos de comando
	Quando clicar em (bandeira verde)
	Executa um script após clicar na bandeira verde
	Quando a tecla (qualquer) for pressionada
	Executa um script quando a tecla específica é pressionada
	Quando este ator for clicado
	Executa um script após o ator ser clicado
	Quando o pano de fundo mudar
	Executa um script quando o pano de fundo muda para outro cenário
	Quando (atributo) for maior que (número)
	Executa um script quando o atributo selecionado ultrapassa um valor específico
	Quando receber uma mensagem
	Executa um script quando recebe uma mensagem
	Envie uma mensagem a todos
	Envia mensagens para todos os atores, essa mensagem é acionada para a execução de outro script.

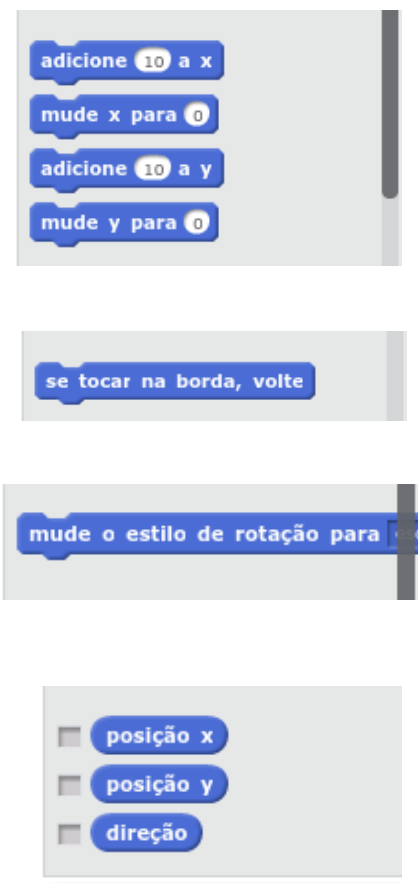
Descrição dos Blocos de comando	
Imagem dos blocos de comandos	Nome e descrição dos blocos de comando
	Envie uma mensagem a todos e espere
	Envia e espera mensagem para todos os atores
	Fim

Fonte: Da autora

Os scripts de movimento estão todos relacionados a execução dos atores do palco.

Quadro 2 – Scripts movimento

Descrição dos Blocos de comando	
Imagem dos blocos de comandos	Nome e descrição dos blocos de comando
	Mova (número) passos
	Faz o ator mover “N” quantidade de passo
	Gire para a direita (número) graus
	Faz o ator girar a direita
	Gire para a esquerda (número) graus
	Faz o ator girar a esquerda
	Aponte para a direção (opção) graus
	Estabelece a direção do ator
	Aponte para (opção)
	Orienta o ator na direção do mouse ou de outro ator
	Vá para eixo X: (número) e eixo Y: (número)
	Vai para uma determinada posição no palco
	Vá para (opção)
	Vai para a localização do mouse ou de outro ator

Descrição dos Blocos de comando	
Imagem dos blocos de comandos	Nome e descrição dos blocos de comando
	(número) e eixo Y: (número)
	Desliza ao longo do tempo até a posição dos eixos
	Adicione (número) a eixo X; mude eixo X para (número); adicione (número) a eixo Y; mude eixo Y para (número)
	Adiciona e muda as coordenadas do ator
	Se tocar na borda, volte
	Esse comando impede o ator sair do palco
	Mude o estilo de rotação para (opção)
	Define o estilo de rotação
	Posição X; posição Y; direção
	Informa a posição e a direção do ator no palco
	Fim

Fonte: Da autora.

Os blocos de comando controle e sensores

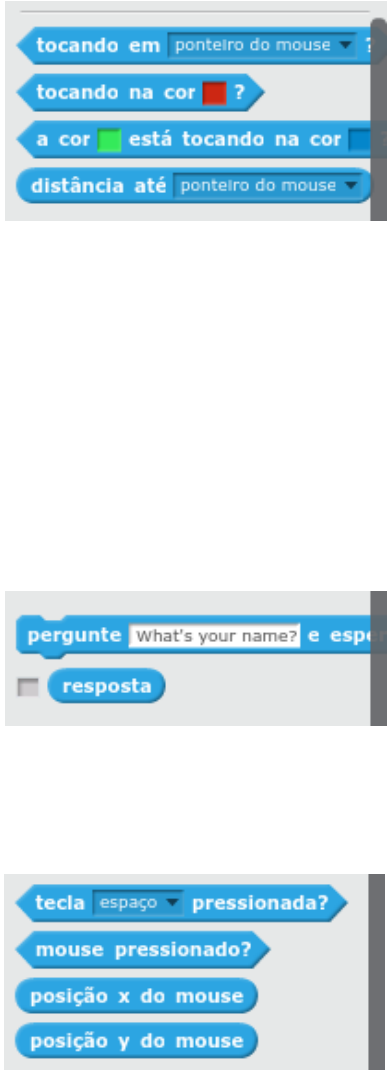
O bloco de comandos controle é responsável pelos laços de repetição. Os laços de repetição são scripts que servem para repetir determinadas vezes um mesmo comando, assim poupando blocos de códigos, diminuindo a quantidade de comando dentro do editor facilitando até mesmo o entendimento dos códigos e a vida de quem está programando seu jogo ou alguma animação dentro da ferramenta.

