



Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Matemática
Programa de Mestrado Profissional
em Matemática em Rede Nacional



Usando o *Scratch* como ferramenta interdisciplinar através da programação

Sérgio Luís Soares Almeida

Brasília

2020

Sérgio Luís Soares Almeida

**Usando o *Scratch* como ferramenta interdisciplinar
através da programação**

Dissertação apresentada ao Departamento de Matemática da Universidade de Brasília, como parte dos requisitos do Programa de Mestrado Profissional em Matemática em Rede Nacional - PROFMAT, para obtenção do grau de Mestre.

Universidade de Brasília - UnB
Departamento de Matemática - MAT
PROFMAT - SBM

Orientadora: Profa. Dra. Tatiane da Silva Evangelista

Brasília
2020

Ficha catalográfica elaborada automaticamente,
com os dados fornecidos pelo(a) autor(a)

SS676u Soares Almeida, Sérgio Luís
Usando o Scratch como ferramenta interdisciplinar
através da programação / Sérgio Luís Soares Almeida;
orientador Tatiane da Silva Evangelista. -- Brasília, 2020.
48 p.

Dissertação (Mestrado - Mestrado Profissional em
Matemática) -- Universidade de Brasília, 2020.

1. Scratch. 2. Interdisciplinar. 3. Pensamento
Computacional. 4. Linguagem de Programação. I. da Silva
Evangelista, Tatiane, orient. II. Título.

Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Matemática

Usando o *Scratch* como ferramenta interdisciplinar através da programação

por

Sérgio Luís Soares Almeida

Dissertação apresentada ao Departamento de Matemática da Universidade de Brasília, como parte dos requisitos do Programa de Mestrado Profissional em Matemática em Rede Nacional - PROFMAT, para obtenção do grau de

MESTRE

Brasília, 28 de Julho de 2020

Comissão Examinadora:

Profa. Dra. Tatiane da Silva Evangelista- FGA/UnB (Orientadora)

Prof. Dr. Matheus Bernardini de Souza - FGA/UnB (Membro Interno)

Prof. Dr. Ronni Geraldo G. de Amorim - FGA/UnB (Membro Externo)

Dedico esse trabalho primeiramente a Deus, que sempre esteve ao meu lado nos momentos de angústia, à minha mãe, Maria de Jesus Soares, aos meus irmãos, Liana Soares e Marcelo Soares e ao meu filho Gabriel Rodrigues que sempre me incentivaram e ainda me incentivam a continuar na vida acadêmica.

Agradecimentos

Agradeço em primeiro lugar a Deus por ter me abençoado e iluminado o meu caminho, me mostrando quais as escolhas corretas e como a vida é maravilhosa. Agradeço a minha mãe, *Maria de Jesus Soares*, o meu maior exemplo de determinação. Agradeço aos meus irmãos *Marcelo Soares* e *Liana Soares* por sempre me apoiarem e me incentivarem.

Agradeço ao meu filho, *Gabriel Rodrigues*, que sempre esteve ao meu lado, me incentivando, me apoiando e me trazendo muita alegria. Obrigado por estar sempre presente em todos os momentos da minha vida.

Agradeço a todos os amigos da turma do PROFMAT 2018. Sempre lembrarei dos momentos maravilhosos que passamos juntos.

Agradeço à minha professora e orientadora, professora Tatiane, pela sua amizade, pelos conselhos, por ter acreditado no meu potencial e nesse trabalho.

Agradeço aos meus amigos *Clébia*, *Cláudio*, *Douglas*, *Fabiano*, *Mayco*, *Wagner* e *Rodrigo* por me mostrarem como a união pode superar qualquer dificuldade e como o estudo em grupo é essencial na vida acadêmica.

Agradeço a todos que sempre me acompanharam e sempre torceram por mim.

*“Educar verdadeiramente não é ensinar fatos novos ou enumerar fórmulas prontas, mas
sim preparar a mente para pensar ”*
Albert Einstein

Resumo

O computador ainda é utilizado na Educação apenas como meio de reprodução de atividades que anteriormente eram desenvolvidas por outros meios de comunicação como, por exemplo, exposição e criação de resumos, reprodução de trabalhos escolares e citação de trechos de livros através de *blogs* e *sites*. Os conhecimentos em computação podem ser usados de maneira que os estudantes não sejam apenas usuários das Tecnologias Digitais de Comunicação e de Informação, mas que também possam ser produtores de tecnologia. Com base na teoria do Pensamento Computacional, a inserção da computação na educação pode ser fator determinante na construção de novos conhecimentos e habilidades advindas do raciocínio lógico e da abstração de problemas. Desta forma, o *Scratch*, uma plataforma de programação idealizada por Mitchel Resnick para crianças a partir de oito anos de idade e criado em 2007 pelo *MIT* (*Massachusetts Institute of Technology*), pode ser usado para os estudos de conteúdos interdisciplinares com a inserção do estudo da computação em sala de aula e na construção de Objetos Digitais de Aprendizagem para a Educação Básica. Assim, em parceria com o Centro de Referência em Tecnologia Educacional do Recanto das Emas no Distrito Federal, foi oferecida uma formação do *Scratch* para professores da rede pública de ensino para incentivá-los a usar essa plataforma de maneira dinâmica e intuitiva. Em razão disso, o objetivo desse trabalho foi relatar este curso e observar como os professores e estudantes podem desenvolver projetos usando a computação e quais os benefícios que esse uso e esses projetos podem ajudar na relação ensino-aprendizagem e assim percebeu-se que o *Scratch* não só é uma ferramenta capaz de ser usada de forma interdisciplinar como também oferece condições para uma mudança no processo de aprendizagem e nas relações entre professores e estudantes, transformando o conhecimento em uma construção contínua.

Palavras-chaves: Interdisciplinar. Computação. Educação. *Scratch*.

Abstract

Computers are still used in Education only as a means of reproducing activities that were previously carried out by other means of communication, such as, for example, displaying and creating abstracts, reproducing schoolwork and citing excerpts from books through blogs and websites. Computer skills can be used in such a way that students are not only users of Digital Communication and Information Technologies, but can also be technology producers. Based on the Computational Thinking theory, the insertion of computing in education can be a determining factor in the construction of new knowledge and skills arising from logical reasoning and problem abstraction. Thereby, Scratch, a programming platform designed by Mitchel Resnick for children from eight years of age and created in 2007 by MIT (Massachusetts Institute of Technology), can be used for the study of interdisciplinary content with the insertion of study of computing in the classroom and in the construction of Digital Learning Objects for Basic Education. Thus, in partnership with the Educational Technology Reference Center of Recanto das Emas in the Federal District, Scratch training was offered to public school teachers to encourage them to use this platform in a dynamic and intuitive way. As a result, the objective of this work was to report the course in question and observe how teachers and students can develop projects using computing, in addition to extolling its benefits in relation to the teaching learning process. Thereby, it was realized that Scratch is not only a tool capable of being used in an interdisciplinary way, but also offers conditions for a change in the learning process and in the relationships between teachers and students, transforming knowledge into a continuous construction.

Key-words: Interdisciplinary. Computing. Education. Scratch.

Lista de ilustrações

1.	Slogan do <i>Scratch</i>	25
2.	Interface do <i>Scratch</i>	26
3.	Área de blocos de comandos do <i>Scratch</i>	26
4.	Área de comandos do <i>Scratch</i>	27
5.	Simulador de ecrã do <i>Scratch</i>	27
6.	Planejamento da movimentação do peixe	29
7.	Região 3	30
8.	Região 1	30
9.	Região 2	31
10.	Região 3	32
11.	<i>Lightbot</i>	35
12.	Exemplo de animação	36
13.	Jogo Soma e Subtração	38
14.	Programação da História de Aristóteles	40
15.	Palco e Programação da bola que muda de cor	41
16.	Palco da Condução de bola	41
17.	Palco do <i>Quiz</i> Reações e Balanceamento	42
18.	Palco e Programação da Receita de Pão	43
19.	Palco do jogo de Tabuleiro	44
20.	Programação do dado	44
21.	Programação dos participantes	45

Lista de tabelas

1.	Sugestões de inserção do PC nas disciplinas de Matemática e Ciências	20
2.	Sugestões de inserção do PC nas disciplinas de Estudos Sociais, Linguagens e Artes	21
3.	Planejamento do Curso sobre o <i>Scratch</i>	34

Sumário

1	INTRODUÇÃO	12
2	REFERENCIAL TEÓRICO	14
2.1.	Conexão: Construcionismo e Construtivismo	15
2.2.	Pensamento Computacional	16
2.3.	Objetos de Aprendizagem	21
3	APRESENTANDO O <i>SCRATCH</i>	25
3.1.	Exemplo de como programar no <i>Scratch</i>	28
4	METODOLOGIA DO CURSO	33
4.1.	Detalhamento dos encontros	34
4.2.	Principais projetos desenvolvidos	39
5	CONCLUSÃO	46
	REFERÊNCIAS	48

1 Introdução

As novas Tecnologias Digitais de Informação e de Comunicação causaram e ainda causam grandes transformações no trabalho, nas relações interpessoais e na sociedade em geral, mas essas transformações não são tão rápidas e tão profundas na educação, especialmente na Educação Básica. O computador, como ferramenta educacional, ainda é muito utilizado apenas como um meio de reprodução de atividades que já eram desenvolvidas anteriormente mudando apenas o meio de comunicação. A forma de apresentar os conteúdos e as aulas, em sua essência, progrediram pouco ou quase nada, com o professor ainda sendo o detentor do conhecimento e os estudantes com o objetivo de reproduzir esses conhecimentos que agora podem ser pesquisados nos *sites* e nos *blogs* escritos ou indicados pelo professor.

