



UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE TECNOLOGIA



CAROLINA FERNANDES
DOS REIS

ANÁLISE DE MÉTODOS DE INTRODUÇÃO
À LÓGICA DE PROGRAMAÇÃO PARA
CRIANÇAS

Limeira
2019



UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE TECNOLOGIA



CAROLINA FERNANDES
DOS REIS

ANÁLISE DE MÉTODOS DE INTRODUÇÃO
À LÓGICA DE PROGRAMAÇÃO PARA
CRIANÇAS

Trabalho de Conclusão de Curso apresentado
como requisito parcial para a obtenção do título de
Bacharel em Sistemas de Informação à Faculdade de
Tecnologia da Universidade Estadual de Campinas.

Orientador: Prof. Dr. Marcos Augusto Francisco Borges

Limeira
2019



UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE TECNOLOGIA



Folha de Aprovação

Trabalho de Monografia da aluna Carolina Fernandes dos Reis apresentado na Faculdade de Tecnologia – FT, da Universidade Estadual de Campinas – UNICAMP, em 04 de Novembro de 2019 e aprovado pela Banca Examinadora composta pelos professores:

Prof. Dr. Marcos Augusto Francisco Borges

FT/UNICAMP

Prof. Dr. Celmar Guimarães da Silva

FT/UNICAMP

Prof. Dr. Plínio Roberto Souza Vilela

FT/UNICAMP

AGRADECIMENTOS

Primeiramente, agradeço a Deus, pela sabedoria e auxílio constante para enfrentar cada dificuldade encontrada ao longo desta etapa final de curso. Agradeço a Ele por ter me encorajado e me concedido forças para dedicação na concretização deste trabalho.

Aos meus pais, por todo o apoio concedido nesses anos de graduação, foram dias complicados de muito estudo, dedicação e perseverança, nos quais sempre me apoiaram.

Agradeço demais ao meu orientador Professor Dr. Marcos Augusto Francisco Borges, pelo conhecimento compartilhado, pelo aprendizado, pela oportunidade no desenvolvimento desta monografia e, principalmente, por acreditar neste projeto.

Aos meus colegas de curso, amigos, pessoal do trabalho que, de forma direta ou indireta, estiveram comigo durante essa caminhada e auxiliaram para que eu concretizasse mais este sonho de finalização de um curso de graduação.

Agradeço, também, aos alunos da disciplina de Gestão de Projetos das turmas A e B do primeiro semestre de 2019, os quais me ajudaram muito na obtenção de informações para o melhor desenvolvimento desta monografia.

Às equipes da Faculdade de Tecnologia e do LIAG UNICAMP/Limeira pelo auxílio e disponibilidade de recursos.

RESUMO

Diante do cenário atual da educação brasileira e motivado por fatores desafiadores ligados as mudanças na educação, este trabalho apresenta exemplos de projetos que buscam inserir a computação nas escolas de maneira diferenciada, auxiliando os alunos no desenvolvimento de seu próprio aprendizado como protagonistas.

Estes projetos acabam sendo uma oportunidade para apoiar as crianças no aprendizado de conceitos que podem ser muito importantes para o futuro delas, como o Pensamento Computacional. Através da abordagem de Aprendizagem Criativa, que visa ensiná-las de maneira lúdica e por meio de seus interesses, é possível notar a importância desse tipo de ensino diferenciado da maneira tradicional. Esta maneira diferente pode acontecer através de atividades lúdicas e práticas, por meio de ferramentas educacionais como o *Scratch* ou o site *Code.org*.

Além de apresentar os projetos realizados na área de ensino de lógica de programação para crianças e apresentar conceitos importantes para a formação das crianças, o foco deste trabalho foram os métodos que os alunos da Unicamp vêm utilizando em ações de promoção do Pensamento Computacional conduzidas há alguns anos pelo LIAG. O objetivo principal deste trabalho foi realizar uma análise dos métodos utilizados nesses projetos, buscando contribuir na atualização, aprimoramento e desenvolvimento dos mesmos para que sejam cada vez mais eficientes e eficazes na construção desse tipo de conhecimento.

Buscou-se, com isso, apresentar a importância do tema não apenas para os dias atuais, mas também para o futuro, pois a revisão e um ciclo de melhoria contínua em métodos de ensino de programação é um fator relevante para aumentar a qualidade dos métodos de ensino, refletindo cada vez mais em uma melhor experiência tanto para as crianças que participam quanto aos discentes que levam este tipo de projeto as escolas.

Palavras-chave: Jogos Educacionais, *Scratch*, *Code.org*, Pensamento Computacional, *STEM*.

ABSTRACT

Given the current scenario of Brazilian education and motivated by challenging factors, like changes in education, this monograph, presents examples of projects that aiming inserts the computation in schools by a different way, helping students in the development of their own learning as protagonists.

These projects are an opportunity to support the children to learn concepts that can be very important for their future, such as Computational Thinking. Through the Creative Learning that focuses on teach in a playful way and based on their interests, it is possible to see its importance and its difference from the traditional way of teaching, that we are used to. This different way can happen through playful and practical activities, through educational tools like Scratch or Code.org.

Besides presenting the projects carried out in the area of the teaching of programming logic for children and presenting important concepts for the formation of children, the focus of this work was on the methods that Unicamp students use in promoting Computational Thinking activities in the last years by LIAG. The main objective of this monograph is to perform an analysis of the methods that are used in the projects, aiming their update, improvement and development, looking for being more efficient and allow the construction of this kind of knowledge.

This work aims to present the importance of the theme not only for the actually days, but also for the future, because a review and a continuous improvement cycle in the mtehdods of programming teaching are a relevant factor to increase the quality of them. This could result on a better experience for children who participate and for the students who lead this type of project at the schools.

Key words: Educational Games, *Scratch*, *Code.org*, Computational Thinking, *STEM*.

SUMÁRIO

1.	INTRODUÇÃO.....	8
2.	FUNDAMENTAÇÃO TEÓRICA	11
2.1	Instrucionismo, Construtivismo e Construcionismo.....	11
2.2	<i>STEM</i>	11
2.3	Pensamento Computacional.....	12
2.4	Lógica de Programação	14
3.	FERRAMENTAS, MÉTODOS E PRÁTICAS.....	16
3.1	Softwares Educacionais	16
3.1.1	<i>Code.org</i>	16
3.1.2	<i>Scratch</i>	17
3.2	Métodos	18
3.2.1	– “Era uma vez”	18
3.2.2	– Outros métodos	20
3.3	Experiências práticas com Pensamento Computacional.....	21
4.	MÉTODOS E MATERIAIS.....	24
5.	RESULTADOS E DISCUSSÕES.....	26
6.	CONCLUSÃO.....	34
	REFERÊNCIAS	36
	APÊNDICES	40
	Apêndice I – Questionário	40
	Apêndice II – Compilado de respostas ao questionário.....	41
	Apêndice III – Tabulação dos resultados.....	51

1. INTRODUÇÃO

De acordo com Valente (1999), a introdução do uso de computadores na educação brasileira tem se dado de maneira apática. Fatores como a infraestrutura inadequada, principalmente de escolas públicas do país, problemas com a conexão de rede e falta de investimentos acabam por tornar este processo ainda mais complicado e lento.

Segundo Bonilla (2009), as escolas públicas brasileiras enfrentam muitas dificuldades, não apenas de ordem estrutural, mas também pedagógica e tecnológica. Poucos alunos têm acesso aos computadores em suas escolas e a proporção de professores que incentivam o uso destes equipamentos nas aulas é ainda menor. Quando os computadores são utilizados, geralmente, é para realizar atividades como “pesquisas na Internet”, que não mudam as dinâmicas tradicionais de ensino já instituídas pelas escolas e não trabalham com conteúdos diferenciados.

Contudo, Valente (1999) cita que a informática deve estar cada vez mais presente na educação não apenas no uso de computadores para pesquisas e de maneira a simplesmente transmitir informações aos alunos, mas sim de maneira a desafiá-los de forma diversificada no processo de construção de seu próprio conhecimento. Para efetivamente transformar a escola em um lugar de inclusão digital, Bonilla (2009) também considera que não basta apenas o acesso básico, é necessário também o investimento na democratização do uso e na formação e/ou qualificação de mais profissionais, em especial de professores.

Em um mundo em que a informática está cada vez mais presente, um cenário considerado ideal seria que a alfabetização em conceitos de computação fosse introduzida às crianças no ensino fundamental para familiarizá-las com estes conceitos, dando-lhes um nível básico acerca de resolução de problemas e uma mínima proficiência em computação (OLIVEIRA, 2017).

A esse respeito é preciso considerar que:

Quando o computador transmite informação para o aluno, o computador assume o papel de máquina de ensinar, e a abordagem pedagógica é a instrução auxiliada por ele. Essa abordagem tem suas raízes nos métodos tradicionais de ensino, porém, em vez da folha de instrução ou do livro de instrução, é usado o computador. Os softwares que implementam essa abordagem são os tutoriais e os de exercício-e-prática. Quando o aluno usa o computador para construir o seu conhecimento, o computador passa a ser uma máquina para ser ensinada, propiciando condições para o aluno descrever a resolução de problemas, usando linguagens de programação, refletir sobre os resultados obtidos e depurar suas ideias por intermédio da busca de novos conteúdos e novas estratégias. Nesse caso, os softwares utilizados podem ser os softwares abertos de uso geral, como as linguagens de programação, sistemas de autoria de multimídia, ou aplicativos como processadores de texto, *softwares* para

criação e manutenção de banco de dados. Em todos esses casos, o aluno usa o computador para resolver problemas ou realizar tarefas como desenhar, escrever, calcular etc. A construção do conhecimento advém do fato de o aluno ter de buscar novos conteúdos e estratégias para incrementar o nível de conhecimento que já dispõe sobre o assunto que está sendo tratado via computador (VALENTE, 1999, p.12).

Implantar mudanças na escola, adequando-a às exigências da sociedade do conhecimento, constitui um dos maiores desafios educacionais (Hargreaves, 1995 *apud* Valente, 1999, p.39).

Para Diniz (2001), a falta de planejamento pedagógico faz com que hoje as escolas brasileiras, de modo geral, preencham salas com computadores – com o nome de “laboratório de informática” – contratem um especialista em informática para administrar o laboratório e nada mais relacionado a isso. O ponto mais importante é deixado de lado: o professor de sala de aula (o especialista em alguma disciplina) continua em sala de aula tradicional, sem saber utilizar essa nova ferramenta a favor do desenvolvimento do aprendizado de seus alunos.

Para reverter esta realidade atual, o processo de construção das mudanças necessárias a serem inseridas nas escolas abrange: resgate do espaço da escola como ambiente educativo; sala de aula com novas experiências de ensino-aprendizagem e nova metodologia; currículo adaptado às necessidades e características dos alunos e do contexto social; novo papel do professor; engajamento do aluno; gestão escolar; contribuição da comunidade e dos pais; auxílio de especialistas externos; a introdução a novas tecnologias (VALENTE, 1999).

Valente (1999) ainda aborda que as mudanças pedagógicas necessárias são complicadas e os desafios enormes. Estas alterações visam um novo processo educacional, com o objetivo de criar ambientes de aprendizagem nos quais os discentes possam desenvolver capacidades como pensamento crítico, trabalhar em grupo e conhecer os próprios potenciais. Desse modo, os discentes poderão adquirir habilidades e valores necessários para sobreviver na sociedade do conhecimento, como parte de algo que ele construiu e não porque foram transmitidas pelo professor.

Motivado por todos estes fatores desafiadores presentes no cenário atual da educação brasileira, este trabalho apresentará exemplos de projetos que buscam inserir mudanças (ainda que pequenas) no modo de ensino, principalmente em escolas públicas brasileiras. Os projetos apresentados tentam inserir a computação nas escolas de maneira diferenciada, auxiliando os alunos no desenvolvimento de seu próprio aprendizado como protagonistas, visando assim, abordar a importância deste tipo de ação nos dias atuais e enfatizar a relação universidade e comunidade que tem se mostrado acertado.

Esta monografia foi desenvolvida no contexto do curso de graduação em Sistemas de

Informação da Faculdade de Tecnologia da Unicamp - Campus Limeira, na linha de pesquisa de Informática na Educação do LIAG (Laboratório de Informática, Aprendizagem e Gestão). Seu objetivo, além de apresentar os projetos realizados na área de ensino de lógica de programação para crianças, é também realizar uma análise acerca dos métodos que são utilizados nestes projetos, buscando contribuir na atualização, aprimoramento e desenvolvimento dos mesmos para que sejam cada vez mais eficientes e eficazes.

Este trabalho divide-se em seis capítulos. Este primeiro destina-se a introdução do tema, contextualizando o cenário atual da educação brasileira e abordando causas e justificativas para escolha do tema, bem como seu objetivo. No Capítulo 2, encontram-se os referenciais teóricos, sendo apresentados alguns conceitos importantes para o desenvolvimento deste trabalho. No Capítulo 3 é discutida a importância da utilização de *softwares* educacionais para introdução do ensino de lógica de programação para crianças, e alguns métodos de ensino de programação para crianças são abordados. Em Métodos e Materiais (Capítulo 4), são apresentadas as etapas e o planejamento das atividades práticas realizadas para coleta de dados. O Capítulo 5 apresenta as discussões e análises de dados relativos às práticas acompanhadas. Por fim, em Conclusão (6), são relatadas as considerações finais, sugerindo também pesquisas a serem realizadas futuramente.

2. FUNDAMENTAÇÃO TEÓRICA

É importante a realização de estudos teóricos a fim de esclarecer conceitos importantes utilizados no desenvolvimento deste trabalho. As seções deste capítulo abordam de modo geral os conceitos de Instrucionismo, Construtivismo e Construcionismo (Seção 2.1), *STEM* (2.2), Pensamento Computacional, Computação Criativa e Aprendizagem Criativa (2.3) e, por fim, é discutido Lógica de Programação.

2.1 Instrucionismo, Construtivismo e Construcionismo

De acordo com Costa (2010), o Instrucionismo, baseado na teoria comportamental de Skinner, é a forma de apenas transmitir conhecimento através do computador. No Instrucionismo, o aluno é um espectador que recebe um pacote de informações previamente programadas. Esta visão está alinhada utilização dos computadores da maneira tradicional, apenas para realização de pesquisas previamente determinadas pelo professor (Bonilla, 2009).