Descrição dos Blocos de comando	
Imagem dos blocos de comandos	Nome e descrição dos blocos de comando
	Espere (número) segundos
	Espera um tempo específico de segundos para poder executar o próximo bloco
	Repita (número) vezes
	Executa os scripts dentro dele em um determinado número de vezes
	Sempre
	Executa repetidamente e para sempre os comandos dentro dele
	Se (condição) então
	Se a condição for verdadeira, ele executa os comandos no seu interior
	Se (condição) então; senão
	Se a condição for verdadeira, executa os comandos envolvidos pela porção "se" e; se não for, executa os comandos envolvidos pela porção senão
	Espere até (condição)
	Espera até que a condição seja verdadeira, depois executa os comandos que se seguem
	Repita até que (condição)
	Repete os blocos que se seguem até que a condição seja verdadeira
	Pare (opção)
	Interrompe todos os scripts em todos os atores

Fonte: Da autora.

O bloco de comando sensores do *Scratch* é usado como ferramenta de receptor de mensagens. Os sensores do *scratch* depende de um estímulo, ele só irá ser executado caso algo aconteça, diferente do controle que executa algo os sensores dependem da ação de outros blocos bem específico.

Quadro 4 – Scripts sensores

Descrição dos Blocos de comando	
Imagem dos blocos de comandos	Nome e descrição dos blocos de comando
	Tocando em (opção)
	Tem o valor verdade se o ator estiver tocando no ator especificado, na borda ou no cursor do mouse
	Tocando na cor (escolher)
	Tem o valor verdade se o ator está tocando na cor especificada
	A cor (escolher) está tocando na cor (escolher)
	Tem o valor verdade se a primeira cor estiver tocando na segunda cor
	Distância até (opção)
	Informa a distância até o ator especificado ou até o cursor do mouse
	Pergunte (escrever pergunta) e espere; resposta
	Faz uma pergunta na tela e armazena a entrada do teclado no bloco resposta
	Tecla (opção) pressionada?
	Tem o valor verdade se a tecla especificada estiver sendo pressionada
	Mouse pressionado?
	Tem o valor verdade se o botão do mouse estiver pressionado

Continua

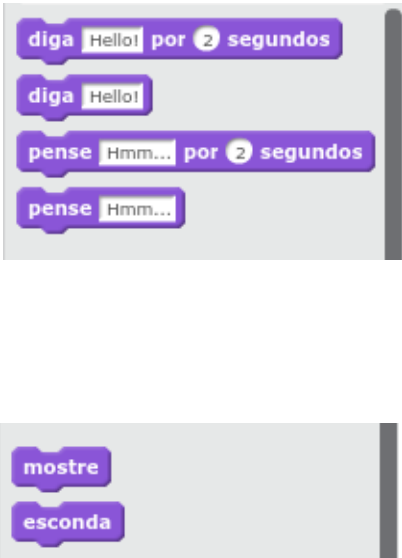
Descrição dos Blocos de comando	
Imagem dos blocos de comandos	Nome e descrição dos blocos de comando
	Posição do eixo X do mouse; posição do eixo Y do mouse
	Tem o valor da coordenada x e y da posição do ponteiro do mouse Fim

Fonte: Da autora.