O uso das tecnologias na educação não pode servir apenas a uma proposta onde o computador passe alguma instrução ao estudante mas, principalmente, crie oportunidades para que o estudante possa programar e criar. É necessária uma nova abordagem no ensino, de forma que o estudante possa ser independente, que saiba procurar, selecionar e resolver os problemas ao seu redor usando as informações disponíveis e, desta forma, aprendendo com as situações apresentadas em seu meio social e cultural.

Nos últimos anos, alguns pesquisadores defendem que os conhecimentos em Computação sejam inseridos no ambiente escolar, dado que “...são tão importantes para a vida na sociedade contemporânea quanto os conhecimentos básicos de Matemática, de Filosofia, de Física, dentre outras, assim como contar, abstrair, pensar, relacionar ou medir” (BRACKMANN, 2017). Em contrapartida, a linguagem de programação, acrescenta ao estudante um raciocínio lógico aguçado capaz de entender essas novas tecnologias não apenas como um mero usuário, mas como um indivíduo com a habilidade de modificar e criar novos conhecimentos que podem ser aplicados em sua vida acadêmica, em seu trabalho e em seu cotidiano.

Diante desse cenário, o *Scratch* é uma ferramenta que propõe a introdução da linguagem de programação acessível e interdisciplinar dando a possibilidade ao estudante de criar seus próprios programas, jogos, histórias e apresentações e com a capacidade de desenvolver o raciocínio lógico de forma gradual e sistemática facilitando a resolução de problemas mais complexos e mais abrangentes.

Em parceria com o Centro de Referência em Tecnologia Educacional (CRTE) do Recanto das Emas - Distrito federal e sentindo também a necessidade de inserir o *Scratch* na sala de aula com base no Currículo de Educação Básica, foi oferecido uma formação com duração de 21 horas diretas e 9 indiretas para professores da rede pública de ensino

de diferentes formações acadêmicas e de diferentes modalidades e funções. Os CRTE são setores da Secretaria Estadual de Educação criados em todas as regiões administrativas do Distrito Federal que oferecem formações variadas e tem como principal atribuição contribuir para a melhoria da qualidade do ensino e a operacionalização do Currículo de Educação Básica, por meio de implementação, acompanhamento e utilização pedagógica das Tecnologias Digitais de Informação e de Comunicação junto às unidades escolares da Rede Pública de Ensino do Distrito Federal. Desta forma pôde-se também estudar de que forma a programação através do *Scratch* realmente contribui na aprendizagem e na compreensão dos conteúdos de forma interdisciplinar.

Outros objetivos inerentes aos estudos realizados nesse trabalho são:

- Analisar como o *Scratch* contribui para a inserção de conteúdos referentes ao currículo escolar.
- Observar como os professores, em suas determinadas áreas, conseguem utilizar os conceitos de linguagem de programação e algoritmos em sua prática pedagógica.
- Observar, através de depoimentos dos professores e da apresentação dos trabalhos realizados pelos seus estudantes, o desenvolvimento construtivista e construcionista gerado pela inserção do *Scratch*.

Para um melhor entendimento, esse trabalho foi dividido de tal forma que um capítulo é destinado a apresentação dos estudos que servem de embasamento teórico para o curso, outro capítulo tem como objetivo apresentar o que é o *Scratch* de forma geral e como ele foi criado e em mais um capítulo será explicada a metodologia do curso com a descrição dos estudos realizados em cada encontro e com os resultados obtidos e, após isso, apresenta-se a conclusão dos estudos realizados.

2 Referencial Teórico

A sociedade não precisa mais esperar o horário fixo de um determinado programa, de um determinado noticiário, ou até mesmo de um jornal impresso ou de um livro. As novas Tecnologias Digitais de Informação e de Comunicação apresentam um novo modelo de espaço e tempo, proporcionando a qualquer pessoa a escolher qual o momento ideal para ter acesso a uma notícia, qual o melhor lugar para assistir a um filme ou série, qual o passatempo a ser acessado, dessa forma cada indivíduo faz a sua programação em seu tempo e em seu espaço, acessando as notícias e as informações que lhes sejam interessantes e descartando o que não o interessa. Esse novo modelo de relação das novas Tecnologias Digitais de Informação e de Comunicação com o tempo e espaço ainda não foram totalmente assimiladas pela escola.

“Estas tecnologias começam a afetar profundamente a educação. Esta sempre esteve e continua presa a lugares e tempos determinados: escola, salas de aula, calendário escolar, grade curricular”(MORAN, 2013).

Em geral, a escola continua a ser uma instituição que, apesar do avanço nas Tecnologias Digitais de Informação e de Comunicação, mantém os mesmos conceitos de ensino focados no professor e no conteúdo tendo como finalidade principal o ensino em vez da aprendizagem.

“As tecnologias chegaram na escola, mas estas sempre privilegiaram mais o controle a modernização da infraestrutura e a gestão do que a mudança”(MORAN, 2013).

Ao mudar o foco do ensino para a aprendizagem, a escola deixa de ser o local exclusivo na construção do conhecimento e o estudante conquista autonomia como cidadão e como agente do próprio conhecimento levando em consideração as informações apresentadas, refletindo e relacionando-as ao tempo e espaço escolhidos. Neste processo, o professor deixa de ser o detentor do conhecimento e passa a ser o mediador da aprendizagem, com o papel de facilitador na organização e criação do conhecimento de forma sólida e multidisciplinar.

2.1. Conexão: Construcionismo e Construtivismo

Um dos primeiros a estudar e a defender o uso da linguagem de programação nas escolas foi o Professor Seymour Papert, do *Massachusetts Institute of Technology (MIT)*, através da linguagem LOGO. Ele percebeu que muitas escolas usam o computador para ensinar as crianças através de instruções e desta forma “o computador está sendo usado para programar a criança” (PAPERT, 1985). Para ele

“... é a criança que deve programar o computador e, ao fazê-lo, ela adquire um sentimento de domínio sobre um dos mais modernos e poderosos equipamentos tecnológicos e estabelece um contato íntimo com alguma das ideias mais profundas da Ciência, da Matemática e da Arte de construir modelos intelectuais” (PAPERT, 1985).

Para isto é introduzida a ideia de que a criança é como um construtor que necessita de materiais para a suas obras.

Na teoria de PIAGET, ao nascer a criança traz conhecimentos hereditários ligados à sobrevivência, como os reflexos de preensão e sucção (pegar e sugar), o da audição e visão (ouvir e olhar na direção), a visão e o tato (olhar e pegar) e, a partir destes conhecimentos a criança constrói novos conhecimentos, isto é, um conhecimento é construído a partir de outros cumulativamente e os primeiros são os conhecimentos hereditários. Assim,

“O sistema cognitivo a ser construído necessariamente é um prolongamento do sistema biológico e a sua construção está essencialmente ligada à sobrevivência do indivíduo, pois a construção do conhecimento, na sua origem, parte da necessidade de saber fazer ou ter os instrumentos necessários para a sobrevivência independente” (OSTI, 2009).