Em uma visão diferente do Instrucionismo há o Construtivismo e o Construcionismo. Para diferenciá-los, Valente (2019) indica que para o construtivismo (*piagetiano*), o conhecimento é construído na cabeça do aluno como tijolos sobrepostos. Em contrapartida, Papert (1986) *apud* Valente (2019), indica que o Construcionismo é uma abordagem na qual o aluno constrói seu próprio conhecimento através do computador.

Na noção de construcionismo de Papert existem duas ideias que contribuem para que esse tipo de construção do conhecimento seja diferente do construtivismo de Piaget. Primeiro, o aprendiz constrói alguma coisa, ou seja, é o aprendizado por meio do fazer, do "colocar a mão na massa". Segundo, o fato de o aprendiz estar construindo algo do seu interesse e para o qual ele está bastante motivado. O envolvimento afetivo torna a aprendizagem mais significativa (VALENTE, 2019).

2.2 STEM

O termo *STEM* é definido no glossário como um acrônimo, em inglês, utilizado para designar o campo do conhecimento composto por ciências, tecnologia, engenharia e matemática (*Science, Technology, Engineering, and Mathematics*). O termo pode conter variações como por exemplo: *STEAM* que além dos campos do conhecimento acima, ainda inclui artes.

O conceito de *STEM education* é uma proposta inovadora de ensino de ciências. De acordo com Pugliese (2017), na comparação do Brasil com diversos outros países, por aqui, o conceito de *STEM education* ainda é tímido. Em várias revistas brasileiras especializadas no

ensino de ciências, dificilmente são encontradas publicações relativas a este conceito. Na mídia é possível encontrar pequenas reportagens introdutórias, porém apenas mostrando *STEM education* como uma proposta inovadora em alta no exterior, sem um aprofundamento metodológico ou debates sobre princípios pedagógicos.

Pugliese (2017) ainda cita que o desenvolvimento de *STEM education* no Brasil ainda é incerto. Para o autor, é possível observar que há uma incorporação incipiente dessa tendência mais concentrada em grupos da rede privada de ensino, em escolas privadas bilíngues, escolas de programação ou robótica para crianças e em iniciativas isoladas em escolas públicas e organizações educacionais.

Hoje todas crianças deveriam ser letradas em *STEM* para atender as demandas do século. Contudo, quando se trata de uma reforma curricular e metodológica do ensino de ciências, essa mudança fica delimitada e complicada. É necessário reconhecer os pontos positivos nas propostas *STEM* em termos de melhorias no ensino das ciências para direcionar o movimento objetivando uma reforma positiva. Além disso, é importante acompanhar e dimensionar programas *STEM education* afim de melhorar cada vez mais a introdução deste conceito no país, da forma mais eficiente e eficaz possível (PUGLIESE, 2017).

2.3 Pensamento Computacional

De acordo com Wing (2006), o Pensamento Computacional (do inglês, *Computational Thinking*) é uma habilidade fundamental a todos, é pensar recursivamente, é planejar, aprender. O conceito está diretamente ligado a solução de problemas e design de sistemas. Wing (2006) *apud* Rodriguez *et al.* (2015) denomina Pensamento Computacional como um processo de raciocínio lógico, que inclui características como: formulação de problemas; organização e análise lógica dos dados; representação por meio de abstrações; soluções automatizadas por algoritmos; identificação, análise e implementação de soluções; e generalização e transferência do processo de solução encontrado para resolução de outros problemas.

Wing (2008) *apud* França (2013) aborda que o Pensamento Computacional é um pensamento analítico e compartilha com a Matemática a resolução de problemas, com a engenharia a concepção de um sistema que opera no limite do mundo real e com a Ciência a compreensão sobre a inteligência e a mente humana. É possível observar que o conceito tem relação com várias outras áreas do conhecimento, assim como o conceito de *STEM*.

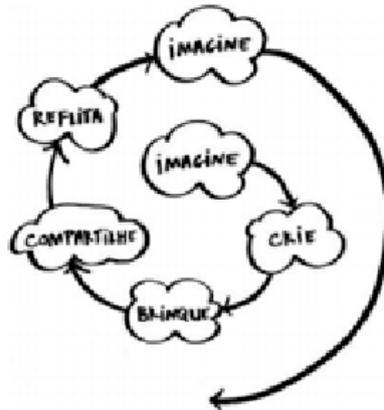
A habilidade de criar programas pode trazer benefícios importantes. Por exemplo, expande a capacidade de aprendizagem, criação e expressão, auxiliando na solução de

problemas e em soluções estratégicas. Programar proporciona a oportunidade de refletir sobre seu próprio pensamento (RESNICK *et al.*, 2009).

Visto a importância do Pensamento Computacional aos interesses e valores de jovens e crianças, destaca-se também a visão da Computação Criativa, a qual o MIT (*Massachusetts Institute of Technology*) (2011) relaciona com as ligações pessoais com a computação baseando-se na criatividade, imaginação e interesses. Os jovens muitas vezes participam como consumidores de tecnologia, porém, alinhados com o Construcionismo, seria interessante tê-los como projetistas / criadores. A Computação Criativa enfatiza o conhecimento e a prática para criar meios computacionais interativos para a aprendizagem.

A Aprendizagem Criativa propõe um processo de aprender centrado no aprendiz, tendo como referência seus interesses. Resnick (2017) ilustra o conceito com um exemplo clássico em um jardim de infância, quando um grupo de crianças brincando no chão com blocos de madeira tenta construir um castelo inspirado na história que a professora lhes contou. Ao tentar empilhar os blocos, a torre se inclina e cai e as crianças buscam uma forma de construção mais estável. Enquanto constroem esse conhecimento, outras crianças contam histórias de moradores deste castelo. Com este exemplo, Resnick (2017) aborda o processo criativo de maneira Espiral. Através das brincadeiras com blocos, construção de castelos e desenvolvimento de uma história, as crianças se envolvem com aspectos criativos, conforme Figura 1. “A espiral da aprendizagem criativa é o motor do pensamento criativo.” Imaginar um castelo e quem vive nele, brincando, interagindo e compartilhando experiências. Quando a torre cai há uma reflexão do porquê e com base nas experiências que passam pelo espiral, as crianças tem novas ideias, imaginam e reconstróem de outras maneiras. Resnick (2017) destaca ainda quatro princípios orientadores da Aprendizagem Criativa (denominado como quatro Ps): Projetos (*projects*), Parcerias (*peers*), Paixão (*passion*) e Pensar brincando (*play*). Os quatro Ps estão presentes no exemplo, pois enquanto desenvolvem o *projeto* de construir a torre, em grupo (*parceria*), de forma motivadora (*paixão*), as crianças estão aprendendo em um processo em que estão *brincando*.

Figura 1: Processo Criativo



Fonte: Resnick (2017).

2.4 Lógica de Programação

Segundo Xavier (2007), lógica de programação é o conjunto de regras utilizadas para criar programas. Através delas, é possível se comunicar com o computador, fazendo com que o dispositivo compreenda e execute o que lhe é determinado. Há três tipos de linguagens de programação: Linguagem de máquina (instruções em binário, em componentes eletrônicos ligados ou desligados, representados por 0 e 1), linguagem de baixo nível (instruções próximas à compreensão da máquina que exigem grande conhecimento de hardware) e linguagem de alto nível (instruções mais próximas da compreensão humana). Neste trabalho, o tipo abordado será o de alto nível.

De acordo com Medina (2005), é possível definir um algoritmo como sendo um procedimento passo a passo para a solução de um problema. Ou ainda como uma sequência detalhada de instruções a serem executadas pelo computador para realização de determinada tarefa. Um exemplo clássico de algoritmo é uma receita culinária. Conforme abordado por Raabe, Zanchett e Vahldick (2015), o primeiro contato de uma pessoa com programação pode ser determinante na forma como perceberão os desafios e enfrentarão dificuldades referentes ao aprendizado de lógica de programação.

Há vários problemas ligados ao ensino de algoritmos e programação que acabam levando a uma desmotivação do aprendiz. Rodrigues (2002) *apud* Klemann *et al.* (2006) destaca como problemas a dificuldade de adaptação, uma vez que estudantes estão acostumados a decorar conteúdos, e a falta de motivação, acreditando ser um obstáculo extremamente difícil de ser superado.

Costa *et al.* (2016) indicam que a inclusão digital deve ocorrer sem distinção de público ou idade. Para eles, seria importante iniciar esta inclusão nos primeiros anos de formação escolar. Sica (2011) *apud* Costa (2016) aborda que o Pensamento Computacional e o raciocínio lógico devem ser passados às crianças desde cedo, para aumentar sua dedução e conclusão de problemas.

Baggio(2000) *apud* Costa (2016) destaca que investir em ações que auxiliem a disseminar o ensino de computação é também investir em novas oportunidades para um público em situação socialmente vulnerável. O acesso e aprendizagem de linguagem de comunicação e informação, trazem não apenas oportunidades econômicas de gerar renda, mas também representam um importante capital social.

Para Oro *et al.* (2015), programação de computadores é uma opção poderosa na qualificação da formação básica das crianças, contribuindo até mesmo para o desenvolvimento em todas as áreas do conhecimento, uma vez que ensina a utilização do raciocínio lógico. Papert (1986) *apud* Oro *et al.* (2015) cita que programar é a capacidade de se comunicar com o computador em linguagem entendível por ambos. Como crianças são ótimas com linguagens, elas facilmente absorvem este tipo de comunicação.

Neste Capítulo 2, foram apresentados conceitos importantes para este trabalho. Instrucionismo foi definido como a forma de transmitir conhecimento através do computador, Construtivismo e Construcionismo formas em que o conhecimento é construído através do computador. Os conceitos *STEM* e *PC* foram os mais importantes utilizados neste trabalho, representando o pensamento analítico que une diversas áreas de ensino com um objetivo multidisciplinar e inovador. Foi abordado a Aprendizagem Criativa e a Computação Criativa, ambos conceitos nos quais os alunos podem aprender brincando. Por fim, definiu-se e buscou desmitificar o conceito de Lógica de Programação, apresentando-o como algo simples, que pode ser ilustrado, por exemplo, com uma receita de bolo. Com o conhecimento acerca destes conteúdos, é possível compreender com mais simplicidade e clareza esta monografia.

3. FERRAMENTAS, MÉTODOS E PRÁTICAS

Este capítulo visa abordar a importância da utilização de *softwares* educacionais e também apresentar as principais ferramentas utilizadas no ensino de programação para crianças, além de apresentar pesquisas e projetos relacionados ao objetivo desta monografia. Para tal, são apresentadas três experiências que se correlacionam com o tema de informática na educação e foram conduzidas pelo grupo de pesquisa do LIAG (Laboratório de Informática, Aprendizagem e Gestão) da Unicamp, além de um projeto da Universidade Tecnológica Federal do Paraná que visa apresentar o Pensamento Computacional nos anos iniciais do ensino fundamental. Por fim, são citados alguns eventos que buscam despertar e incentivar o interesse de jovens pela tecnologia.

3.1 Softwares Educacionais

A utilização de jogos computadorizados na educação proporciona ao aluno motivação, desenvolvendo também hábitos de persistência no desenvolvimento de desafios e tarefas. Os jogos, sob a ótica de crianças e adolescentes, se constituem a maneira mais divertida de aprender. Além disso, eles proporcionam a melhora da flexibilidade cognitiva, pois funcionam como uma ginástica mental, aumentando a rede de conexões neurais e alterando o fluxo sanguíneo no cérebro quando em estado de concentração. Os jogos educacionais se baseiam em uma abordagem autodirigida, isto é, aquela em que o sujeito aprende por si só, através da descoberta de relações e da interação com o software (TAROUCO e FABRE, 2004).

Atualmente, há uma vasta quantidade de *softwares* que buscam ensinar conceitos de programação a estudantes, como por exemplo: *Code.org*, *Scratch*, *AppInventor*, entre outros. Nas seções deste capítulo são destacados os mais conhecidos e aqueles que serão utilizados na metodologia de aplicação deste trabalho.

3.1.1 *Code.org*

O *Code.org* é uma organização sem fins lucrativos que visa expandir a ciência da computação em escolas. Busca fornecer um currículo mais amplo utilizando do ensino de lógica em escolas primárias e secundárias. É responsável pela organização da “Hora do Código” (detalhes do evento na Seção 3.3) (CODE, 2019).

Com o objetivo de tornar a ciência da computação parte do discurso internacional de educação, mais de 40% do tráfego do site vem de fora dos Estados Unidos e esse número

continua a subir. Contam com a parceria dos ministérios da educação de todo o mundo e trabalham com organizações internacionais, visando sua expansão (CODE, 2019).

O *Code.org* tem seus próprios cursos e faz parcerias, expandindo assim seu projeto pelo mundo. Todo seu conteúdo é livre para ser usado abertamente. A Figura 2 mostra a tela inicial do site que apresenta informações a respeito da “Hora do Código”, além de áreas exclusiva para alunos e professores que desejam participar do projeto.