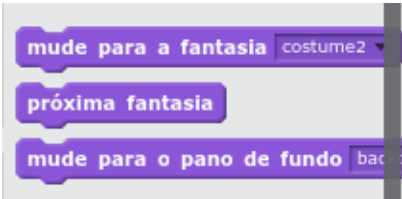
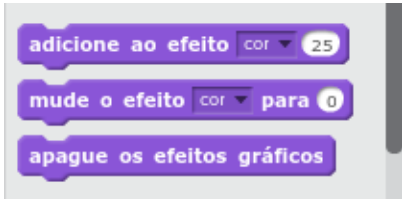
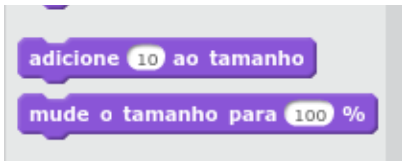
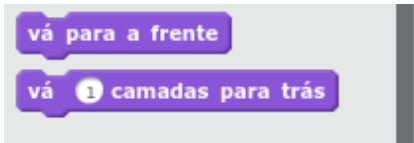
Os blocos de comando aparência e operadores

Os blocos de comandos aparência como o nome já diz trata-se da parte visual do ator e do cenário, ou seja, a interface de interação que o projeto vai ter com o usuário, seja pela construção de um jogo, umas histórias animadas, um quiz de matemática por exemplo ou uma animação.

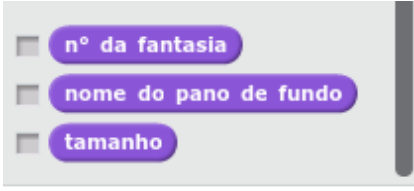
Quadro 5 – Scripts aparência

Descrição dos Blocos de comando	
Imagem dos blocos de comandos	Nome e descrição dos blocos de comando
	Diga (digite) por (número) segundos
	Mostra texto em um balão de fala durante segundos específico
	Diga (digite)
	Mostra texto em um balão de fala
	Pense (digite) por (número) segundos
	Exibe palavras em um balão de pensamento por um número de segundos
	Pense (digite)
	Exibe palavras em um balão de pensamento
	Mostre
	Faz o ator aparecer no palco
	Esconda
	Faz o ator se sumir no palco
	Mude para fantasia (opção)

Continua

Descrição dos Blocos de comando	
Imagem dos blocos de comandos	Nome e descrição dos blocos de comando
   	Muda de fantasia a especificada, para alterar o aspecto (aparência) de um ator Próxima fantasia Muda para a próxima fantasia na lista de fantasias do ator Mude para pano de fundo (escolher) Muda para o palco especificado Adicione ao efeito (opção) (número) Altera o efeito gráfico aplicado a um ator Mude o efeito (opção) para (número) Atribui um número específico ao efeito gráfico aplicado a um ator Apague os efeitos gráficos Limpa todos os efeitos gráficos aplicados a um ator Adicione (número) ao tamanho Altera uma quantidade especificada ao tamanho do ator Mude o tamanho para (número) % Define o tamanho do ator para % (por cento) específico do tamanho original Vá para a frente Desloca o ator para a frente de todos os outros atores Vá (número) camadas para trás

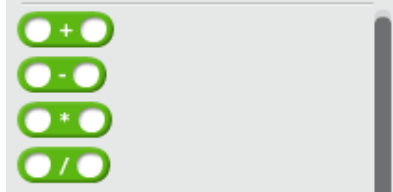
Continua

Descrição dos Blocos de comando	
Imagem dos blocos de comandos	Nome e descrição dos blocos de comando
	Faz o ator recuar um dado número de camadas, de modo a poder ser sobreposto por outros atores
	N° da fantasia
	Mostra na tela o número da fantasia atual do ator
	Nome do pano de fundo
	Informa o nome do pano de fundo atual do palco
	Tamanho Mostra na tela o tamanho do ator, em % do tamanho original Fim

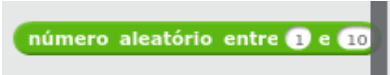
Fonte: Da autora.

Os blocos de comandos de operadores são a parte lógica matemática do Scratch esses comandos são de mais fácil compreensão, pois desde sempre estudamos na matemática.

Quadro 6 – Scripts operadores

Descrição dos Blocos de comando	
Imagem dos blocos de comandos	Nome e descrição dos blocos de comando
	(Número) + (número)
	Realiza a operação matemática da adição com dois números
	(Número) - (número)
	Realiza a operação matemática da subtração com dois números
	(Número) * (número)

Continua

Descrição dos Blocos de comando	
Imagem dos blocos de comandos	Nome e descrição dos blocos de comando
  	Realiza a operação matemática da multiplicação com dois números
	(Número) / (número)
	Realiza a operação matemática da divisão com dois números
	Número aleatório entre (número) e (número)
	Escolhe um valor inteiro aleatório dentro do intervalo especificado
	(Número) < (número)
	Tem o valor de verdade se o primeiro valor for menor que o segundo
	(Número) = (número)
	Tem o valor "verdade" se os dois valores forem iguais
	(Número) > (número)
	Tem o valor "verdade" se o primeiro valor for maior que o segundo
	(Condição A) e (condição B)
	Tem o valor "verdade" se ambas as condições forem verdadeiras
	(Condição A) ou (condição B)
	Tem o valor "verdade" se alguma das condições for "verdade"
	Não (condição)