O desenvolvimento passa por quatro etapas: o período sensório-motor, o período pré-operatório, o período operatório concreto e o período operatório-formal. No período sensório-motor, anterior a linguagem, que vai do nascimento até aproximadamente os dois anos, as percepções sensoriais e esquemas motores da criança são construídos a partir de reflexos inatos (sucção e preensão) usados pelo bebê para lidar com o ambiente. No período pré-operatório, dos dois aos sete anos, a linguagem é um fator importante no desenvolvimento, pois a partir dela os acontecimentos passam a ser inseridos no quadro conceitual e racional e assim a criança torna-se capaz de reproduzir situações não apenas atuais. No período operatório concreto, em média dos sete aos doze anos, a criança passa a se interessar pelo grupo social, ampliando o campo da socialização e inserindo-se em outros grupos como o de amigos da escola, do clube, da rua, entre outros. O pensamento torna-se crítico apoiando-se nas operações mentais marcadas pela reversibilidade, revelando a presença de estruturas lógicas. No período operatório-formal, que se constitui a partir dos doze anos (em média), o jovem passa agora a discutir, questionar os valores dos pais e

construir os seus próprios. Torna-se mais consciente de seu próprio pensamento e consegue refletir sobre ele.

Para a aprendizagem construtiva piagetiana, o estudante, antes de ir à escola, já apresenta conhecimentos que foram obtidos através dos materiais fornecidos pelo meio cultural onde vive, o que PAPERT chama de “aprendizado sem ensino”. Desta forma, aprender a falar, aprender a geometria necessária para se deslocar e aprender a conversar logicamente e a dar respostas aos pais são exemplos dessa aprendizagem. Nesse cenário, o desenvolvimento mais lento de um conceito seria explicado por PIAGET através da maior complexidade ou da formalidade necessários para a assimilação desse conceito mas, para PAPERT, o fator crítico seria “a relativa pobreza do meio cultural em materiais que tornariam o conceito simples e concreto” ou ainda “o meio cultural pode fornecer materiais mas bloquear o seu uso” (PAPERT, 1985) onde, por exemplo, uma pessoa pode deixar de aprender qualquer coisa que culturalmente reconheçam como Matemática, mas que não tenham dificuldades matemáticas quando não o reconhecem como tal.

Piaget diferencia o pensamento “concreto” do pensamento “formal”. Quando a criança ingressa no primeiro ano escolar, aos seis anos, o pensamento concreto já está em formação e é fortalecido nos próximos anos. O pensamento formal se desenvolve por volta dos doze anos e “como sugerem alguns pesquisadores, algumas pessoas nunca desenvolvem o pensamento formal de maneira completa”(PAPERT, 1985). Desta forma, ao oferecer à criança materiais, mesmo que “artificiais”, para que possa construir o seu próprio conhecimento permite que a transição do pensamento concreto para o pensamento formal tenha menos obstáculos. É nesse contexto que o computador pode ser o elo entre o concreto e o formal, fornecendo os materiais necessários para a construção do pensamento, pois conhecimentos que só eram acessíveis através de processos formais podem ser abordados concretamente através do estudo da sua lógica de programação.

2.2. Pensamento Computacional

O termo “*computational thinking*” (pensamento computacional) passou a ser mais discutido a partir do artigo de Jeannette M. Wing em 2006, onde afirma que “o pensamento computacional baseia-se no poder e nos limites dos processos de computação, sejam eles executados por um ser humano ou por uma máquina”(WING, 2006). Ainda, segundo Wing:

“O pensamento computacional é uma habilidade fundamental para todos, não apenas para cientistas das computação. À leitura, escrita e aritmética, devemos acrescentar o pensamento computacional à capacidade analítica de cada criança” (WING, 2006).

No entanto o termo pensamento computacional não é definido em termos precisos.

Em dois *workshops* sobre o âmbito e a natureza do pensamento computacional, patrocinados pela *National Academy of Sciences* dos Estados Unidos da América, em 2009 e 2011, a *National Research Council* publicou:

“Os debates realizados no Workshop de fevereiro de 2009 não chegaram a um acordo geral entre os participantes sobre o conteúdo preciso de pensamento computacional, e muito menos a sua estrutura. No entanto, a falta de desacordo explícito sobre seus membros poderia ser entendida como refletindo uma intuição compartilhada entre os participantes do workshop que o pensamento computacional, como um modo de pensamento, tem o seu próprio caráter distintivo” (NATIONAL RESEARCH COUNCIL, 2011).

As organizações *International Society for Technology in Education* (ISTE) e a *American Computer Science Teachers Association* (CSTA) trabalharam com pesquisadores da Ciência da Computação e das áreas de Humanas e construíram uma definição para o pensamento computacional pautada em elementos objetivos que pudesse nortear as atividades realizadas na Educação Básica identificando os seguintes conceitos: coleta de dados, análise de dados, representação de dados, decomposição do problema, abstração, algoritmos, automação, paralelização e simulação. Assim o pensamento computacional foi definido como:

“Pensamento computacional é um processo de solução de problemas que inclui (mas não está limitado a) as seguintes características: Formular problemas de uma forma que nos permita usar um computador e outras ferramentas para ajudar a resolvê-los; Organizar e analisar logicamente os dados; Representar dados através de abstrações, como modelos e simulações; Automatizar soluções por meio de pensamento algorítmico; Identificar, analisar e implementar possíveis soluções com o objetivo de alcançar a combinação mais eficiente e eficaz de etapas e recursos; Generalizar e transferir esse processo de solução de problemas para uma ampla variedade de problemas” (ISTE/CSTA, 2011).

Estas características podem ser melhor compreendidas ao fazer um estudo mais detalhado de cada uma delas:

- **Formular problemas de uma forma que nos permita usar um computador e outras ferramentas para ajudar a resolvê-los**

Ao enfrentar situações problemas, as primeiras análises a serem feitas são: como podemos resolvê-las, que ferramentas podem ser utilizados ou ainda quem é o mais qualificado a resolvê-las. Ao encontrar respostas as essas análises, pode-se definir quais estruturas devem ser usadas para cada situação problema.

- **Organizar e analisar logicamente os dados**

A situação problema inicial será subdividida em situações problemas menores de tal forma que a análise dos dados determinará a ordem em que cada uma dessas subdivisões deverão ser resolvidas.

- **Representar dados através de abstrações, como modelos e simulações**

Com a subdivisão da situação problema inicial, verificar quais características são repetidas em cada subdivisão e ao reconhecer esses padrões destacar o que é mais relevante para a construção de modelos e para se fazer abstrações relativas para definir melhor a solução.

- **Automatizar soluções por meio de pensamento algorítmico**

Criar instruções claras, detalhadas de como cada subdivisão da situação problema deve ser resolvida e assim criar algoritmos de soluções que podem ser traduzidos e replicados quantas vezes forem necessários.

- **Identificar, analisar e implementar possíveis soluções com o objetivo de alcançar a combinação mais eficientes e eficaz de etapas e recursos**

Neste estágio, a análise é feita com base nas instruções dadas, verificando se existem redundâncias, se existem soluções mais práticas e eficazes para cada subdivisão e se a solução é realmente viável.

- **Generalizar e transferir esse processo de solução de problemas para uma ampla variedade de problemas**

Com a solução encontrada, analisar se a estrutura encontrada pode ser estendida para outras situações problemas semelhantes, aumentando assim a abrangência de todo o processo.

Os pesquisadores do ISTE/CSTA observam ainda que essas habilidades são “apoia-das e reforçadas por uma série de disposições ou atitudes que são dimensões essenciais do pensamento computacional” assim como “confiança em lidar com a complexidade, persistência em trabalhar com problemas difíceis, tolerância para a ambiguidade e capacidade de lidar com problemas abertos” (ISTE/CSTA, 2011).

Essas características deixam transparecer que o pensamento computacional é uma habilidade que qualquer pessoa deveria ter, independentemente da área de conhecimento ou atividade profissional, já que não envolve apenas os conceitos da Computação pois também agrega práticas de projetar sistemas, entender o comportamento humano e o pensamento crítico (WING, 2006).

Não devemos confundir “Pensamento Computacional” com “Alfabetismo Digital”, pois este último é a simples aptidão de manusear aplicativos eletrônicos. As novas gerações que ingressam hoje na escola e que estão imersos na linguagem digital dos computadores,

vídeo games e da Internet são considerados “nativos digitais”, mas Mitch Resnick (2012) questiona a fluência digital destes jovens e explica que embora eles tenham nascido em um ambiente tecnológico e demonstrem muita experiência e facilidade em interagir com tecnologias, não tem esta mesma habilidade para criar e se expressar utilizando essas novas tecnologias, comparando este jovem com alguém que sabe ler, mas não sabe escrever.