Figura 2: Site *Code.org* - Tela inicial

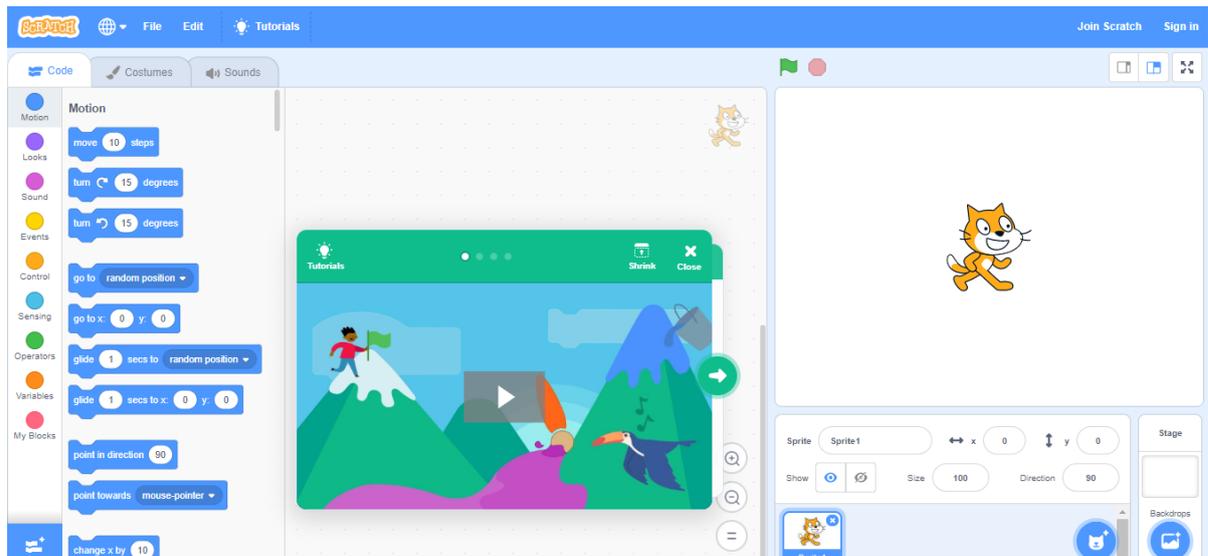


Fonte: *Code.org*

3.1.2 Scratch

O *Scratch* é uma linguagem de programação e uma comunidade *online* nas quais as crianças podem programar e compartilhar multimídia interativa com outras pessoas. À medida que criam com o *Scratch*, as crianças aprendem a pensar de forma criativa, a trabalhar de forma colaborativa e a pensar de forma sistemática — competências essenciais para todos na sociedade de hoje. O *Scratch* foi criado e é mantido pelo grupo *Lifelong Kindergarten* do *MIT Media Lab*. A Figura 3 mostra a tela inicial da criação de um projeto simples no site do *Scratch* (SCRATCH, 2019).

O *Scratch* é destinado especialmente para jovens entre os 8 e os 16 anos de idade, mas é usado por pessoas de todas as idades. O *ScratchJr*, uma versão simplificada do *Scratch*, foi concebida para crianças dos 5 aos 7 (SCRATCH, 2019).

Figura 3: O ambiente *Scratch*

Fonte: *Scratch*

3.2 Métodos

Com o objetivo de introduzir lógica de programação para crianças através de softwares educacionais, foram identificados alguns métodos de ensino voltados às crianças. Na Seção 3.2.1 é apresentado o método “Era uma vez”, construído pelo LIAG. Outros métodos são apresentados na Seção 3.2.2.

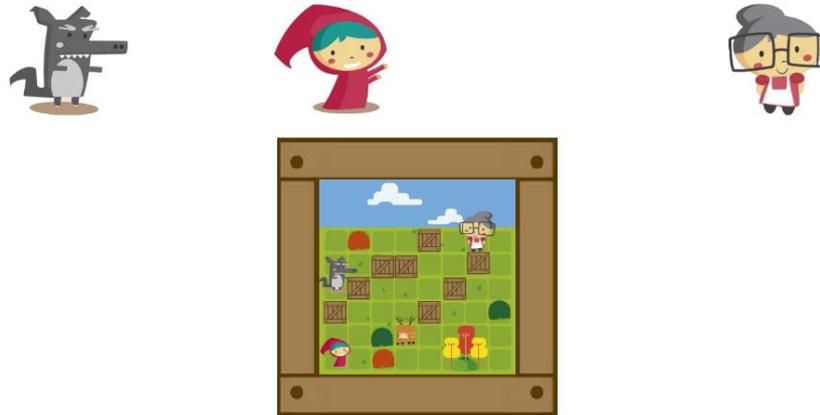
3.2.1 – “Era uma vez”

O método “Era uma vez” de Matsuzaki *et al.* (2016), foi desenvolvido buscando ensinar programação para crianças de maneira lúdica. O método tem seis aulas apresentando conceitos teóricos, dinâmicas em grupos, além de exercícios práticos utilizando a ferramenta *Scratch*.

As aulas têm como tema principal o conto infantil “Chapeuzinho Vermelho”. Toda parte gráfica é original e foi desenvolvida especificamente para o projeto. O método foi desenvolvido para crianças de 10 a 13 anos e visa ser aplicado em escolas ou mesmo em eventos de programação voltados para crianças.

As atividades a serem desenvolvidas pelas crianças se baseiam na história da Chapeuzinho Vermelho, com a personagem principal devendo “encontrar” sua Vovó, passando por diversos labirintos, evitando ser capturada pelo Lobo Mau. Na Figura 4 são apresentados os personagens utilizados no método.

Figura 4: Personagens do método "Era uma vez"



Fonte: Site LIAG

As aulas propostas pelo “Era uma vez” introduzem de forma gradativa conceitos teóricos de maneira a sempre incentivar os alunos. A primeira aula introduz o conceito de algoritmo aplicando atividades de algoritmo no papel, no qual, os alunos devem pintar quadrados dos tabuleiros da forma como desejarem e posteriormente devem passar esta folha para que outro colega descreva em forma de algoritmo o passo-a-passo para redesenhar o mesmo tabuleiro pintado pelo aluno inicialmente. A aula mostra o conceito de algoritmo como sendo uma “receita”. Na segunda aula é aplicado o “algoritmo do bis”, onde a turma deve descrever comandos para abertura da caixa. E as crianças devem citar exemplos de algoritmos no dia a dia. A terceira aula traz comandos e coordenadas para movimentação dos personagens no labirinto. Ocorre a ambientação dos alunos com a ferramenta *Scratch* através da resolução de algoritmos simples. Na quarta aula ocorre a introdução ao conceito de laço de repetição através da dinâmica “*cup song*”, que apresenta movimentos sequenciais durante a música. Na aula de número cinco os alunos devem encontrar o erro cometido nos exercícios que lhe são apresentados com as soluções incorretas. Por fim, na sexta e última aula o conceito de condição (se – então) é introduzido aos alunos. Além disso, há a previsão de realizar uma revisão para fixar os conceitos aprendidos (LIAG, 2019).

O método está disponível no site do LIAG¹ com todo o material das aulas e mais detalhes do plano de aula acima citado, para aplicação do método oferecidos de forma livre.

¹ <https://liag.ft.unicamp.br/computacaocriativa/era-uma-vez/>

3.2.2 – Outros métodos

Wiese *et al.* 2018, do departamento de Computação da Universidade Tecnológica Federal do Paraná realizaram um projeto de pesquisa voltado ao ensino de programação para crianças de até 12 anos utilizando o *Scratch* e disponibilizaram este um curso na plataforma *Udemy*. (Vide Figura 5). O curso é *online* e gratuito, apresenta 33 aulas, totalizando aproximadamente 4 horas. É sugerido realizar o curso em 6 semanas. O curso não conta com atividades *offline*. É utilizado o método tradicional de ensino, no qual um dos voluntários gravou aulas compartilhando sua tela apresentando o *Scratch* e também explicou alguns conceitos relacionados a programação.

Figura 5: Tela inicial do curso Programação em blocos com Scratch

Programação em blocos com Scratch!
 Aprendendo conceitos da engenharia de software e programação de forma divertida e interativa
 ★★★★★ 4,5 (106 classificações) 5.123 alunos inscritos
 Criado por DACOM - Departamento de Computação Universidade Tecnológica Federal do Paraná, Igor Wiese, Renan Kodama, Leonardo Mendonça Tuchinski, Erica Saito, BeaYasLore Senger, Correia e Plens, Ilzimara Silva
 Última atualização em 8/2018 Português

Pré-visualizar este curso

Você comprou este curso em 25 de Setembro de 2019

Ir para o curso

Compartilhar este curso

Preço atual
Gratuito

Este curso inclui

- Vídeo sob demanda de 4 horas
- Acesso total vitalício
- Acesso no dispositivo móvel e na TV

Conteúdo do curso Expandir tudo 33 aulas 03:48:32

+ Introdução	1 aula	02:16
+ 1ª Semana	5 aulas	29:05
+ 2ª Semana	6 aulas	36:01
+ 3ª Semana	5 aulas	41:14
+ 4ª Semana	5 aulas	42:16
+ 5ª Semana	5 aulas	30:30
+ 6ª Semana	6 aulas	47:10

Fonte: Udemy

A plataforma *Code.org* possui quatro cursos *online* que incentivam o aprendizado de ciência da computação para crianças de maneira lúdica. Nos cursos são abordados conceitos como depuração, sequência, laços, repetição, condicionais entre outros conceitos. Há também atividades *offline*. Os personagens utilizados nos cursos são muito conhecidos das crianças atualmente, como por exemplo *Frozen*, *Minicraft*, *Angry Birds*, o que atrai ainda mais a atenção das crianças (Figura 6).

Figura 6: Cursos *Code.org*

Fundamentos de Ciência da Computação [Ver meus cursos recentes >](#)

Comece aprendendo uma introdução à ciência da computação no Estúdio de Código com estes cursos de 20 horas para todas as idades.



Curso 1
4-6 anos de idade

Comece com o Curso 1 para alunos que estão começando a ler.



Curso 2
A partir de 6 anos (as crianças já devem saber ler)

Comece com o Curso 2 para alunos que sabem ler.



Curso 3
8-18 anos de idade

O Curso 3 é uma continuação do Curso 2.



Curso 4
10-18 anos de idade

O curso 4 é destinado aos alunos que já concluíram os cursos 2 e 3.

Aulas off-line

Se não tiver computadores, teste essas lições off-line em sua sala de aula.



Fonte: *Code.org*

3.3 Experiências práticas com Pensamento Computacional

No desenvolvimento desta monografia, foi realizado um breve levantamento bibliográfico acerca de projetos que levam o ensino de programação para crianças. Foi possível observar que há várias pesquisas e projetos voltados a este fim. Artigos disponibilizados no Congresso da Sociedade Brasileira de Computação e, principalmente, os trabalhos desenvolvidos no LIAG e na Unicamp foram considerados para mostrar ações práticas relacionadas ao Pensamento Computacional.

O projeto de Stella (2016) teve como objetivo analisar a utilização de tecnologias associadas às disciplinas da grade curricular do ensino fundamental. Com o auxílio da ferramenta *Scratch*, o projeto buscou apresentar uma introdução a conceitos de linguagem de programação de maneira lúdica, de forma a complementar o desenvolvimento do raciocínio lógico das crianças. O mesmo foi realizado com crianças entre 8 e 11 anos, no contraturno (horário contrário as aulas) da escola. Foram feitas cinco dinâmicas que tinham como objetivo incentivar o Pensamento Computacional e o interesse por programação. Com isto, foi possível analisar o interesse e a motivação dos participantes. Baseado em metodologias instrucionista e construcionista foram elaborados desafios, como: utilização livre dos blocos de comandos, criação de uma fábula animada e a reprodução de um cartão de natal. Em seus resultados, Stella (2016) aponta que as crianças aprovaram a participação no projeto e se mostraram bem

interessadas nos recursos disponibilizados pelo *Scratch*.

Matsuzaki *et al.* (2016) apresentou em seu projeto práticas pedagógicas visando explorar o conteúdo de Lógica de Programação e Pensamento Computacional, de maneira motivadora e criativa. Foram propostas atividades de Programação para crianças de 10 a 13 anos de idade usando como ferramenta de programação o ambiente *Scratch*.

No trabalho de Teche e Recchina (2017), o foco foi nos professores do ensino fundamental que, de maneira indireta, também podem trabalhar o Pensamento Computacional com crianças. Foram apresentadas aos docentes ferramentas tecnológicas que podem auxiliar na transmissão de conteúdos estabelecidos no currículo escolar, tendo como objetivo identificar se os professores, quando instruídos, se sentem confortáveis para incluir essas ferramentas em seu dia a dia. Para isso, foram utilizados conceitos como: Construcionismo, Letramento Digital, Pensamento Computacional e Computação Criativa, apresentados de maneira lúdica através das ferramentas: *Scratch*, Arduino e Circuitos em Papel. O grupo tinha 9 docentes com média de idade de 44 anos. Foram realizadas cinco oficinas nas quais foram aplicadas três ferramentas, analisando assim o interesse e progressão dos participantes. Segundo os autores, o grupo foi participativo e se mostrou muito motivado em aprender e aplicar tais ferramentas em sala de aula.

Na pesquisa de Glizt e Koscianski (2017), buscou-se analisar as contribuições do Pensamento Computacional no desenvolvimento do raciocínio lógico dos alunos, por meio de atividades lúdicas, tais como conversão de números binários, métodos de ordenação, algoritmos, linguagem de programação e lógica, em uma turma do ensino fundamental da rede pública municipal de ensino da cidade de Ponta Grossa, Paraná. Com o estudo foi possível evidenciar que, quando os alunos se envolvem em atividades que abrangem conceitos computacionais, habilidades além do raciocínio lógico são desenvolvidas, possibilitando adquirir conhecimentos antes delimitados a áreas específicas.

De acordo com Batista *et al.* (2015), as olimpíadas científicas também são uma iniciativa para a popularização e difusão da ciência e tecnologia junto aos jovens. Existem diversos exemplos mostrados por Batista *et al.* (2015) que ocorrem no Brasil:

- A Olimpíada Brasileira de Informática (OBI) visa despertar nos alunos o interesse por uma ciência (principalmente ciências da computação), através de desafios, engenhosidade e uma saudável competição. Atualmente sua organização é realizada pelo Instituto de Computação da Unicamp, sendo dividida em duas modalidades: Iniciação e Programação e em vários níveis conforme o ano escolar do aluno (OBI, 2019);

- A Olimpíada Brasileira de Robótica (OBR) visa estimular alunos às carreiras científico-tecnológicas, identificar jovens talentosos e promovendo debates e atualizações no ensino-aprendizagem do Brasil. Costuma ser realizada com estudantes de escolas públicas ou privadas e divide-se em duas modalidades teórica e prática. Esta olimpíada ocorre desde 2007 e é apoiada pelo Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), Ministério da Educação (MEC) e Ministério da Ciência, Tecnologia, Inovações e Comunicações (MCTIC), além do suporte da Sociedade Brasileira de Computação (SBC) e RoboCup Federation. (OBR, 2019);
- Robocode Brasil é uma competição que utiliza Aprendizagem Baseada em Problemas (PBL) para estimular alunos a aprender conceitos de linguagem de programação no qual equipes criam “robôs” virtuais para competirem entre si. Tem como objetivos a programação de estratégias e comparação de comportamentos para vencer disputas. Esta competição é destinada tanto a alunos de cursos técnicos quanto estudantes do ensino superior (ROBOCODE, 2019).