Continua

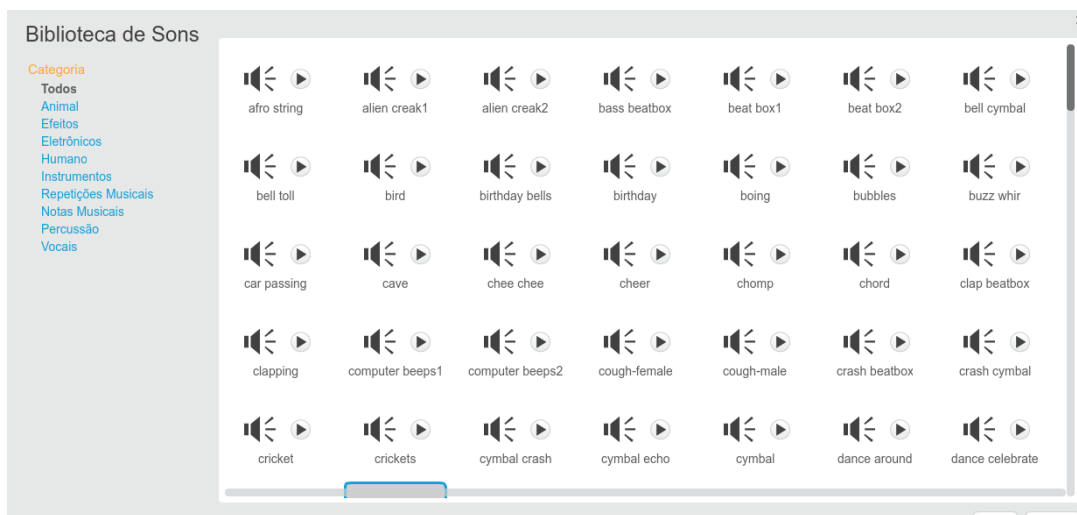
Descrição dos Blocos de comando	
Imagem dos blocos de comandos	Nome e descrição dos blocos de comando
	Tem o valor "verdade" se a condição for "falsa"; tem o valor "falsa" se a condição for "verdade"
	Junte (digite) com (digite)
	Concatena, ou seja, combina, junta cadeias de caracteres
	Letra (número) de (palavra)
	Apresenta o caractere na posição especificada da cadeia de caracteres
	Tamanho de (palavra)
	Informa o número de caracteres em uma cadeia de caracteres
	Resto de (número) por (número)
	Tem o valor do resto da divisão do primeiro número pelo segundo número
	Arredondamento de (número)
	Tem o valor do número inteiro mais próximo de um número dado
	(Função) de (número)
	Calcula uma função
Fim	

Fonte: da autora.

Os blocos de comando Som

Os blocos de script de Som além dos comandos específicos têm variedades de sons. O *Scratch* possui uma biblioteca de sons organizado por categorias.

Figura 1 - Biblioteca de sons



Quadro 7 – Scripts de sons

Descrição dos Blocos de comando	
Imagem dos blocos de comandos	Nome e descrição dos blocos de comando
	Toque o som (opção)
	Reproduz um som
	Toque o som (opção) até o fim
	Reproduz um som e espera ele terminar de tocar
	Pare todos os sons
	Interrompe todos os sons
	Toque o tambor (opção) por (número) batidas
	Toca um som de tambor durante um determinado número de batidas

Continua

Descrição dos Blocos de comando	
Imagem dos blocos de comandos	Nome e descrição dos blocos de comando
	Silêncio por (número) batidas Silencia (não toca nada) durante um número específico de batidas
	Toque a nota (opção) por (número) batidas Toca uma nota musical por um número especificado de batidas
	Use o instrumento (opção) Define o tipo de instrumento que o ator usa para blocos toque a nota
	Adicione (número) ao volume Altera o volume de som do ator em uma quantidade especificada
	Mude o volume para (número) % Atribui um dado valor ao volume de som de um ator
	Volume Relata o volume do som do ator
	Adiciona (número) BPM ao tempo Altera o tempo do ator em quantidade determinada
	Mude o tempo para (número) BPM Atribui ao tempo de um ator um valor dado em batidas por minuto

Continua

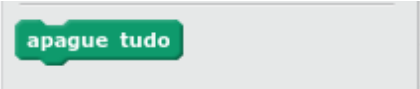
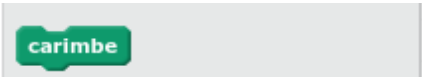
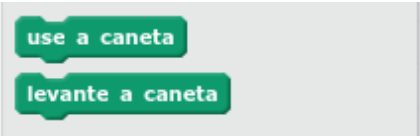
Descrição dos Blocos de comando	
Imagem dos blocos de comandos	Nome e descrição dos blocos de comando
	Tempo
	Relata o tempo de um ator em batidas por minuto Fim

Fonte: da autora.