O desenvolvimento do pensamento computacional nos jovens exige que a formação do professor também passe por mudanças, não apenas reproduzindo os conhecimentos através de aplicativos, mas também resolvendo problemas através das tecnologias. Como disse Brackmann (2017):

“A integração das TICs no currículo e na aprendizagem no campo da sua formação inicial é um desafio, pois durante o processo de tornar o pensamento computacional como uma competência fundamental para o século XXI, os professores também devem obrigatoriamente se aproximar dessas tecnologias e refletir dentro da sua área como a diversidade de conceitos, teorias, modelos e práticas podem ser aplicados dentro de sua disciplina de forma abrangente” (BRACKMANN, 2017).

O Pensamento Computacional pode e deve ser inserido na Educação Básica através das disciplinas já existentes e que estão de acordo com a Base Nacional Comum Curricular - BNCC, de tal forma que o seu uso possa ser incentivado em várias disciplinas para que atinja um público cada vez maior. O CSTE/ISTE (2009) também realizou um levantamento com diversas propostas de adoção de forma Pluri, Multi e Transdisciplinar, de forma que BRACKMANN (2017) as combinou através de duas Tabelas.

A Tabela 1 mostra sugestões de inserção do Pensamento Computacional (PC) nas disciplinas de Matemática e Ciências.

Tabela 1 – Sugestões de inserção do PC nas disciplinas de Matemática e Ciências

Conceitos de PC	MATEMÁTICA	CIÊNCIAS
Coleção de Dados	Encontrar uma fonte de dados de uma experiência, por exemplo: cara ou coroa, lançar dados	Coletar dados de um experimento
Análise de Dados	Contar a ocorrência de jogadas, lançamentos de dados e análise de resultados	Analisar dados de um experimento
Representação de Dados	Utilizar gráfico de barras e de pizza para representação de dados. Usar conjuntos, listas, representações gráficas, etc. para a visualização de informações	Resumir dados de um experimento
Decomposição de Problemas	Aplicar ordem de operadores	Realizar uma classificação de espécies
Abstração	Usar variáveis na álgebra. Estudar funções de álgebra através de comparação em computadores.	Construir um modelo de uma entidade física
Algoritmos e Procedimentos	Realizar divisões longas, fatorar	Criar um procedimento experimental
Automação	Utilizar ferramentas como: Geometer, Sketch Pad, Star Logo, linhas de código em Python, etc.	Usar simulação de dados
Paralelismo	Resolução de sistemas lineares. Multiplicação de matrizes.	Realizar experimentos com diferentes parâmetros simultaneamente.
Simulação	Desenhar o gráfico de uma função em um plano cartesiano e modificar os valores das variáveis	Simular os movimentos do Sistema Solar

Fonte – BRACKMANN, 2017

Na Tabela 2, BRACKMANN (2017) mostra que o Pensamento Computacional também pode ser aplicado em outras disciplinas, com sugestões de inserção nas disciplinas de Estudos Sociais, Linguagens e Artes.

Tabela 2 – Sugestões de inserção do PC nas disciplinas de Estudos Sociais, Linguagens e Artes

Conceitos de PC	ESTUDOS SOCIAIS	LINGUAGENS E ARTES
Coleção de Dados	Estudar estatísticas de guerras ou dados populacionais	Identificar padrões em diferentes tipos de frases
Análise de Dados	Identificar as tendências dos dados estatísticos	Representar padrões de diferentes tipos de frases
Representação de Dados	Resumir e representar tendências	Escrever um rascunho
Abstração	Resumir fatos. Deduzir conclusões dos fatos	Uso de metáforas e analogias. Escrever uma história com diversas vertentes
Algoritmos e Procedimentos	-	Escrever instruções
Automação	Usar planilhas eletrônicas	-
Paralelismo	-	Utilizar o corretor ortográfico
Simulação	Incentivar com jogos que utilizem bases históricas	Encenação de uma história

Fonte – BRACKMANN, 2017

Assim percebemos, através das Tabelas 1 e 2 e de como as disciplinas se relacionam, como o desenvolvimento do Pensamento Computacional nos jovens pode ser importante na aprendizagem de qualquer disciplina desde que usado corretamente e com objetivos claros e bem definidos. Desta forma a inserção do Pensamento Computacional em sala de aula irá proporcionar a construção de novos conhecimentos e de novas abordagens para problemas do nosso dia-a-dia.

2.3. Objetos de Aprendizagem

Os Objetos de Aprendizagem podem ser um grande aliado do professor como ferramenta de aprendizagem e de revisão de conceitos em vários conteúdos e propostas de ensino. A metodologia e o planejamento de suas aplicações são fatores determinantes para o sucesso da aprendizagem e para o desenvolvimento de estratégias para resolução de problemas, o seu uso deve estar previamente ligado aos objetivos do conteúdo e da

aprendizagem que se quer alcançar. Podem ser de qualquer mídia ou formato, como uma apresentação em *stop motion* (animação quadro-a-quadro) ou uma simulação.

Não há um consenso sobre o conceito de Objetos de Aprendizagem, cada autor define de acordo com uma concepção própria levando em consideração a utilidade e importância do Objeto para o ensino e a aprendizagem. Desta forma, é importante que o conceito de um Objeto de Aprendizagem esteja de acordo com o objetivo do estudo e do planejamento do momento de aprendizagem. Entretanto, uma definição de Objetos de Aprendizagem significativa é dada por Tarouco (2003):

“Um Objeto de Aprendizagem é qualquer recurso, suplementar ao processo de aprendizagem, que pode ser reusado para apoiar a aprendizagem, termo geralmente aplicado a materiais educacionais projetados e construídos em pequenos conjuntos visando a potencializar o processo de aprendizagem onde o recurso pode ser utilizado”(TAROUCO; FABRE; TAMUSIUNAS, 2003).

Nesta definição devem ser destacadas as palavras “recurso”, “suplementar” e “reusado”, pois um Objeto de Aprendizagem não deve ser o objeto da aprendizagem mas um recurso que o estudante possa utilizar para ajudá-lo a alcançar a aprendizagem do conteúdo previamente exposto dentro dos objetivos pretendidos e que podem ser reutilizados não só para um conteúdo específico mas também em conjunto com outros Objetos de Aprendizagem que podem ser combinados e recombinados dentro de um mesmo contexto e com conteúdos relacionados.

A construção de um objeto de aprendizagem envolve um planejamento prévio que possa responder às perguntas:

a) Quais os objetivos a serem alcançados?

Os professores ao planejarem suas aulas ou atividades de ensino e aprendizagem definem que objetivos querem alcançar, mas esses podem ser diferentes dos pretendidos pelos idealizadores e desenvolvedores de determinados objetos de aprendizagem já conhecidos. Desta forma, o professor precisará adaptar o conteúdo educacional digital, podendo modificar a abordagem, combinar, adaptar ou mesmo criar outros objetos de aprendizagem. Esta é exatamente a proposta de construção de um conteúdo digital baseado em objetos de aprendizagem, que são desenvolvidos para compor ou produzir novos conteúdos educacionais digitais de ensino.

b) Qual o público alvo, suas habilidades, seu conhecimento prévio?

Quando são definidos os objetivos a serem alcançados por determinado objeto de aprendizagem, deve-se levar em consideração não só o conteúdo educacional digital, mas também é importante que este esteja adequado ao público alvo a que este se destina. As habilidades e o conhecimento prévio constituem pré-requisitos

necessários à aprendizagem e podem ser imprescindíveis ao conteúdo educacional digital tornando assim o objeto de aprendizagem mais eficaz ou menos eficaz em seu propósito. Mas deve-se também considerar que todo indivíduo traz consigo seus próprios conhecimentos e suas próprias habilidades, mais adequadas a algumas áreas específicas, mas isto não impede o seu desenvolvimento em outras áreas. Essas diferenças individuais e cognitivas e esses modos próprios de processar informações devem ser objetos de reflexão do professor ao elaborar qualquer atividade.

c) Qual a interface a ser usada?

A forma em que se apresenta esse objeto de aprendizagem é outro importante aspecto a ser considerado. A interface desse objeto de aprendizagem será responsável pela interação entre o público alvo e o seu próprio aprendizado e desta forma deve estar adequado aos seus conhecimentos prévios e aos objetivos do professor. Uma interface extremamente intuitiva pode ser importante para um público alvo com pouca experiência em computadores, por exemplo, mas também poderá ser tão fácil para quem tem um pouco mais de experiência que o aprendizado desejado pode ser relegado a um segundo plano. O contrário também pode acontecer, a interface ser extremamente complicada que pode desestimular o uso do objeto de aprendizagem. A escolha desta interface deve também levar em conta qual a tecnologia digital de informação e comunicação que será utilizado, por exemplo, um *smartphone*, um computador, ou simplesmente uma TV.

d) Quais ferramentas serão utilizadas para a sua construção?