Além das Olimpíadas, há também iniciativas utilizando softwares educacionais para introdução de lógica de programação às crianças, como os exemplos a seguir:

- A “Hora do Código” é um movimento global que busca desmistificar a programação mostrando que a mesma está ao alcance do aprendizado de qualquer pessoa. O movimento começa com atividades de programação com 1 hora de duração, mas acaba expandindo-se para quem sabe despertar o interesse de crianças e jovens em Ciência da Computação. A princípio, o movimento é realizado todo ano durante a Semana da Educação em Ciência da Computação (geralmente em dezembro) (HORA DO CÓDIGO, 2019);
- O *Scratch Day* é um evento global, no qual as pessoas se reúnem para celebrar o *Scratch*. Neste evento ocorre a troca de conhecimento, compartilhamento de projetos, aprendizagem, entre outras atividades. No ano de 2018, aconteceram mais de 1200 eventos relacionados ao *Scratch Day* pelo mundo (SCRATCH DAY, 2019).

Estes trabalhos, projetos e olimpíadas são exemplos de ações que estão acontecendo no sistema de educação que vem sendo apoiadas por pesquisas.

4. MÉTODOS E MATERIAIS

Este trabalho baseou-se nos projetos desenvolvidos durante a disciplina Gestão de Projetos, ministrada pelo Professor Dr. Marcos Augusto Francisco Borges, na Faculdade de Tecnologia da Unicamp no campus I de Limeira. Nos últimos anos, quando ministra esta disciplina, o professor propõe uma ação de Pensamento Computacional em escolas, conduzida por seus discentes de graduação, como experiência prática de projeto. Ao longo dos anos, esta atividade vem se reestruturando e aprimorando sua aplicação. A ação tem alcançado muitas crianças – atingiu alguns milhares até esta data - e é realizada, majoritariamente, nas escolas públicas de Limeira, com alunos do fundamental I e/ou II.

Durante as aulas da disciplina da Unicamp, os alunos de graduação são apresentados a conceitos básicos referentes a Gestão de Projetos e Pensamento Computacional. A disciplina tem quatro horas de aulas semanais e usa como apoio a ferramenta *Google Classroom*, na qual são disponibilizados o programa, as atividades, e material de apoio. Ao longo do período da disciplina de graduação, um membro do LIAG, com experiência nesse tipo de ação, fica focado em apoiar os grupos na escolha das ferramentas e métodos e na identificação de oportunidades nas escolas de Limeira para aplicação dos projetos.

Os alunos da disciplina devem se planejar (conforme as boas práticas de Gestão de Projetos), no decorrer do semestre, e realizar projetos que devem consistir em aulas para crianças e jovens envolvendo tecnologia. São requisitos desta ação: envolver conceitos de Pensamento Computacional, oferecer um mínimo de 8 horas de aula para um mínimo 10 alunos, sempre com pelo menos dois participantes acompanhando as aulas com as crianças, realizar pesquisa de satisfação e ter um plano de aula pré-definido e aprovado pelo professor antes do início das atividades. A sugestão é de que os alunos utilizem o método “Era uma vez” do LIAG, porém os discentes da disciplina da Unicamp podem optar por outros métodos ou mesmo pela adaptação do “Era uma vez” quando necessário / possível.

Com o objetivo de levantar quais métodos os alunos da Unicamp vêm utilizando e quais os resultados obtidos, este projeto construiu um questionário com 10 questões (apresentado no Apêndice I), para ser aplicado.

A primeira questão busca identificar quais as ferramentas que foram utilizadas pelos grupos durante o projeto. O objetivo desta questão é verificar quais as ferramentas mais utilizadas para que, através das respostas obtidas, possa-se discutir sugestões de outros métodos e ferramentas. A Questão 2 busca mapear as escolas nas quais os projetos forem desenvolvidos

e a infraestrutura disponível nos laboratórios dessas escolas, visando analisar o cenário atual da infraestrutura das escolas e entidades participantes. A terceira questão almeja levantar a idade dos participantes do projeto (crianças e jovens participando das aulas), para analisar se há alguma relação entre essa faixa etária, o método aplicado e o resultado geral obtido com o projeto. A Questão 4 pergunta a quantidade de horas de aulas dadas, para verificar se o tempo sugerido pela disciplina foi adequado para apresentação dos conceitos abordados. A quinta questão busca identificar a visibilidade dos conceitos *STEM* e Pensamento Computacional através do conhecimento de cada um dos termos descritos pelos alunos da disciplina de Gestão de Projetos. Na Questão 6, os participantes devem descrever os métodos de ensino utilizados, para verificar quantos grupos seguem a sugestão inicial do projeto de utilização do “Era uma vez” e também identificar quais são as adaptações conduzidas e o que as motivaram. Ainda buscando entender com mais detalhes as adaptações necessárias, na Questão 7, os grupos devem abordar as maiores dificuldades encontradas, visando entender os principais pontos que os grupos precisaram ajustar durante as aulas. A Questão 8 tem o objetivo de entender o que os grupos identificaram como pontos positivos no desenvolvimento de seus projetos. Na Questão 9, busca-se entender se os grupos gostaram de realizar este tipo de projeto e se as crianças também gostaram de participar de ações deste tipo. A décima pergunta tem como objetivo entender a visão dos alunos da disciplina com relação ao motivo deste tipo de trabalho não ocorrer com mais frequência nas escolas.

Este questionário foi aplicado em um semestre no qual foram conduzidas muitas ações em escolas e, no próximo capítulo, é realizada uma análise das respostas obtidas.

5. RESULTADOS E DISCUSSÕES

Este projeto buscou analisar ações de desenvolvimento do Pensamento Computacional. O objetivo dessa análise foi levantar as principais dificuldades encontradas durante o desenvolvimento do projeto e propor melhorias, para auxiliar ainda mais aqueles que se interessarem em levar o conhecimento da lógica de programação às crianças de maneira lúdica e de forma eficiente e eficaz, fazendo com que as crianças aprendam brincando e resolvendo problemas.

Os discentes de duas turmas da disciplina de Gestão de Projetos da Faculdade de Tecnologia da Unicamp no primeiro semestre de 2019 (a Turma A, noturna, foi composta por 47 alunos do curso de “Tecnologia em Análise e Desenvolvimento de Sistemas” e a Turma B, diurna, 36 alunos do curso de “Sistemas de Informação”), dividiram-se em 17 grupos de 4 ou 5 alunos. Este trabalho disponibilizou o questionário apresentado no Capítulo 4, de forma *online* no *Google Formulários*. No dia 02 de junho, foi divulgado no Mural do *Google Classroom* de ambas as turmas o pedido para que os grupos respondessem o questionário. No decorrer das aulas, foi reforçada pelo professor a importância da contribuição dos alunos com suas experiências durante o projeto. Este questionário ficou disponível para ser respondido entre os dias 02 e 18 de junho de 2019. Os alunos da disciplina estavam concluindo seus projetos e puderam contribuir no desenvolvimento desta monografia com o relato de suas experiências. Todos os membros dos grupos podiam, de maneira anônima, responder ao questionário disponibilizado. O questionário foi respondido por, pelo menos, um membro de cada grupo, totalizando 42 respostas. As respostas obtidas foram tabuladas e analisadas, separando-as por grupos. Mesmo questões que foram abordadas de maneira diferente por cada membro, foram consideradas. Para algumas questões, a resposta final considerada foi o que a maioria dos membros apontou.

Os projetos foram realizados em 3 diferentes escolas públicas municipais de Limeira, com crianças de idade média entre 8 e 12 anos. Houve também grupos que realizaram seus projetos em uma ONG com crianças de aproximadamente de 12 anos, um dos grupos trabalhou em instituições com crianças especiais. Outros em EJAs (educação de jovens e adultos), atuando com o público adulto, com idades entre 30 e 60 anos. Os projetos atingiram mais 200 pessoas somente nesse semestre.

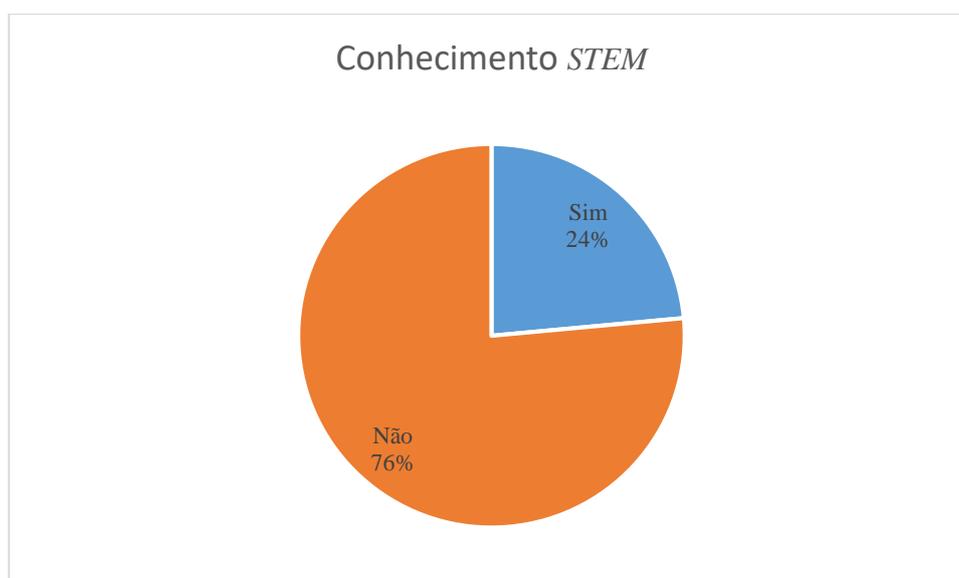
O membro do LIAG, responsável por auxiliar os grupos no desenvolvimento de seus projetos, também disponibilizou informações referentes aos grupos e escolas que participaram

do projeto, informações que contribuirão para algumas análises desta pesquisa. No dia 18 de junho, ocorreu o evento “Segunda Mostra do LIAG”, no qual foi possível conhecer um resumo dos projetos conduzidos. Foi uma experiência muito interessante, pois os alunos de graduação que apresentaram as aulas demonstraram entusiasmo com o projeto, compartilhando as experiências vividas e os resultados alcançados que foram positivos. Todas as apresentações foram filmadas e estão disponíveis em uma página com acesso livre pela Internet².

Com base nas respostas obtidas (apresentadas de forma completa no Apêndice II), foi realizada uma tabulação e análise das respostas dos 17 grupos que contribuirão na obtenção dos dados do estudo desta monografia. Esta análise é detalhada a seguir e posteriormente são discutidos os resultados.

Com base nas respostas relativas a quinta questão, foi possível observar que *STEM* foi pouco descrito pelos grupos, sendo que apenas 24% dos grupos descreveram o conceito de forma correta (vide Gráfico 1). O conceito de Pensamento Computacional, colocado como pré-requisito para as aulas, foi citado de maneira correta por 76% dos grupos, conforme mostrado no Gráfico 2. Isto mostra que mesmo entre alunos de graduação na área de informática, o desconhecimento de *STEM* ainda é bem significativo. É interessante transmitir novos conceitos aos jovens para que assim possam entender melhor e aplique-os em outros trabalhos.

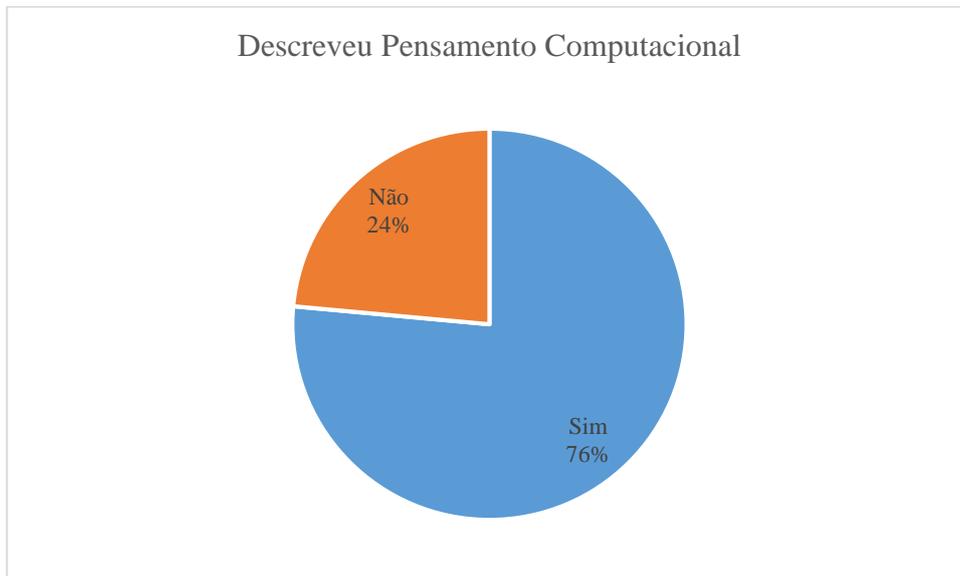
Gráfico 1 - Descrição do conceito *STEM*.



Fonte: Elaborado pelo autor.

² Disponível em: <https://liag.ft.unicamp.br/2mostraliag/>

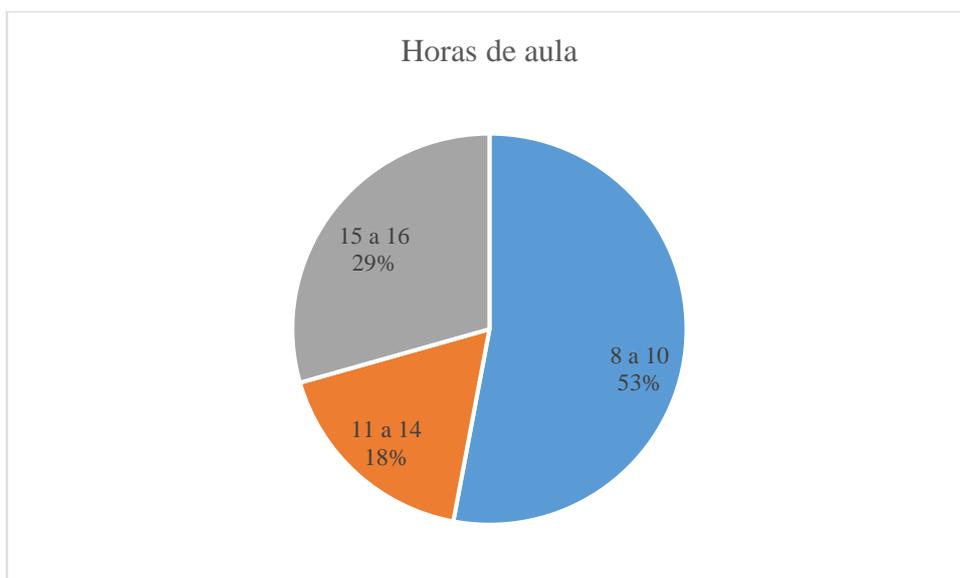
Gráfico 2 - Descrição do Pensamento Computacional.