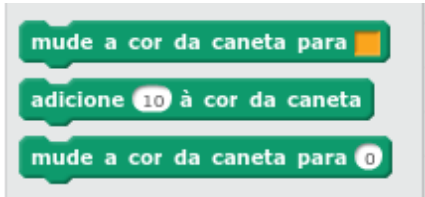
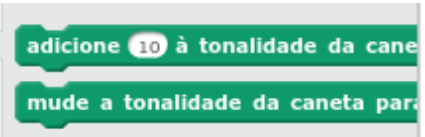
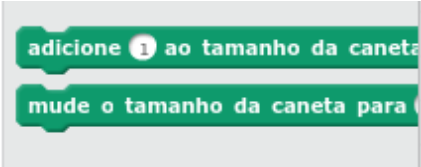
Os blocos de comando Caneta

Os scripts de caneta no *scratch* são os comandos onde você pode desenhar, pintar e escrever no palco.

Quadro 8 – Scripts caneta

Descrição dos Blocos de comando	
Imagem dos blocos de comandos	Nome e descrição dos blocos de comando
  	Apague tudo
	Apaga do palco todas as marcas de caneta e todos os carimbos
	Carimbe
	Carimba no palco a imagem do ator
	Use a caneta
	Usa a caneta do ator, para que ela risque à medida que ele se move
	Levante a caneta
	Levanta a caneta do ator, para que ela não risque à medida que ele se move
	Mude a cor da caneta para (escolher cor)
	Estabelece a cor da caneta com base em uma amostra feita usando a pipeta

Continua

Descrição dos Blocos de comando	
Imagem dos blocos de comandos	Nome e descrição dos blocos de comando
  	Adicione (número) à cor da caneta
	Altera o valor especificado à cor da caneta
	Mude a cor da caneta para (número)
	Estabelece a cor da caneta de acordo com o valor especificado
	Adicione (número) à tonalidade da caneta
	Altera o valor especificado à intensidade da caneta
	Mude a tonalidade da caneta para (número)
	Atribui um dado valor à intensidade da caneta
	Adicione (número) ao tamanho da caneta
	Altera a espessura da caneta.
	Mude o tamanho da caneta para (número)
	Estabelece a espessura da caneta
	Fim

Fonte: da autora.

Os blocos de comando Variáveis

Na paleta de Script temos o bloco de variáveis que aparece vazio, com duas opções criar uma variável e criar uma lista. Então para que apareça a lista de blocos desse script é preciso criar uma variável. Observe a imagem abaixo:

Figura 2 – Menu de criação de variáveis



Fonte: scratch.mit.edu

Ao clicar em criar uma variável, conforme na imagem abaixo vemos que existe três campos para preencher, o primeiro campo é onde irá nomear variável e os outros dois campos abaixo são duas opções as quais deve-se escolher apenas um.

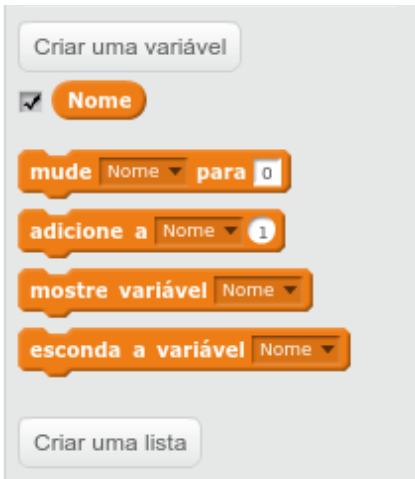
Figura 3 – Tela de nomear e salvar variáveis



Fonte: scratch.mit.edu

Nomeando a variável e escolhendo a opção é só clicar em “ok” e aparecerá os blocos de comando.

Quadro 9 – Scripts variável

Descrição dos Blocos de comando	
Imagem dos blocos de comandos	Nome e descrição dos blocos de comando
	Nome
	Informa o valor da variável
	Mude (opção) para (número)
	Atribui à variável um valor especificado
	Adicione a (opção) (número)
	Altera o valor da variável em um valor especificado
	Mostre variável (opção)
	Mostra o monitor da variável no palco
	Esconda a variável (opção)
	Esconde o monitor da variável de modo que não seja visível no palco.

Fonte: da autora