Em muitos casos pode-se obter um objeto de aprendizagem em repositórios como o Banco Internacional de Objetos Educacionais (BIOE), podendo também reagrupá-los ou combiná-los, mas em outros casos precisa-se criar um objeto que mais se adequa ao público alvo. Para construir um objeto de aprendizagem, o professor necessita ter o conhecimento da área de domínio e das estratégias pedagógicas a serem utilizadas e essa construção pode ser facilitada com o uso de ferramentas de autoria, que são recursos mais amigáveis e intuitivos que desenvolvimentos de programas tradicionais. Existem várias ferramentas gratuitas que podem ajudar na construção de objetos de aprendizagem, como o Picasa, o Hot Potatoes, o Geogebra e, é claro, o *Scratch*.

e) O Objeto de Aprendizagem tem relação com mais de um conteúdo e pode ser re combinado e reutilizado?

Uma das características mais importantes de um objeto de aprendizagem é o fato de poder ter relação com mais de um conteúdo e de também poder ser re combinado e reutilizado. A sua utilização em diferentes contextos de aprendizagem depende também da adaptação do conteúdo e da abordagem que o professor dará em relação a este conteúdo. Os objetos de aprendizagem reutilizáveis podem ser só para instrução

ou uma combinação de instrução e interação e ainda podem “ter uma lógica e uma estrutura para gerar e/ou combinar com outros objetos de aprendizagem, permitindo ou não avaliar as interações dos estudantes com esse objeto” (TAROUCO; FABRE; TAMUSIUNAS, 2003).

Ao responder essas perguntas os objetos de aprendizagem construídos serão mais eficientes e com maior capacidade de serem reutilizados e recombinaados com outros conteúdos.

3 Apresentando o *Scratch*

O *Scratch* é uma plataforma de programação feita por blocos lógicos capaz de integrar sons e imagens em histórias interativas, jogos e animações entre outros programas interativos desenvolvidos pelo usuário de forma intuitiva e sem o conhecimento prévio de outras linguagens de programação. Foi idealizado por Mitchel Resnick para pessoas com idades a partir de oito anos e criado em 2007 pelo Lifelong Group, no *MIT (Massachusetts Institute of Technology) Media Lab*, onde continua sendo desenvolvido e moderado. Hoje é usado em mais de 150 países e está disponível em mais de 40 idiomas. É importante ressaltar que esse programa possui versões *on-line* e *off-line*, isto é, pode ser usado através de um navegador de Internet no caso *on-line* e assim o usuário pode acessar todos os programas desenvolvidos na plataforma de onde estiver ou, em caso de não possuir Internet, pode instalar em seu computador e usá-lo em sua máquina local para desenvolver seus projetos.

O *Scratch* é bastante acessível já que usa uma interface gráfica em que os programas são construídos através de blocos lógicos auto-encaixantes lembrando a montagem de um quebra-cabeça e possibilita aos usuários resolver problemas usando a criatividade, o raciocínio lógico e o pensamento computacional de forma lúdica, divertida e também colaborativa, pois seus projetos e programações podem ser compartilhados. O seu slogan é baseado em três princípios: imagine, programe e compartilhe.

Figura 1 – Slogan do *Scratch*



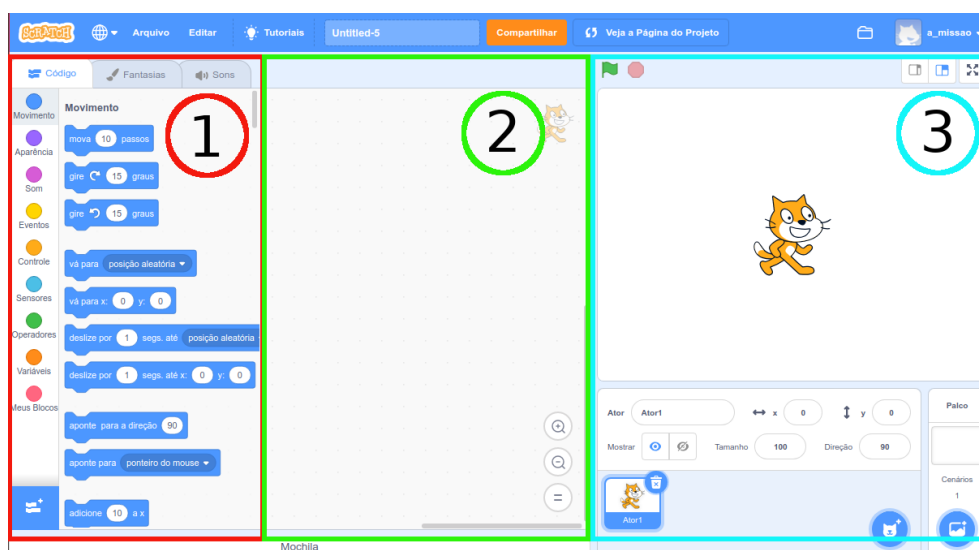
Fonte – Site do *Scratch* (<https://scratch.mit.edu/>)

O *Scratch* permite desenvolver jogos, animações e textos interativos, integrando, desta forma, diversas áreas do conhecimento de forma divertida. Assim os usuários podem imaginar uma história ou um jogo e, depois de brincarem e aprenderem com a programação ainda podem compartilhar com colegas e com outros usuários da plataforma suas criações, tornando-se mais criativas e estimuladas a procurarem novas formas de resolverem problemas através da programação e do Pensamento Computacional.

A sua interface é de fácil compreensão (vide Figura 2) e o uso de suas ferramentas muito intuitivas e sem complicações, possuindo três principais áreas:

1. Blocos de comando, organizados a partir da sua funcionalidade.
2. A área de comandos a serem seguidos, onde os blocos serão encaixados.
3. Palco ou simulador do Ecrã, onde se pode ver a simulação ou execução dos comandos.

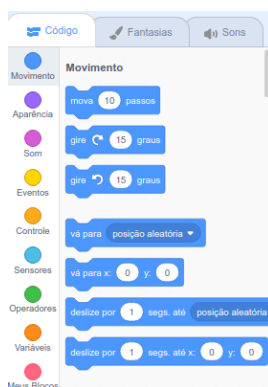
Figura 2 – Interface do *Scratch*



Fonte – Adaptado do Site do *Scratch*

A primeira área, ou área de blocos de comando (vide Figura 3), estão os comandos que podem ser usados na programação a ser realizada. Esses comandos estão subdivididos em categorias a partir de suas funcionalidades, como movimento, aparência, som, etc.

Figura 3 – Área de blocos de comandos do *Scratch*

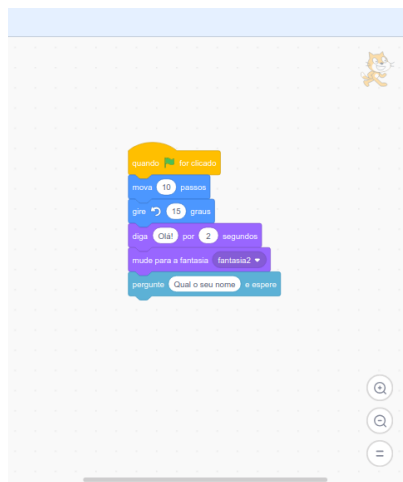


Fonte – Adaptado do Site do *Scratch*

Nesta área também aparecem as abas de fantasias e sons que podemos usar do próprio *Scratch* ou de nossos arquivos pessoais. Note que para cada categoria de blocos de comandos existe uma legenda de cores e todos os blocos relacionados a essas categorias tem a mesma cor de suas respectivas legendas.

A segunda área, chamada área de comandos (vide Figura 4), é a área onde os blocos são arrastados com o mouse e encaixados de maneira lógica para formar a programação a ser seguida pelos atores escolhidos.

Figura 4 – Área de comandos do *Scratch*

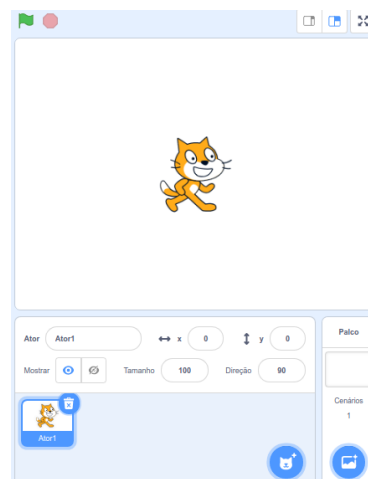


Fonte – Adaptado do Site do *Scratch*

É nessa área que todas as programações são montadas como um quebra-cabeça e cada ator terá sua área de comandos específica.