Fonte: Elaborado pelo autor.

No Gráfico 3, baseado na Questão 4, foi identificado que a maioria (53%) dos grupos utilizou o tempo recomendado pelo método “Era uma vez” para aplicação das aulas (8 a 10h). O tempo se mostra adequado para transmissão dos conceitos e entende-se que não há necessidades de ajuste.

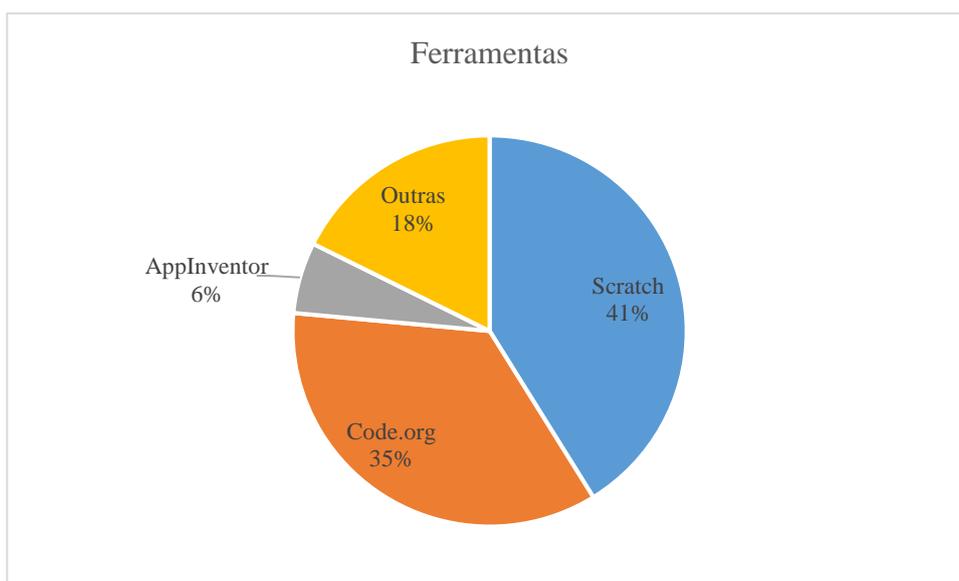
Gráfico 3 – Tempo de horas aula ministradas (em horas).



Fonte: Elaborado pelo autor.

Com base nas respostas da Questão 1, o Gráfico 4, mostra as ferramentas utilizadas pelos grupos: *Scratch*, *Code.org*, *AppInventor* e outras. A maioria (41%) dos grupos utilizou *Scratch*, porém é possível identificar que o *Code.org* teve uma grande utilização (35%). Embora a ferramenta *Scratch* foi a recomendada pela disciplina, é possível observar que o *Code.org* teve uma utilização em nível ao *Scratch*. Alguns grupos abordaram que passaram a utilizar o *Code.org*, pois “a plataforma possui temas que despertam mais o interesse das crianças” e que o site do *Code.org* contém diversos níveis de atividades, tanto as mais simples quanto as mais complexas. Essa diversidade e grande quantidade de atividades disponibilizadas no *Code.org* facilita a escolha da atividade prática utilizada na aula conforme o nível de conhecimento prévio da turma sobre computação.

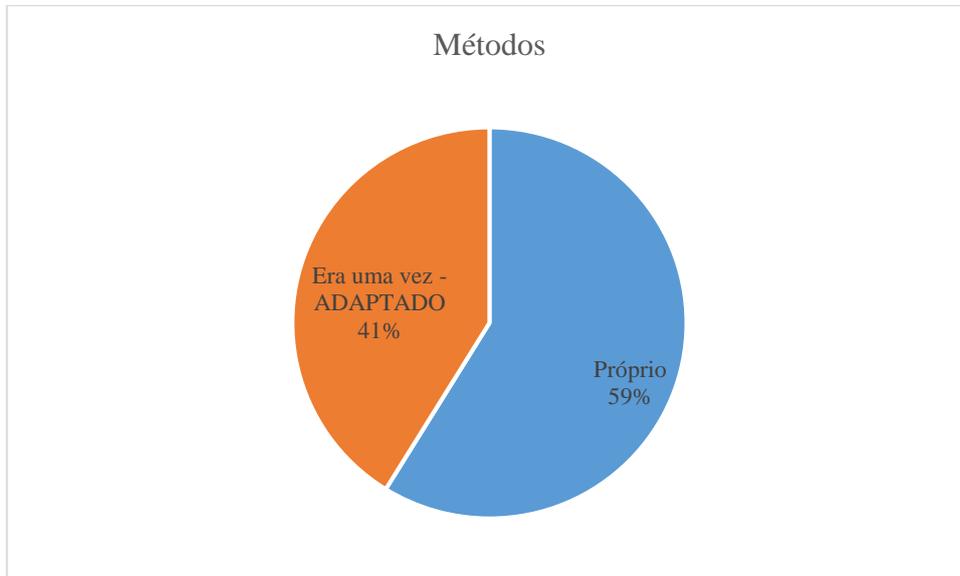
Gráfico 4- Ferramentas utilizadas.



Fonte: Elaborado pelo autor.

Conforme observado nas respostas dadas para a sexta questão, e ilustrado no Gráfico 5, 59% dos grupos utilizaram um método de ensino criado pelos próprios alunos. Embora tenha sido o recomendado, apenas 41% utilizaram o método “Era uma vez”, e mesmo estes realizaram algum tipo de adaptação. Nenhum grupo conseguiu aplicar o método em sua forma original e também não foram utilizados outros métodos citados na Seção 3.2.2. Nos métodos criados pelos próprios alunos, foi possível observar a ampla utilização dos cursos disponibilizados pela plataforma *Code.org*.

Gráfico 5 - Métodos aplicados.

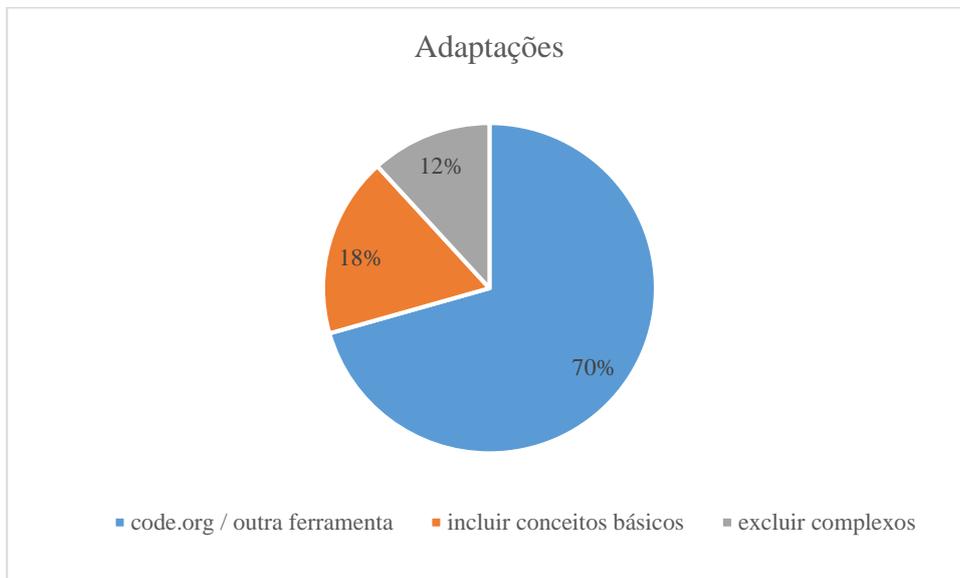


Fonte: Elaborado pelo autor.

Com relação as adaptações e as dificuldades abordadas pelos grupos nas respostas às questões 6 e 7, o Gráfico 6 mostra que os grupos realizaram adaptações no método “Era uma vez”: é possível observar que 70% usaram outras ferramentas no decorrer da aplicação do método. Alguns grupos incluíram conceitos mais básicos, como, por exemplo, ligar o computador e utilizar o *software Microsoft Paint* (programa que já vem instalado no Windows e é utilizado para desenhar ou até mesmo editar imagens de forma simples). Outros grupos excluíram conceitos mais “complexos” devido a idade das crianças ser abaixo daquela prevista para o “Era uma vez”.

Ainda baseando-se nas respostas às questões 6 e 7, que estão diretamente relacionadas, as maiores dificuldades encontradas pelos grupos foram em manter a atenção das crianças, principalmente nas aulas teóricas, nas quais apenas introduziam contextos de computação sem o auxílio de ferramentas *online*. Isso deve ser considerado, justificando a exploração de atividades práticas em todas as aulas. Algumas barreiras como a alfabetização, pouca familiaridade com computadores, dificultou não apenas a aplicação do método “Era uma vez”, mas também o desenvolvimento do projeto, que precisou de adaptações por parte de alguns grupos, o que justifica a exclusão de conceitos mais complexos e a inclusão dos básicos. Ainda com base nessas questões, um grupo indicou ter dificuldade no agendamento dos laboratórios e na comunicação com a escola, o que são dificuldades associadas a Gestão de Projetos e não a aplicação das aulas em si e não serão consideradas neste trabalho.

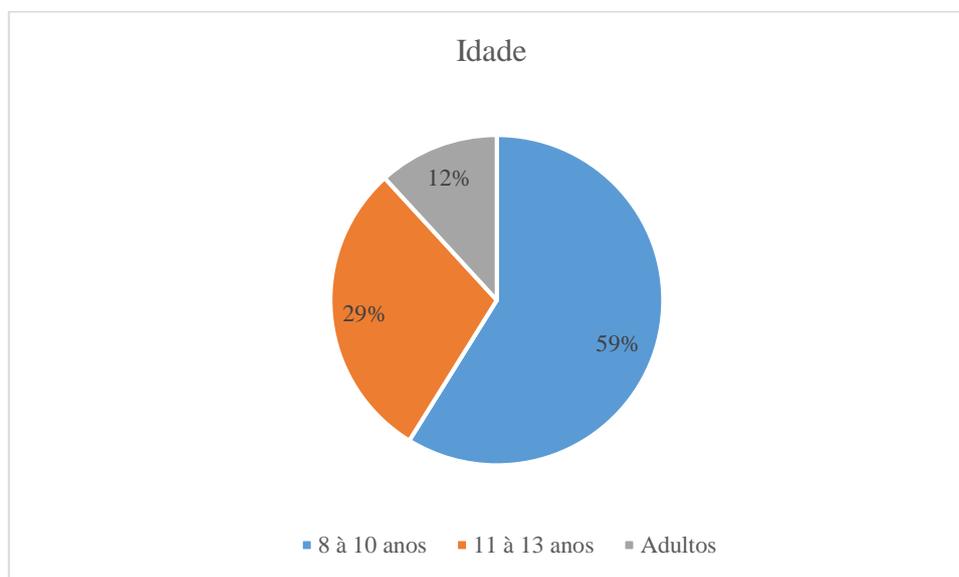
Gráfico 6 - Tipos de adaptações necessárias.



Fonte: Elaborado pelo autor.

Além das dificuldades, outro fator que pode ter impactado em adaptações do método é que a maioria dos grupos realizou seus projetos em escolas municipais de Limeira-SP, do Fundamental I (primeiro a quinto anos, 6 a 10 anos em média). Com base nas respostas à Questão 3, a maioria das crianças que participaram dos projetos tinha entre 8 e 10 anos (59%), o que é ilustrado no Gráfico 7. É uma faixa etária abaixo do recomendado pelo método “Era uma vez”, o que também justifica a necessidade da inclusão de conceitos mais simples, como explicações sobre o computador e seus *softwares* (*Paint*). A idade das crianças também pode ter impactado na exclusão de alguns conteúdos.

Gráfico 7 - Idade dos participantes.



Fonte: Elaborado pelo autor.

Após as análises apresentadas, considerando as adaptações realizadas pelos grupos, entende-se que o método “Era uma vez” pode ser usado de forma a introduzir programação às crianças, porém é interessante que os grupos façam adaptações de acordo com a realidade na qual ele será usado. Com base nas adaptações que os grupos realizaram e nas dificuldades encontradas, sugere-se que seja feita uma atualização dos personagens utilizados, além da inclusão de mais aulas práticas. A integração da utilização da plataforma *Code.org* no decorrer das aulas deve ser analisada, pois foi muito utilizada pelos grupos. No *Code.org*, por exemplo, há vários personagens atuais conhecidos pelas crianças, o que pode facilitar sua motivação na aprendizagem. Outra proposta é seguir a estratégia do *Code.org*, oferecendo várias versões, cada uma adequada a uma faixa etária, dado que o método “Era uma vez” original se dirige a alunos de uma faixa etária específica.

Os métodos de ensino tradicionais podem não despertar mais o interesse de crianças e jovens. Conforme abordado na décima questão pelos alunos que apresentaram os treinamentos, os grupos entendem que este tipo de ação não ocorre com mais frequência nas escolas públicas devido a falta de profissionais qualificados para tal fim e a infraestrutura inadequada das escolas. Os resultados da questão sobre infraestrutura das escolas não foram divulgados com detalhes, devido premissa de sigilo deste trabalho.

Foi possível concluir, baseando-se nas respostas da Questão 9, que projetos deste tipo são de grande importância para divulgação de trabalhos voluntários que visam trazer benefícios

a sociedade. 94% dos grupos expressaram na pesquisa que gostariam de realizar este tipo de ação novamente (Gráfico 8). Isto reforça ainda mais a importância da relação universidade e sociedade. Através de ações como as apresentadas neste projeto, é possível impactar a educação brasileira, contando com o apoio voluntário de jovens universitários, que indicaram estar dispostos a auxiliar.

Gráfico 8 - Realizar o projeto novamente.



Fonte: Elaborado pelo autor.

Na questão referente aos pontos positivos identificados pelos grupos, foram relacionados a motivação das crianças em aprender de uma maneira diferente e o apoio a uma atividade relacionada ao Pensamento Computacional.