A terceira área (vide Figura 5) é onde se encontra o simulador do Ecrã (tela onde será projetada a programação), a relação dos atores e o palco com seus cenários.

Figura 5 – Simulador de ecrã do *Scratch*



Fonte – Adaptado do Site do *Scratch*

O Ecrã é “mapeado” através de coordenadas cartesianas e o primeiro ator (também chamado de *sprite*) que aparece (um gato) se posiciona inicialmente no ponto (0,0), isto é, em $x = 0$ e $y = 0$ e o palco apresenta-se em branco.

Por ser uma linguagem de programação, toda ação deve ser dada na forma de comandos devidamente expressos nos seus blocos lógicos que são arrastados para uma área de comandos, conectados uns aos outros ou executados através de sub-rotinas ou de subdivisões.

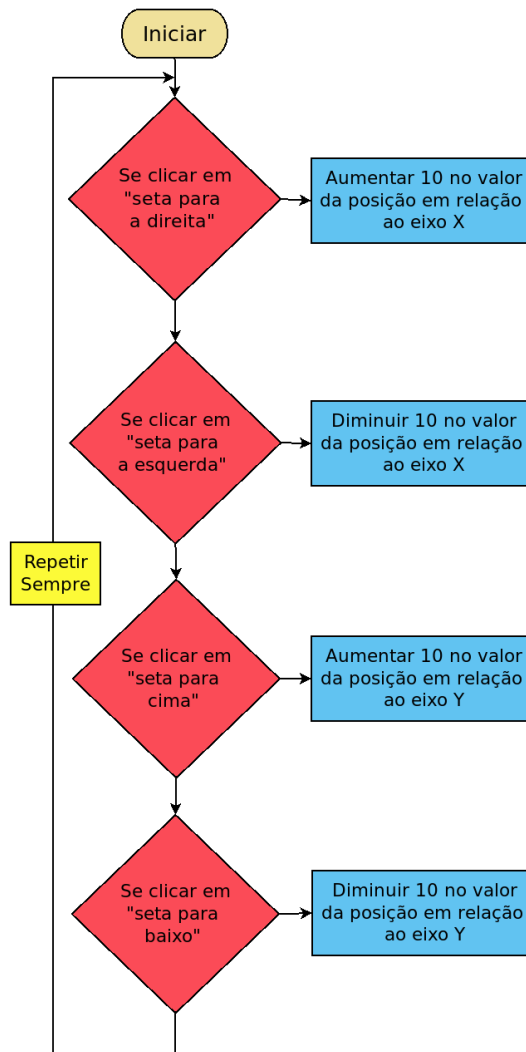
As suas características de linguagem de programação e de um quebra-cabeça fazem com que o *Scratch* possa ser usado tanto como uma ferramenta para produção de Objetos de Aprendizagem como um meio de desenvolver o Pensamento Computacional do estudante e, em ambos os casos, esse uso poderá ser de forma interdisciplinar. Como não é necessário digitar um comando e a resposta à programação é quase que imediata, o usuário pode experimentar rapidamente o que criou e corrigir os erros caso seja necessário fazendo assim um ciclo contínuo de novas ideias, desenvolvimento, experimentação, aprendizagem e compartilhamento levando a novas ideias.

3.1. Exemplo de como programar no *Scratch*

Um exemplo de programação no *Scratch* é a simulação de um peixe nadando. Nesta simulação, um peixe irá se mover no palco de acordo com as setas do teclado do computador.

Antes de desenvolver a programação, é interessante fazer um pequeno planejamento de todos os passos que serão necessários para o resultado esperado. Esse planejamento pode ser feito ao desenvolver um diagrama simples, onde o programador pode elencar todos as ações necessárias. A Figura 6 é um exemplo de como esse planejamento pode ser feito para essa simulação.

Figura 6 – Planejamento da movimentação do peixe

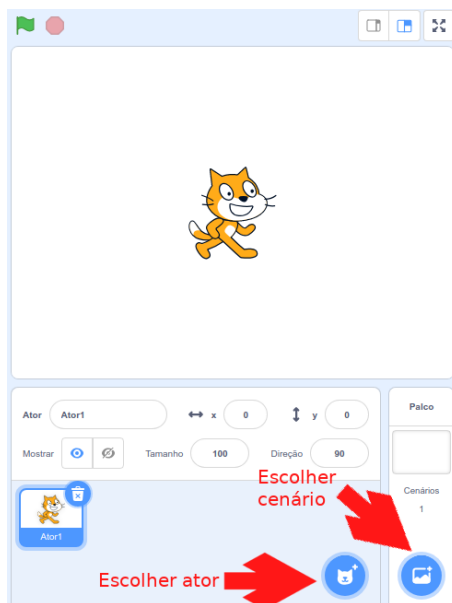


Fonte – Criado pelo Autor

Após tudo planejado, é necessário escolher o ator e o cenário. Essas escolhas são feitas na região 3, logo abaixo do palco, conforme a Figura 7.

No banco de imagens do *Scratch* encontramos várias opções de atores e cenários e, nesse caso, escolheu-se como ator a imagem *fish* e como cenário a imagem *Underwater 1*. Com o ator e o cenário escolhidos, pode-se começar a programar de acordo com o planejado.

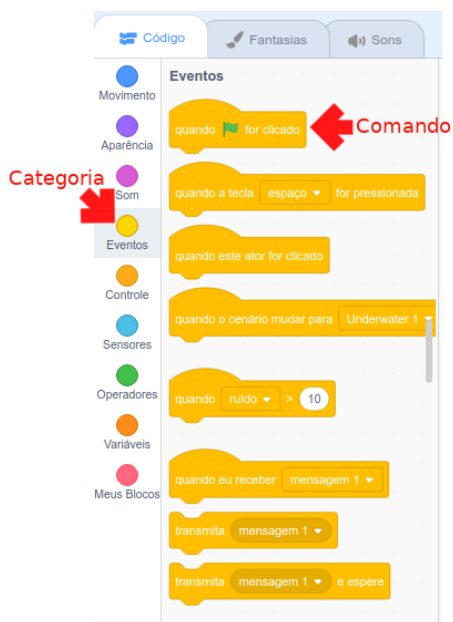
Figura 7 – Região 3



Fonte – Adaptado do Site do Scratch

Todos os comandos necessários para a programação estão na região 1 separados por categorias. A primeira categoria a ser escolhida será a de “Eventos” e o primeiro comando desta categoria será “quando *bandeira verde* for clicado” conforme a Figura 8.

Figura 8 – Região 1



Fonte – Adaptado do Site do Scratch

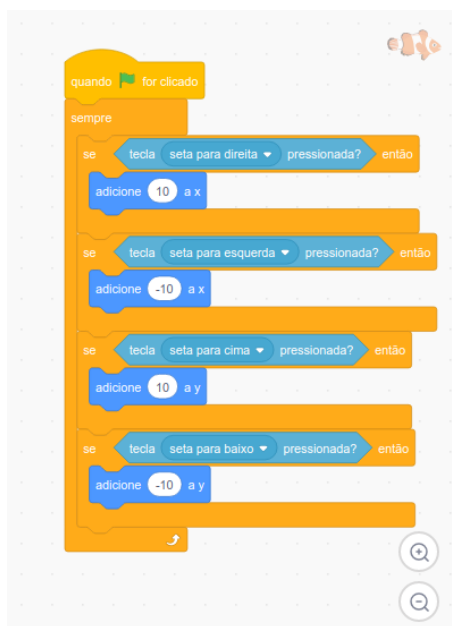
Então, clica-se no comando com o *mouse* e o arrasta até a região 2 para começar a programação. Este comando indica que, ao clicar na bandeira verde acima do palco, o

programa começará a ser executado.

Ao observar o planejamento feito na Figura 6, as próximas ações irão sempre se repetir, então o próximo comando deverá expressar essas repetições. Esse é o comando “sempre” que está dentro da categoria “controle”. Esse comando tem uma abertura que será preenchida com os comandos a serem escolhidos. Quando esse comando é arrastado até a região 2, ele se encaixa logo abaixo do primeiro comando que foi escolhido anteriormente.

O próximo comando a ser escolhido é o comando “se - então” que também está na categoria “controle”. Novamente esse comando será arrastado com o *mouse* para a região 2 e será encaixado dentro do comando “sempre”. Este comando estará sempre testando se a condição colocada em sua lacuna é verdadeira ou falsa, caso a condição seja verdadeira, ele executará o que for determinado dentro dele e, caso seja falso, irá passar para o próximo comando. Nesta lacuna será colocada a condição “tecla *seta para direita* pressionada?” que se encontra na categoria “sensores” e dentro estará o comando “adicione 10 a x” que está na categoria “movimento”. Esse conjunto de comandos será repetido mais três vezes como foi planejado na Figura 6. Ao final, a programação estará organizada conforme a Figura 9.

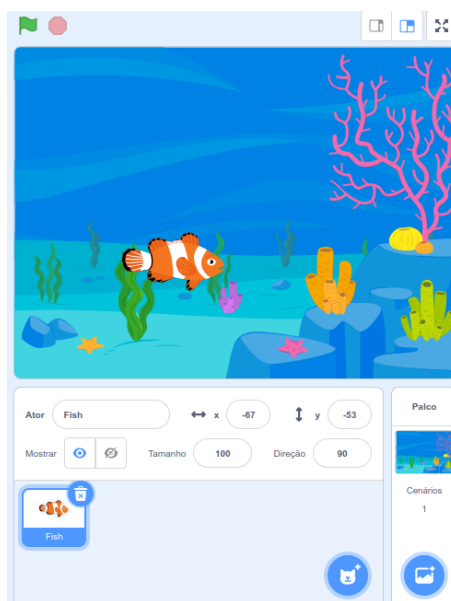
Figura 9 – Região 2



Fonte – Adaptado do Site do *Scratch*

E, ao clicar na bandeira verde acima do palco, o peixe irá se movimentar de acordo com as setas pressionadas no teclado do computador.

Figura 10 – Região 3



Fonte – Adaptado do Site do *Scratch*

4 Metodologia do curso

A Metodologia do curso foi desenvolvida de forma a apresentar o *Scratch* como ferramenta para desenvolver objetos de aprendizagem pelo professor e também para estimulá-lo a usar este programa com os seus estudantes para resolverem problemas usando o pensamento computacional como premissa. Desta forma o professor também estimula os estudantes a desenvolverem habilidades relacionadas não só ao conteúdo educacional apresentado como também à construção de soluções de problemas.

Para tanto foram realizados sete encontros com quatorze professores de modalidades de ensino e formações específicas diferentes, sendo:

- 4 professores que atuam na modalidade de Anos Iniciais - 1º ao 5º ano.
- 2 professores que atuam em salas de recursos para estudantes com necessidades especiais.
- 4 professores que atuam na modalidade de Anos Finais - 6º ao 9º ano - sendo 3 professores de Matemática e 1 de Ciências.
- 4 professores que atuam na modalidade de Ensino Médio - 1º ao 3º ano do Ensino Médio - sendo 2 professores de Matemática e 2 de Química.

Os encontros foram distribuídos de acordo com a Tabela 3 abaixo:

Tabela 3 – Planejamento do Curso sobre o *Scratch*

Encontro	Conteúdo	Objetivo
1	Pensamento Computacional	Resolver problemas usando programação através do Pensamento Computacional como premissa
2	Apresentação do <i>Scratch</i>	Ambientar os professores ao programa
3	Contando Histórias	Contar uma história através da programação do <i>Scratch</i>
4	Programação de jogos	Desenvolver um jogo com a programação do <i>Scratch</i>
5	Objetos Digitais de Aprendizagem	Conhecer o que são Objetos Digitais de Aprendizagem e como utilizá-los
6	Programando Objetos Digitais de Aprendizagem	Desenvolver Objetos de Aprendizagem com o <i>Scratch</i>
7	Projetos desenvolvidos	Mostrar para a turma os projetos desenvolvidos com os estudantes

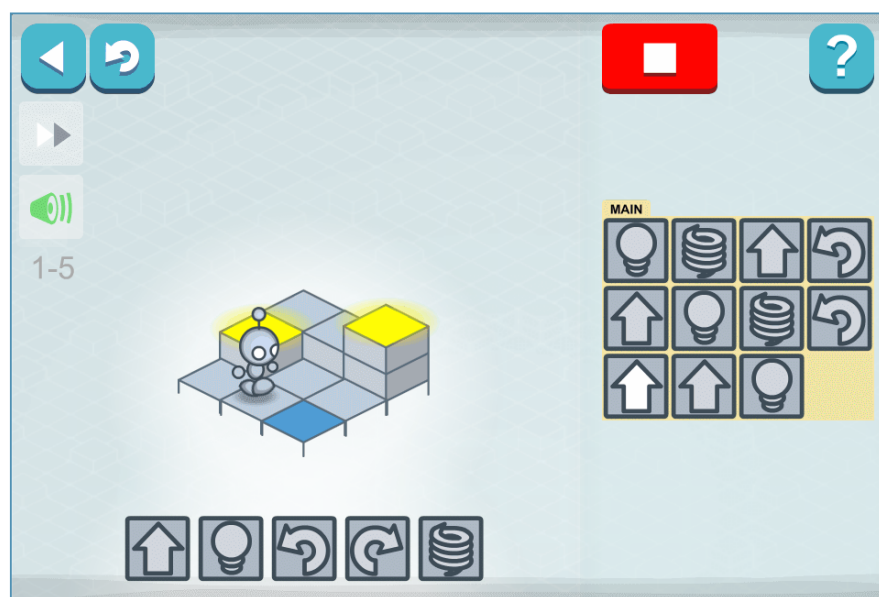
Fonte – Criado pelo autor

O curso também contou com exercícios e estudos à distância, para que os professores pudessem planejar, desenvolver e verificar a aceitação de seus estudantes ao *software*. Desta forma, cada professor teve a oportunidade de, ao final do curso, mostrar quais foram os problemas propostos ou como desenvolveram os programas a serem apresentados, além de mostrarem como os estudantes se desenvolveram não só em suas disciplinas, como também em relação ao modo de resolverem problemas.

4.1. Detalhamento dos encontros

1. Pensamento Computacional

Através do site <https://lightbot.com/flash.html> foi introduzida a ideia de Pensamento Computacional. Neste site a pessoa tem que programar um robô através de setas e outras imagens para que “acenda” os blocos azuis e os transforme em amarelos (vide Figura 11).

Figura 11 – *Lightbot*

Fonte – <https://lightbot.com/flash.html>

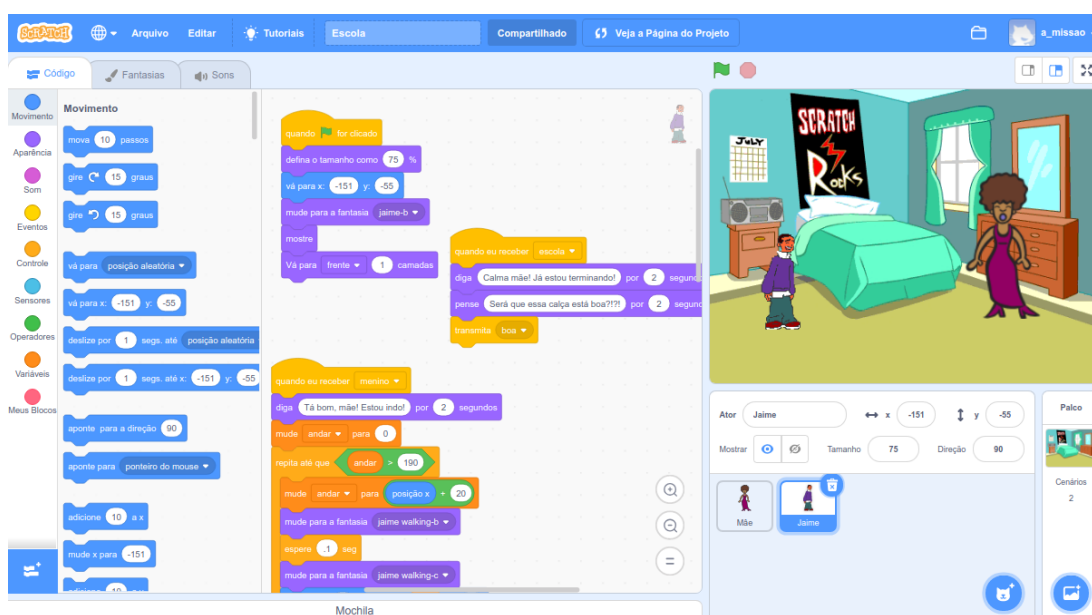
É interessante notar que para que este robô acenda todos os blocos azuis, o jogador deve primeiro escolher o caminho em que o robô irá passar, dividir o caminho em pequenos blocos e escolher, dentre disponíveis, os comandos necessários e, por fim, testar a programação. Essas são características do Pensamento Computacional aplicada a um problema de caminhos.