Com base no que foi respondido no questionário, há indícios de que, se introduzida da maneira correta, este tipo de ação poderá auxiliar cada vez mais as crianças nas resoluções de problemas de maneira efetiva e criativa através de atividades mais motivadora, baseadas em imaginação, paixão e na troca de experiências com os colegas.

Este tipo de projeto tem grande potencial para mitigar alguns problemas da educação brasileira, podendo ser visto como um primeiro passo em direção a aplicação de tecnologia e Pensamento Computacional nas escolas.

6. CONCLUSÃO

Motivado por fatores desafiadores da educação brasileira atual, este trabalho buscou abordar a importância do conhecimento de conceitos como *STEM* e Pensamento Computacional, para promover melhorias no ensino tradicional, principalmente em escolas públicas brasileiras.

Primeiramente, este trabalho apresentou um entendimento do cenário atual da educação brasileira, sugerindo oportunidades para melhoria dessa realidade. Os conceitos de *STEM* e, principalmente, de Pensamento Computacional, foram discutidos e apresentados, inclusive através de experiências práticas, identificando e levantando alguns projetos que têm potencial para iniciar mudanças na educação do país.

O objetivo principal desta monografia foi realizar um estudo acerca de ações de promoção do Pensamento Computacional sendo conduzidas há alguns anos pelo LIAG da FT-Unicamp. Buscou-se, com isso, contribuir para a identificação e aprimoramento de uma estratégia de aplicação de atividades motivadora.

Após o entendimento do cenário atual, e tendo identificado qual é o problema que se deseja solucionar, foi acompanhada e analisada uma amostra de projetos executados durante a disciplina de Gestão de Projetos da Faculdade de Tecnologia com esse objetivo. Fatores como dificuldades encontradas nas aplicações do método de ensino de programação, adaptações realizadas durante a execução dos projetos, pontos positivos a serem mantidos foram levados em consideração, para as propostas de melhorias sugeridas e as oportunidades identificadas.

De modo geral, os itens apresentados no decorrer do trabalho foram suficientes para atingir o objetivo proposto que é apresentar a importância do tema não apenas para os dias atuais, mas também para o futuro, pois a revisão e um ciclo de melhoria contínua em métodos de ensino de programação é um fator relevante para aumentar a qualidade dos métodos de ensino, refletindo cada vez mais em uma melhor experiência tanto para as crianças que participam quanto aos discentes que levam este tipo de projeto as escolas.

O resultado do projeto mostra que o método “Era uma vez”, principal base oferecida para as atividades, não atendeu as expectativas. A maior parte das ações não se baseou nesse método e aquelas que se basearam, fizeram adaptações. O trabalho conclui que o método pode ser melhorado, incluindo a utilização de outra plataforma *online* (*Code.org*, por exemplo) e também com a criação de variações do método, para atingir públicos de outras idades, perfis e experiências prévias. Seria um trabalho futuro interessante colocar em prática mudanças na metodologia e analisar sua efetividade, para sempre buscar aprimorá-la e

atualizá-la.

O objetivo deste trabalho foi atingido, contribuindo com a análise de ações associadas ao Pensamento Computacional em escolas por voluntários de cursos de graduação. Foi também possível observar que este tipo de ação traz resultados positivos tanto para quem o aplica quanto para quem o recebe, com mais de 200 pessoas recebendo uma introdução a conceitos do Pensamento Computacional de maneira criativa, divertida e mais eficaz que a maneira tradicional de ensino. Assim, mostra-se que é possível levar o Pensamento Computacional para as escolas públicas do Brasil, algo essencial para a melhoria do ensino no país e que deve ser considerado prioritário pela nossa sociedade.

REFERÊNCIAS

Acerca do Scratch. Disponível em: < <https://scratch.mit.edu/about>>. Acesso em: 06 de set. de 2019.

Acompanhe o Robocode 2019. Disponível em: <<https://liag.ft.unicamp.br/robocode/robocode-2019/>>. Acesso em: 15 de set. de 2019.

BATISTA, E.J.S. *et al.* **Utilizando o Scratch como ferramenta de apoio para desenvolver o raciocínio lógico das crianças do ensino básico de uma forma multidisciplinar.** – in Anais do XXI Workshop do Congresso Brasileiro de Informática na Escola - Universidade Federal de Mato Grosso do Sul – Campus Ponta Porã, 2015.

BONILLA, M. H. **Inclusão digital nas escolas.** in: PINHEIRO, Antonio Carlos Ferreira; ANANIAS, Mauricéia (Org.). Educação, direitos humanos e inclusão social: histórias, memórias e políticas educacionais. João Pessoa: Editora universitária da UFPB, 2009, p. 183-200. (V. 2).

Code.org - Sobre. Disponível em: < <https://code.org/about>>. Acesso em: 06 de set. de 2019.

Computação Criativa uma introdução ao pensamento computacional baseada no conceito de design. MIT (*Massachusetts Institute of Technology*), Setembro de 2011 Tradução por EduScratch, Outubro de 2011.

COSTA, T. C. A. **Uma abordagem construcionista da utilização dos computadores na educação.** 3º Simpósio Hipertexto e Tecnologias na Educação, 2010.

COSTA, T. A.; *et al.* **A Importância da Computação para Alunos do Ensino Fundamental: Ações, Possibilidades e Benefícios.** in Anais do XXII Workshop de Informática na Escola. 2016. 9p.

DINIZ, S. N. F. **O uso das novas tecnologias em sala de aula.** Dissertação de pós

graduação. Universidade Federal de Santa Catarina Programa de Pós-graduação em Engenharia de Produção. - Florianópolis - SC 2001.

Era uma vez Sobre. LIAG (Laboratório de Informática, Aprendizagem e Gestão). Disponível em: < <https://liag.ft.unicamp.br/computacaocriativa/era-uma-vez/>>. Acesso em: 12 de out. de 2019.

FRANÇA, Rozelma Soares de; AMARAL, Haroldo José Costa do. **Proposta Metodológica de Ensino e Avaliação para o Desenvolvimento do Pensamento Computacional com o Uso do Scratch.** in Anais do XIX Workshop de Informática na Escola, II Congresso Brasileiro de Informática na Educação, 2013.

GLIZT, F. R. O. **O Pensamento Computacional nos anos iniciais do ensino fundamental.** Dissertação (Mestrado em Ensino de Ciência e Tecnologia) - Universidade Tecnológica Federal do Paraná, Ponta Grossa, 2017.

Glossário. STEM . Disponível em: <<http://porvir.org/stem//>>. Acesso em: 14 de ago. de 2019.

KLEMMANN, M. *et al.* **Estratégias Pedagógicas no Ensino de Algoritmos e Programação associadas ao uso de jogos educacionais.** Novas tecnologias na educação. CINTED – UFRGS, 2006.

MATSUZAKI, I. *et al.* **Proposta de ensino de programação para crianças com Scratch e Pensamento Computacional.** Relatório final, Programa Institucional de Bolsas de Iniciação Tecnológica – CNPq - UNICAMP, 2016.

MEDINA M., FERTING C.. **Algoritmos e programação: teoria e prática.** São Paulo: Novatec; 2005.

OLIVEIRA, S. G. S.. **A alfabetização científica no ensino fundamental: desafios encontrados pelos docentes em escolas municipais de Ilhéus-Bahia.** Universidade estadual de Santa Cruz – Programa de pós-graduação em educação em ciências. Ilhéus – BA: UESC, 2017.

ORO, N. T. et al. **A Olimpíada de Programação de Computadores para Estudantes do Ensino Fundamental: A interdisciplinaridade por meio do software Scratch.** in Anais do XXI Workshop do Congresso Brasileiro de Informática na Escola. 2015.

O que é a hora do código. Disponível em: < <http://programae.org.br/horadocodigo/>>. Acesso em: 18 de jun. de 2019.

O que é a OBR ? (Olimpíada Brasileira de Robótica). Disponível em: < <http://www.obr.org.br/o-que-e-a-obr/>>. Acesso em: 15 de jun. de 2019.

PUGLIESE, G. O. **Os modelos pedagógicos de ensino de ciências em dois programas educacionais baseados em STEM (Science, Technology, Engineering, and Mathematics).** 2017. Dissertação (mestrado) - Universidade Estadual de Campinas, Instituto de Biologia, Campinas, SP.

RAABE, A.; ZANCHETT, G.; VAHLDICK, A. **Jogos de programar como uma abordagem para os primeiros contatos dos estudantes com a programação** - in Anais dos Workshops do Congresso Brasileiro de Informática na Educação, 2015.

RESNICK, M. *et al.* **Scratch: programming for all.** Communications of the ACM, 2009. vol 52.

RESNICK, M. **Lifelong Kindergarten: Cultivating Creativity through Projects, Passion, Peers, and Play.** MIT Media Lab. Publicado pela MIT Press (2017). Trechos do Capítulo 1: Aprendizagem Criativa.

RODRIGUEZ, C. L. *et al.* **Pensamento Computacional: transformando ideias em jogos digitais usando o Scratch.** in Anais do XXII Workshop do Congresso Brasileiro de Informática na Escola. 2015.

Scratch Day. Disponível em: < <https://day.scratch.mit.edu/>>. Acesso em: 20 de jun. de 2019.

Sobre a OBI (Olimpíada Brasileira de Informática). Disponível em:

<<https://olimpiada.ic.unicamp.br/info/>>. Acesso em: 15 de jun. de 2019.

STELLA, A. L. **Utilizando o Pensamento Computacional e a Computação Criativa no Ensino de Linguagem de Programação Scratch para alunos do ensino fundamental.** 2016. 92 f. Dissertação (Mestrado) - Curso de Mestrado em Tecnologia, Universidade Estadual de Campinas, Limeira, 2016.

TECHE, T. V.; RECCHINA, F. A. **Aplicação do Pensamento Computacional e Computação Criativa em escolas públicas.** Trabalho de conclusão de curso (Bacharel em Sistemas da Informação) - Faculdade de Tecnologia, UNICAMP, Limeira, 2017.

TAROUCO, L. M. R.; FABRE, M. C. J. M. **Jogos Educacionais.** Revista Novas Tecnologias na Educação V.2, N.1, 2004.

VALENTE, J. A. **Computadores na Sociedade do Conhecimento.** 1.ed. Campinas: UNICAMP/ NIED, 1999. 156p. Editora: Unicamp NIED.

VALENTE, J. A. **Informática na educação: instrucionismo x construcionismo.** Disponível em: < <http://www.divertire.com.br/educacional/artigos/7.htm>>. Acesso em: 20 de out. de 2019.

WIESE *et al.* **Programação em blocos com Scratch. Aprendendo conceitos da engenharia de software e programação de forma divertida e interativa.** Criado por DACOM - Departamento de Computação Universidade Tecnológica Federal do Paraná. Atualizado em ago. 2018. Disponível em: <<https://www.udemy.com/course/programacao-em-blocos-com-scratch/>>. Acesso em: 03 de set. de 2019.

WING, J. M. **Computational thinking.** Communications of the ACM, v. 49, n. 3, p. 33-35, 2006.

XAVIER, G. F. C. **Lógica de Programação.** Editora: SENAC - São Paulo. 2007.

APÊNDICES

Apêndice I – Questionário

Questionário referente a Introdução de Lógica de Programação às crianças através da realização de um projeto social durante a disciplina de Gestão de Projetos. ³

- 1) Qual ferramenta foi utilizada no Projeto? (Scratch, *Code.org*, outras)
- 2) Em qual escola foi realizado? Descreva a infraestrutura da escola (laboratórios, internet...)
- 3) Qual o critério utilizado para escolha dos alunos que participaram do projeto? Qual a idade média?
- 4) Quantas horas / aulas foram ministradas?
- 5) Você possui algum conhecimento sobre *STEM*, e / ou Pensamento Computacional? Descreva-o.
- 6) Utilizou algum método para introduzir programação às crianças? Descreva-o. (“Era uma vez”, adequação do “Era uma vez”)
- 7) Encontrou dificuldades na aplicação do método? Quais?
- 8) Quais os pontos positivos que foram possíveis observar durante a aplicação do método?
- 9) Descreva como foi a experiência de realizar o projeto de Introdução a Programação às crianças. Faria novamente? Por quê? Acredita que as crianças tenham gostado?
- 10) Por que acredita que este tipo de ação não é adotada frequentemente pelas escolas?

³ Disponível em: < <https://forms.gle/Kp4hCoTiwhMp2Y3B9>>

Apêndice II – Compilado de respostas ao questionário

Compilado das respostas dos grupos ao questionário *online*

Grupos:

Grupo 1 - Ferramenta utilizada: *Code.org*. Método utilizado: Próprio.

O projeto foi realizado com duas turmas de 25 alunos cada, com média de idade entre 16 a 80 anos. Uma das turmas era composta por alunos portadores de necessidades especiais (autismo e síndrome de *down*). A outra turma era composta por alunos do EJA (Maiores de 40 anos, aprendendo a ler e escrever). Foram realizadas 8 aulas com cada turma, totalizando 16 horas aula. Não descreveram conhecimento no conceito de *STEM* e definiram Pensamento Computacional como resolução de problemas utilizando da ciência da computação. Por ter um público alvo diferente do proposto inicialmente e que apresentavam dificuldade com leitura e escrita, o grupo teve que se adequar e utilizou a primeira aula do *Code.org*, porém, posteriormente acabou apresentando aulas criadas pelo próprio grupo que se adequavam mais a necessidade dos alunos do EJA. Algumas aulas precisavam ser repetidas, pois os alunos tinham dificuldades em guardar os conteúdos. Os pontos positivos identificados pelo grupo foi que houve um grande aprendizado referente ao pensamento lógico tanto para as crianças de necessidades especiais quanto aos adultos. Tentaram ser o mais didático possível e por isso conseguiram atrair ainda mais a atenção dos alunos. O grupo relatou que foi uma experiência incrível, fantástica. Gostariam de fazer o projeto e com certeza fariam novamente. Acreditam que este tipo de ação não é realizado em mais organizações e escolas devido à falta de estrutura, organização e iniciativa das escolas e também pela falta de programas que ajudem nisto.

Grupo 2 - Ferramenta utilizada: jogos educativos (www.jogosdaescola.com.br; jogoseducativos.hvirtua.com.br). Método utilizado: Próprio, devido ao público diferente que tinham outras necessidades.

O projeto foi realizado com alunos acima de 30 anos. Foram realizadas 4 aulas de 2 horas cada, totalizando 8 horas aula. Não descreveram conhecimento no conceito de *STEM* e Pensamento Computacional como relacionado a resolução de problemas. A turma tinha um perfil um pouco diferente do proposto inicialmente para desenvolver o projeto, então o grupo não utilizou um método de introdução a programação exatamente. Utilizaram um método de introdução básico ao uso do computador, além de jogos educativos para exercitar português, matemática e principalmente a lógica. As dificuldades encontradas foram com alguns alunos que estavam mais no início do processo de alfabetização, porém o grupo estava sempre disposto a auxiliar no desenvolvimento dos mesmos. Os pontos positivos que o grupo apontou foi a

motivação em ver a evolução dos alunos no aprendizado de maneira diferente da tradicional, na qual utilizaram o computador como ferramenta principal e também o interesse dos alunos que viram o projeto como uma forma mais interessante de aprender. O grupo relatou que foi uma experiência única, muito bacana levar uma nova maneira de aprender àqueles alunos. E que ter a oportunidade de ajudar as pessoas assim foi muito interessante, pois puderam levar conceitos tecnológicos aos alunos que não tinham acesso a este tipo de coisa. Combinar o que os discentes buscavam (alfabetização) com tecnologia foi ainda mais bacana e gratificante. O grupo relatou que com certeza, fariam o projeto novamente. Acreditam que os alunos tenham gostado pela forma como receberam a notícia de ser a última aula (ficaram tristes). Até perguntaram quando teria mais vezes este tipo de projeto na escola. O grupo acredita que este tipo de ação não ocorre com frequência, pois faltam pessoas dispostas a dar um pouco do seu tempo para ajudar o próximo. Também por conta da metodologia tradicional que as escolas utilizam que ainda não estão adaptadas a realidade computacional. Está relacionado a não darem espaços para estes tipos de projeto nas escolas.

Grupo 3 - Ferramenta utilizada: *Scratch*. Método utilizado: “Era uma vez” (adaptado). Incluíram a utilização de um *Doodle* do *Google*, entre outros tipos de exercícios (*Code.org*, por exemplo).

O projeto foi realizado com alunos do 7º ano do ensino fundamental, com média de idade de 13 anos. Foram realizadas 4 aulas de 2 horas cada, totalizando 8 horas aula. Tentaram tornar os exercícios mais lúdicos ao relacioná-los com contos de fada (proposta do método “Era uma vez”. Contudo o grupo achou o material didático da metodologia um pouco limitado, por isso foi necessário realizar adaptações. Mesmo com as adaptações o grupo encontrou dificuldades na aplicação do “Era uma vez”, pois as crianças tinham a idade um pouco acima da recomendada, então acabaram perdendo um pouco a atenção das crianças que não se interessaram tanto pelo conto da Chapeuzinho Vermelho.

O grupo descreveu o conhecimento sobre *STEM* e Pensamento Computacional da seguinte forma:

“Algoritmos são ótimos para exercitar o raciocínio lógico do indivíduo.”

“Devemos incentivar as futuras gerações a desenvolverem pensamento crítico, analítico, pois é uma ferramenta muito útil e que pode ser implementada em qualquer área profissional.”

Os pontos positivos foram que os alunos puderam exercitar o raciocínio lógico e puderam ter uma experiência básica e uma noção de como funciona um programa (algoritmo). Os alunos puderam aprender se divertindo. O grupo conta que talvez realizaria novamente este tipo de projeto novamente, se fosse o mesmo fosse mais elaborado e tivesse mais versões do

“Era uma vez” (talvez adaptado para diferentes faixas etárias). Nas dificuldades encontradas o grupo reforçou que encontrou dificuldades na adaptação do método para as crianças que tinham uma idade acima da média proposta pela metodologia e acreditam que para as crianças com idade abaixo da sugerida também seria um pouco complicado a utilização. Isto não deveria acontecer, pois programação deve ser para todas as idades. Contudo, o grupo relatou que a experiência foi extremamente gratificante, principalmente no último dia quando as crianças disseram que iriam sentir falta das aulas. O impacto para o grupo como um todo foi muito prazeroso e enriquecedor. Acreditam que este tipo de ação não é frequentemente adotado nas escolas, pois há um desinteresse, acomodação por parte dos pais, alunos (narrow-minded persons), professores e da escola como um todo. Falta de conhecimento, iniciativa e divulgação deste tipo de projeto. E também por proporcionar uma dinâmica bem diferente do sistema tradicional de ensino que temos atualmente, acaba sendo difícil de implementar isto no dia a dia. O projeto é um grande passo, mas ainda falta muito para disseminar o conhecimento de uma maneira mais eficiente as crianças. Não sabem a complexidade de implementar projetos assim, mas acreditam que seria muito importante para desenvolver o conhecimento dos alunos.

Grupo 4 - Ferramenta utilizada: *Scratch*. Método utilizado: Próprio.

O projeto foi realizado com alunos de aproximadamente 11 anos. Foram realizadas 5 aulas de 2 horas cada, totalizando 10 horas aula. Não descreveram conhecimento no conceito de *STEM* e Pensamento Computacional como resolução de problemas. O grupo relatou que não foram encontradas dificuldades. O ponto positivo apontado foi com relação aos métodos diferentes de ensino que dão resultados mais eficientes. O grupo acredita que as crianças gostaram do projeto, contudo relataram que o método “Era uma vez” precisa ser atualizado e / ou alterado.

Grupo 5 - Ferramenta utilizada: *Code.org*, Método utilizado: Próprio.

O projeto foi realizado com crianças do 6º ano do ensino fundamental, com média de idade de 12 anos. A turma foi dividida em duas devido a quantidade de pessoas que o laboratório comportava, então foram realizadas 7 aulas com 1 hora de duração para cada uma das duas turmas, totalizando 14 horas aula. O grupo descreveu não descreveu conhecimento no conceito de *STEM*, já o Pensamento Computacional definiu como: “consiste em sistematizar problemas através de conceitos de computação e lógica (utilização da lógica para solucionar problemas)”. O grupo desenvolveu seu próprio método devido as crianças terem pouco contato com o computador. O método procurou abranger como ligar o computador, o que é um computador, digitação, comandos do teclado, a usar o *paint* e por fim apresentaram o *Code.org*. O grupo relatou que as dificuldades encontradas foram que as crianças faltaram em algumas aulas e por

isso tinham dificuldades posteriormente e que foi necessário adequar as atividades programadas inicialmente, houve alteração de cronograma e etc. Os pontos positivos: possível passar as crianças uma visão bem ampla de programação e todas conseguiram completar o desafio final. Conseguiram também estabelecer um nivelamento dos alunos que demonstravam estar motivados. Foi uma boa experiência, muito gratificante ver o desenvolvimento dos alunos no decorrer das aulas, descobrindo como solucionar problemas “brincando”. No geral, o grupo afirmou que faria novamente este tipo de projeto, porém alguns membros do grupo disseram que não fariam, pois é algo de muita responsabilidade. Acreditam que as crianças gostaram, pois no último dia houve uma despedida com abraços. Acreditam que este tipo de ação não ocorre com frequência devido à falta de profissionais para isso, pois sair do comodismo é complicado. Faltam também recursos a fim de tornar esta atividade padrão, pois as escolas públicas têm poucos recursos (financeiros, infraestrutura) e não investem em laboratórios e tecnologias. Não há também independência nas escolas para adotarem métodos de ensino diferentes dos propostos pelo MEC.

Grupo 6 - Ferramenta utilizada: *Code.org*. Método utilizado: Próprio.

O projeto foi realizado com crianças com média de 10 anos. Foram realizadas 7 aulas de 2 horas cada, totalizando 14 horas aula. Não descreveram conhecimento no conceito de *STEM* e nem em Pensamento Computacional. A dificuldade encontrada pelo grupo, foi em garantir o entendimento de todos alunos da turma. Já o ponto positivo abordado foi que alguns alunos se mostraram muito interessados na aula e aprenderam tudo que lhes foi passado. Ótima experiência que rendeu inclusive aprendizado ao grupo. Gostariam de fazer novamente, com toda certeza. Acreditam que as crianças tenham gostado das aulas, pois sempre estavam ansiosas para as próximas. Acreditam que este tipo de ação não ocorre com frequência nas escolas, devido à falta de pessoas disponíveis para ministrar este tipo de aula, falta também verba, espaço, recursos e etc.

Grupo 7- Ferramenta utilizada: *AppInventor*. Método utilizado: Próprio.

O projeto foi realizado com alunos (indicados pela coordenação) do 6º ano do ensino fundamental, com média de idade de 12 anos. Foram realizadas 4 aulas de 2 horas cada, totalizando 8 horas aula. Não descreveram conhecimento no conceito de *STEM* e Pensamento Computacional como algo que usa da computação para resolver problemas. O grupo utilizou um método próprio, no qual a primeira aula abordou apenas a parte teórica e posteriormente deram uma aula sobre o computador, por fim ocorreram as aulas práticas. O grupo encontrou dificuldade em prender atenção das crianças, pois elas queriam apenas aulas práticas. Houve também vários percalços como o agendamento da sala e algumas dificuldades de comunicação

entre o grupo e a escola. Porém durante a realização do projeto as crianças estavam empolgadas e com vontade de aprender. Os pontos positivos foram que levando a teoria primeiramente, as crianças puderam colocaram em prática com mais facilidade os conceitos aprendidos e assim conseguiram realizar todos exercícios propostos de forma mais eficiente. O projeto foi muito trabalhoso, porém recompensador, então o grupo apontou que o realizaria novamente. Acreditam que estes projetos não acontecem com frequência nas escolas por conta da comodidade, pois as escolas principalmente as públicas não têm a obrigação de cumprir mais do que é definido em sua grade. E apenas acreditar na mudança do poder público nunca foi a solução.

Grupo 8 - Ferramenta utilizada: *Scratch*. Método utilizado “Era uma vez” (adaptado), inclusão de mais exercícios introdutórios.

O projeto foi realizado com alunos entre 11 e 12 anos. Foram realizadas um total de 12 horas aula (10 horas + 2 horas de aula extra). Não descreveram conhecimento no conceito de *STEM* e nem em Pensamento Computacional. O grupo apontou que a dificuldade encontrada foi com relação a utilização do método “Era uma vez” devido as diferentes faixas etárias das crianças envolvidas. No geral apontaram que foi uma ótima experiência, cansativa, porém com o entusiasmo dos alunos aprendendo coisas novas de informática foi muito gratificante. Porém é um projeto que exige bastante dedicação de tempo e esforço. O grupo relatou que faria novamente o projeto, se o mesmo fosse mais estruturado ao longo dos anos, com mais acompanhamento. As crianças adoraram participar. Acreditam que estas ações não ocorrem com mais frequência nas escolas por falta de pessoas interessadas em trabalhar na inclusão digital das crianças.

Grupo 9 – Ferramenta utilizada: *Scratch*. Método utilizado “Era uma vez” (adaptado).

O projeto foi realizado com alunos entre 8 e 9 anos. Foram realizadas 5 aulas de 2 horas cada, totalizando 10 horas aula. Inicialmente o grupo iria utilizar o método “Era uma vez”, porém no decorrer do projeto foi necessário realizar algumas adaptações em aulas devido à idade das crianças. Posteriormente a adaptação o grupo não encontrou mais dificuldades na aplicação do método que se tornou até mais simples.

O grupo descreveu o conceito *STEM* como: “um método que busca despertar o interesse em profissões relacionadas à tecnologia, ciência, engenharia, matemática. Inserindo estes temas de forma prática no dia a dia.” Já o Pensamento Computacional usa métodos de resolução de problemas, com base em princípios computacionais, algoritmo e lógica. A dificuldade encontrada foi em prender a atenção das crianças nas explicações, principalmente das aulas teóricas. Os pontos positivos foram que as crianças trabalharam em duplas e a união fez com

que realizassem um bom trabalho em grupo. As crianças estavam empolgadas por estarem desenvolvendo o próprio jogo, por mais simples que fosse. Outro ponto positivo foi que algumas das crianças tinham mais facilidade e podem até mesmo desenvolver uma carreira na área. A experiência foi muito gratificante, pois despertar o interesse das crianças e poder ajudar a descobrirem opções de profissões no futuro foi incrível. Realizariam o projeto novamente, um dos integrantes do grupo relatou que até pensa em seguir na área de licenciatura, para se especializar em dar aula de programação as crianças. Acreditam que estes projetos não ocorrem com frequência por falta de interesse ou mesmo falta de iniciativa, pois como foi observado pelo grupo algumas crianças não sabiam que as coisas poderiam ser programadas. Ocorre também a falta de informações dos professores e especialistas sobre a importância do tema. Um dos membros presta suporte técnico aos computadores da prefeitura da cidade e relatou que é visível a falta de conhecimento básico de informática, tecnologia, o que evidencia ainda mais a falta de interesse em adotar esse tipo de medida nas escolas públicas.

Grupo 10- Ferramentas utilizadas: *Scratch*, *Code.org* e *kahoot.it*. Método utilizado “Era uma vez” (adaptado).

O projeto foi realizado com alunos entre 10 e 11 anos. Foram realizadas aproximadamente um total de 8 horas aula. O grupo descreveu o Pensamento Computacional como: “capacidade de formar uma linha de raciocínio lógica capaz de resolver problemas.” E o conceito *STEM* “integrar raciocínio entre ciência tecnologia, matemática e engenharia, tudo, pois o mercado exige uma multidisciplinaridade em todos sentidos de pensar, agir, estudar, etc. O *STEM* agrupa disciplinas.” Inicialmente, o grupo iria utilizar o método “Era uma vez”, porém devido as dificuldades encontradas na utilização do *Scratch*, optaram pela utilização dos cursos do *Code.org*. Os pontos positivos foram que puderam abordar diversos assuntos dentro da computação e do pensamento lógico e que os alunos progrediram muito durante as aulas. O *Code.org* é muito mais simples e traz conceitos mais básicos de programação para crianças, portanto acabou sendo mais ‘tangível’ a realidade dos alunos. O grupo relatou que a experiência foi incrível e maravilhosa e que com certeza realizariam o projeto novamente. Acreditam que as crianças gostaram muito, pois no final das aulas sempre perguntavam pelo retorno. As crianças sempre davam feedbacks que gostaram demais das aulas e não queriam que o curso acabasse. Um dos membros até relatou que gostaria de continuar com o projeto na cidade em que mora. Acreditam que este tipo de ação não ocorre com frequência devido as escolas ficarem presas ao currículo dos cursos que já vem definido, o importante para as escolas é apenas cumprir prazos deixando de lado projetos como este. Acreditam também que há falta de conhecimento e importância sobre este tipo de projeto. Entre várias outras questões,

principalmente a falta de informação por parte de docentes e coordenação das escolas para incentivar estas atividades. Para as crianças terem acesso a importância da programação desde o início era necessária uma expansão do projeto para muitas outras escolas, e isso demanda disponibilidade e interesse por parte de nós, alunos. Há pouco conhecimento dos benefícios que acarreta, não apenas as crianças, mas também para todos que participam da ação.

Grupo 11 - Ferramenta utilizada: *Scratch*. Método utilizado “Era uma vez” (adaptado).

O projeto foi realizado com alunos com média de 10 anos. Foram realizadas 4 aulas de 2 horas cada, totalizando 8 horas aula. Um dos membros do grupo relatou que possui conhecimento sobre o conceito *STEM* e Pensamento Computacional bem a fundo, pois deu aula de inglês particular e teve a oportunidade de desenvolver materiais mais práticos visando o cotidiano de seus alunos. Durante o projeto foi necessário fazer uma adequação no método “Era uma vez”, e acabaram não utilizando apenas o *Scratch*, mas também trazendo outros conteúdos interessantes que fossem úteis para o entendimento geral das aulas (por exemplo utilização do *Code.org*). O grupo não encontrou dificuldades, além da necessidade de adequação da metodologia de ensino. Os pontos positivos foram: o interesse dos alunos, dificilmente era necessário cobrar atenção. Os alunos eram participativos, o que motivou muito a equipe. Foi uma experiência incrível e motivadora. Ensinar é sempre bom, principalmente crianças que serão o futuro da sociedade. Fariam novamente (porque acreditam que poderiam transmitir muitos outros conhecimentos relacionados a computação para os discentes) e acreditam que a educação é a única forma de mudar o mundo e poder contribuir com isso é muito gratificante. As crianças com certeza gostaram, assim como mostra o feedback nos questionários que o grupo aplicou durante o projeto. Acreditam que este tipo de ação não ocorre com frequência porque o método educacional está ultrapassado na nossa sociedade e há falta de infraestrutura adequada para este tipo de projeto. O nosso país ainda tem um conceito de que apenas “jogar conteúdo” e esperar que as pessoas absorvam é eficaz. Uma vez que tal conteúdo é necessário para etapas específicas se cumprirem, por exemplo: vestibular, concursos públicos e etc, acham importante abordá-los, contudo isso precisa ser revisto. Enquanto não compreendermos que é necessário começar a mudar camada por camada da educação, inclusive mostrando quão importante é saber ensinar, não iremos evoluir.

Grupo 12 - Ferramenta utilizada: *Code.org*. Método utilizado: Próprio.

O projeto foi realizado com alunos do 4º ano com média de 9 anos. Devido a capacidade do laboratório a turma foi dividida em duas. Foram realizadas 6 aulas com cada turma de 1 hora, totalizando 12 horas aula. Não descreveram conhecimento no conceito de *STEM* e Pensamento Computacional como resolução de problemas que utilizam a computação para tal.

Foi necessário a adequação do método “Era uma vez”, então passaram a utilizar algumas aulas do *Code.org*, pois era mais simples. Outra adequação necessária foi a retirada total ou parcial de algumas explicações teóricas, devido a idade dos alunos ser abaixo da média recomendada. Além das dificuldades na adequação da metodologia o grupo não encontrou outras dificuldades. O ponto positivo foi realizar um projeto real e social, que foi muito gratificante, foi uma experiência ótima. As crianças gostaram muito e queriam que o projeto continuasse, mostraram estar ansiosas para que outros alunos da faculdade voltem lá nos próximos anos. O grupo gostaria de realizar o projeto novamente, pois o interesse das crianças em conceitos de computação foi muito interessante de ver. Acreditam que este tipo de ação não ocorre com frequência, pois a capacitação na área de pedagogia e informática são dispersas. Uma está totalmente desconexa da outra, limitando profissionais que possam fazer projetos como este.

Grupo 13 – Ferramenta utilizada: *VisualIG IDE (Portugol)*. Método utilizado: Próprio.

O projeto foi realizado com alunos do 5º ano com média de 10 anos. Foram realizadas 8 aulas com 2 horas cada, totalizando 16 horas aula. Devido a realização deste mesmo projeto nesta turma no ano anterior, onde os alunos já haviam aprendido lógica de programação por meio do método “Era o uma vez”. Este grupo levou um desafio ainda maior aos alunos e criaram um método próprio onde 30% das aulas eram teóricas e 70% práticas. Utilizaram o portugol para ensinar uma linguagem de programação estruturada. No método houve a aplicação de novos conceitos e sempre ocorria a revisão das aulas anteriores, afim de desenvolver cada vez mais o aprendizado dos alunos. Desenvolveram com as crianças uma calculadora e também uma tabuada, demonstrando através de aulas práticas as estruturas lógicas necessárias e no decorrer das aulas práticas foi possível observar o rápido progresso da turma. O grupo não descreveu o conceito de *STEM*, porém definiu Pensamento Computacional como “a lógica que é utilizada para resolução de problemas, como por exemplo, os algoritmos”. As dificuldades encontradas foram de atrair o interesse das crianças principalmente nas aulas teóricas e motivá-las a prestar a atenção. Os pontos positivos foi ver que os alunos conseguiram realizar as tarefas com facilidade, pois o portugol é muito próximo ao nosso idioma facilitando assim a compreensão de seus significados e também vê-los concluir as atividades práticas propostas. A experiência teve um resultado inesperado, pois inicialmente acreditavam que as crianças já teriam um bom conhecimento adquirido anteriormente na primeira aplicação do projeto, então acabaram tendo muita expectativa, contudo tiveram que passar conceitos mais básicos. Por conta disso achavam que as crianças não iam aprender muito. Todavia, no final o grupo foi surpreendido ao receber o comentário das crianças citando os conceitos ensinados. Apesar de tudo, alguns membros relataram que não fariam novamente, já outros, gostariam de realizar. As

crianças gostaram, pois, demonstravam estar empolgadas e pelo feedback que obtiveram da direção foi uma ótima experiência. Acreditam que estes trabalhos ocorrem apenas de maneira voluntária porque dificilmente as escolas oferecem cursos assim e os alunos das universidades nem sempre conseguem ministrar aulas fora do horário. Há também a falta de profissionais capacitados. De modo geral é necessário ter conhecimento básico para introduzir lógica de programação, sendo necessário algum tipo de capacitação ou treinamento.

Grupo 14 – Ferramenta utilizada: *Code.org*. Método utilizado: Próprio. O projeto foi realizado com alunos com média de 10 anos. Foram realizadas 4 aulas com 2 horas cada, totalizando 8 horas aula. Não descreveram conhecimento no conceito de *STEM* e nem em Pensamento Computacional. O grupo utilizou o método das aulas disponíveis no *Code.org*. Não encontraram dificuldades. Os pontos positivos relatados foram que a turma teve um rápido desenvolvimento e aprendizado com o método apresentado. O grupo relatou que faria novamente este tipo de projeto, pois foi uma experiência incrível mostrar algo novo as crianças e acreditam que para elas também deve ter sido. As crianças gostaram muito como mostra o feedback dado no questionário aplicado durante o projeto. A motivação do grupo foi o retorno positivo das crianças com relação as aulas dadas.

Acreditam que este tipo de ação não ocorre com frequência nas escolas, pois não há muita divulgação. Os custos são altos e depende muito de órgãos públicos aprovarem o que torna o processo falho. Há também a alta de interesse e de infraestrutura adequada das escolas. Projetos sólidos como este desenvolvido pelo LIAG da Unicamp se torna viável e interessante para colocar em prática este tipo de ação, porém nem todas universidades conseguem atuar desta forma. Seria necessária uma aceitação mais ampla por parte, principalmente, de órgãos públicos.

Grupo 15 - Ferramenta utilizada: *Scratch*. Método utilizado: “Era uma vez” (adaptado).

O projeto foi realizado com alunos com média de 8 anos. Foram realizadas 8 aulas com 1 hora cada para duas turmas (foi necessário a divisão das turmas por conta da capacidade do laboratório), totalizando 16 horas aula. Não descreveram conhecimento no conceito de *STEM* e Pensamento Computacional como uso da computação na tomada de decisão baseada na solução de problemas. Utilizaram o *Scratch Portable* já instalado nos computadores, sem a necessidade de utilização *online*. O grupo encontrou dificuldade na adaptação do método, pois as crianças tinham apenas 8 anos, média de idade bem abaixo da recomendada pelo método. Os pontos positivos abordados foram que as crianças gostaram bastante e puderam entender vários conceitos relacionados a lógica de programação. A experiência foi muito boa, diferente da rotina e o grupo relatou que foi bem recebido pelas crianças que estavam empolgadas e

gostaram muito. Acreditam que este tipo de ação não ocorre com frequência nas escolas, pois muitas não têm acesso a essa oportunidade, este tipo de projeto, ou mesmo a iniciativas assim.

Grupo 16 - Ferramenta utilizada: *Scratch*. Método utilizado: “Era uma vez” (adaptado).

O projeto foi realizado com alunos de média de 9 anos. Foram realizadas 6 aulas com 2 horas cada, totalizando 12 horas aula. Não descreveram conhecimento no conceito de *STEM* e nem em Pensamento Computacional. O grupo relatou dificuldades na adequação do método e na alfabetização de algumas das crianças. O maior ponto positivo identificado foi o interesse das crianças em aprender os conceitos de programação de uma maneira diferente da tradicional. A experiência foi bem legal para aprender conceitos de lógica e o grupo acabou também aprendendo lições de vida com as crianças. Acreditam que este tipo de ação não ocorre com frequência nas escolas, pois há uma falta de conhecimento destes projetos e também falta de professores qualificados.

Grupo 17 – Ferramenta utilizada: *Code.org* e atividades *off* usando papel para aprender alguns conceitos mais teóricos. Método utilizado: Próprio no qual abordaram o funcionamento dos computadores, sistema binário e até mesmo programação em blocos. Não encontraram dificuldades na aplicação deste novo método, as propostas funcionaram bem.

O projeto foi realizado com alunos de média de 11 anos. Foram realizadas 4 aulas com 1 hora e meia cada, totalizando 6 horas. O grupo não descreveu conhecimento no conceito de *STEM* e definiram o Pensamento Computacional como algo relacionado a programação e resolução de problemas. Os pontos positivos identificados foram a empolgação e empenho das crianças, durante não apenas nas atividades práticas, mas também nas atividades *off*. Foi muito gratificante e interessante a realização do projeto, pois era perceptível a evolução das crianças durante as atividades. Os que acabavam mais rápido ajudam os outros, foi muito bacana ver as próprias crianças incentivando o trabalho em grupo. A evolução e progresso das crianças era notável e o grupo ficou muito satisfeito com as aulas. Foi unanime entre os membros do grupo que seria interessante continuar com o projeto, mas é complicado por falta de tempo. As crianças com certeza gostaram muito. Acreditam que este tipo de ação não ocorra com frequência, pois demanda tempo, infraestrutura adequada, além de pessoas com conhecimento e disponibilidade para ensinar.

Apêndice III – Tabulação dos resultados

Tabulação das respostas dos grupos ao questionário *online*

Ferramentas	Grupos
<i>Scratch</i>	7
<i>Code.org</i>	6
AppInventor	1
Outras	3

Métodos	Grupos
Próprio	10
“Era uma vez” - ADAPTADO	7

Faria novamente ?	Grupos
Sim	16
Não	1

Idade	Grupos
8 à 10 anos	10
11 à 13 anos	5
Adultos	2

Tipo Adaptação	Grupos
<i>Code.org</i> / outra ferramenta	12
incluir conceitos básicos	3
excluir complexos	2

Horas / aulas	Grupos
8 a 10	9
11 a 14	3
15 a 16	5

<i>STEM</i>	Grupos
Sim	4
Não	13

Pensamento Computacional	Grupos
Sim	13
Não	4

Dificuldades
Prender a atenção nas aulas teóricas / Agendamento de salas, comunicação com a escola
Não
Não
Diferença na faixa etária
Leitura
Dificuldade das crianças com alguns conceitos mais complexos
Não
Prender a atenção nas aulas teóricas
Idade das crianças
Início do processo de alfabetização
Não
Garantir o entendimento dos conceitos abordados
Dificuldade de entendimento com a ferramenta <i>Scratch</i>
Crianças não tinham familiaridade com o Computador
Idade das crianças / Falta de atenção
Falta de atenção das crianças
Guardar conteúdo/ Leitura e escrita

Pontos Positivos
Conseguiram realizar as atividades com facilidade
Método diferente de ensino
Empolgação
Entusiasmo das crianças
Interesse dos alunos
Empolgação
Rápido desenvolvimento da turma
Perceber a evolução das crianças na aprendizagem
Interesse dos alunos
Evolução / Interesse dos alunos
Alunos participativos
Interesse dos alunos
Método abrangeu vários assuntos
Motivação das crianças em querer aprender
Empolgação
Exercitar o raciocínio lógico de forma diferente
Grande evolução no aprendizado de conceitos lógicos

Porque não é realizado com frequencia
Comodidade, desinteresse em fazer além do que o poder público obriga
Infraestrutura inadequada
Falta de profissionais , tempo
Falta de pessoas interessadas em trabalhar nessa inclusão digital das crianças.
Falta de profissionais
Falta de infraestrutura
Falta de projetos, divulgação
Falta de profissionais / voluntários / interesse
Falta de profissionais
Falta de profissionais / tempo / espaço nas escolas
Método educacional ultrapassado / Falta de infraestrutura / Necessário grande mudança na educação
Falta de profissionais / recursos / verbas
Escolas ficam presas a seus currículos antigos / Faltam de conhecimento das vantagens deste tipo de ação
Falta de profissionais / recursos / métodos antigos
Falta de iniciativa / interesse de pais , alunos e professores
Desinteresse / Falta de iniciativa
Falta de estrutura, organização e iniciativa das escolas, também de programas que ajudem nisto.