Para melhor estruturar essas ideias, os professores leram o primeiro capítulo do artigo “Desenvolvimento do Pensamento Computacional Através de Atividades Desplugadas na Educação Básica” - “Pensamento Computacional” (BRACKMANN, 2017) e, desta forma, discutiram suas características e como elas também serão trabalhadas durante o curso.

2. Apresentação do *Scratch*

Neste encontro os professores foram apresentados à página do *Scratch* através de uma história animada construída no *software* (vide Figura 12) para descontrair e iniciar uma conversa sobre a computação e a sala de aula. Após esse primeiro momento foram realizados pedidos de abertura de contas especiais de professor na plataforma do *Scratch*, para que, desta forma todos possam utilizar a plataforma com seus estudantes. Após esse procedimento, os professores abriram pela primeira vez a área de criação de programas e fizeram seu primeiro programa escolhendo um *sprite* com que mais se identificassem e, com algumas orientações de como escolher os comandos, se apresentassem através de uma história animada.

Figura 12 – Exemplo de animação



Fonte – Criado pelo autor no site <https://scratch.mit.edu>

Ao final desse encontro, foi solicitado a cada professor que, até o próximo encontro, examinassem o conteúdo que estavam desenvolvendo em sala e observassem os seus estudantes para verificar quais as dificuldades encontradas.

3. Contando Histórias

Aproveitando a ideia da pequena história desenvolvida no encontro passado, neste encontro os professores programaram histórias animadas mais elaboradas, com movimentos dos *sprites* e com pausas para cada um desses personagens. Utilizaram também as trocas de cenários e de figurinos (no *Scratch* chamado de fantasias) dos personagens.

Para isso, antes da programação, cada professor planejou a história a ser contada, com os diálogos que os personagens iriam desenvolver, quais os movimentos seriam necessários (dentro da capacidade do *Scratch* e de cada *sprite* escolhido) e em quais cenários essa história seria contada subdividindo, desta forma, a história em pequenos pedaços e assim introduzindo a ideia de criação de um pequeno algoritmo que pudesse depois ser transformado em uma programação.

Com os algoritmos prontos, pôde-se dar início a programação do *Scratch*, utilizando principalmente os conjuntos de comandos de movimento, de aparência (onde se encontram os comandos de fala escrita), de eventos e de controle.

Ao fim do encontro, para a fase à distância, os professores desenvolveriam uma história a ser contada através do *Scratch* para os estudantes e também mostrar-lhes como desenvolver as suas próprias histórias animadas.

4. Programação de jogos

Para a melhor compreensão de como desenvolver um jogo, de forma dirigida e conjunta, todos os professores construíram um algoritmo em que um *sprite* pudesse “flutuar” no ecrã do *Scratch*. Considerando esse algoritmo desenvolvido em conjunto como apenas uma base de vários jogos, cada professor poderá escolher como o seu jogo será apresentado, escolhendo cenários e personagens distintos mas com a mesma movimentação inicial. Por exemplo, o cenário pode ser o fundo do mar e o personagem um mergulhador ou então o cenário pode ser o céu de uma cidade e o personagem uma ave, nestes dois casos o movimento de “flutuar” será o mesmo mas os obstáculos apresentados para cada personagem criarão jogos diferentes entre si, com objetivos também diferentes.

Ao escolherem os personagens e o cenário, o jogo começa a ficar mais pessoal, e assim o restante do algoritmo e dos objetivos dos jogos serão escritos de forma mais individual.

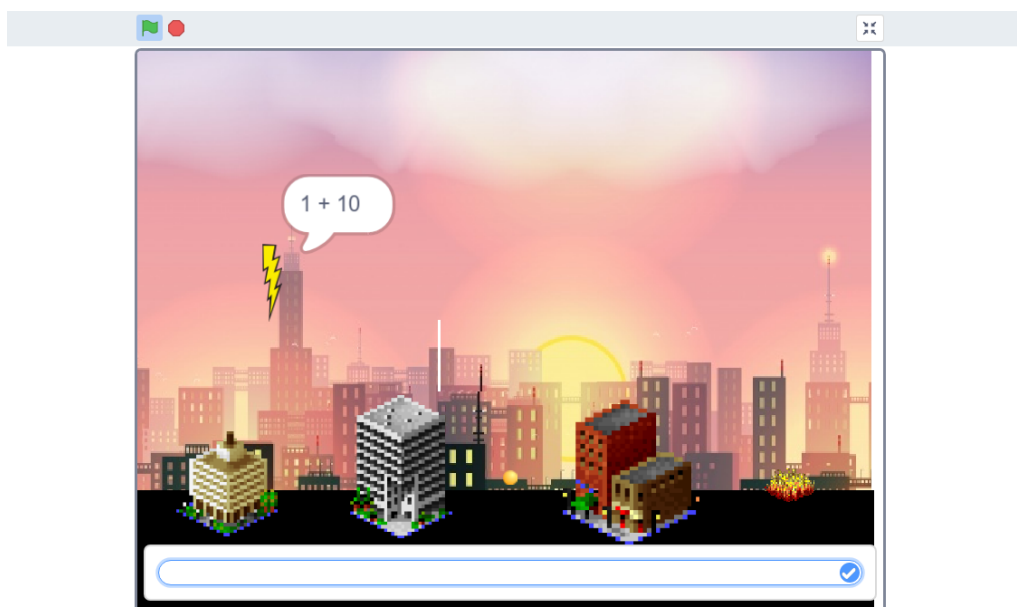
A transformação desse algoritmo em uma programação do *Scratch* foi realizada utilizando os conjuntos de comandos utilizados no encontro anterior juntamente com os conjuntos de sensores, operadores e variáveis disponíveis.

5. Objetos Digitais de Aprendizagem

O quarto encontro teve como objetivo principal o estudo e o primeiro esboço de um Objeto Digital de Aprendizagem a ser desenvolvido pelos professores de acordo com suas disciplinas e conteúdos, levando em conta as dificuldades observadas em sala de aula, e, quando pronto, apresentado aos estudantes.

Em um primeiro momento, os professores tiveram contato com um jogo desenvolvido com o objetivo de praticar os conceitos de soma e subtração de números naturais restringidos até o número 10 entre as parcelas (vide Figura 13). Desta forma eles puderam ver como elaborar um Objeto Digital de Aprendizagem com o *Scratch*.

Figura 13 – Jogo Soma e Subtração



Fonte – Criado pelo autor no site <https://scratch.mit.edu>

Após o jogo os professores leram e discutiram em grupos de no máximo 3 pessoas o primeiro capítulo do livro “Objetos de aprendizagem: Teoria e prática” - “Objetos de aprendizagem: Conceitos Básicos” (TAROUCO; FABRE; TAMUSIUNAS, 2003) e, após essa discussão, começaram a delinear, cada um, um projeto de Objeto Digital de Aprendizagem que pudessem desenvolver no *Scratch* e para a fase à distância, eles ficaram com a incumbência de escrever um algoritmo que pudesse detalhar todos os momentos desse Objeto de Aprendizagem seja ele uma história, um jogo ou qualquer outra ideia que tivessem delineado neste encontro.

6. Programando Objetos Digitais de Aprendizagem

Com os algoritmos prontos, os professores programaram, através de novas orientações de quais comandos estavam à disposição e como funcionam sub-rotinas no *Scratch*, caso fossem necessárias, os seus Objetos Digitais de Aprendizagem através do *Scratch*, tendo como objetivo, apresentar aos seus estudantes na próxima fase à distância. Desta forma, todos puderam perceber como se estrutura, na prática, uma programação e também como fazer um Objeto de Aprendizagem voltado para as necessidades específicas de cada professor, de cada conteúdo e o mais importante, de cada estudante que irá usá-lo.

A fase à distância desse encontro foi apresentar aos estudantes os Objetos Digitais de Aprendizagem desenvolvidos por eles e fazer uma nova observação da aceitação desta ferramenta e como foi o desenvolvimento da aprendizagem de cada estudante.

7. Projetos desenvolvidos

Este encontro foi planejado para que cada professor apresentasse os projetos desenvolvidos por eles e pelos seus estudantes. Além dessa apresentação, os professores fizeram considerações sobre como cada estudante percebeu os Objetos Digitais de Aprendizagem desenvolvidos e como esses mesmos estudantes desenvolveram os seus próprios programas dentro da plataforma, utilizando-a para contar histórias, para elaborar jogos, ou mesmo para resolver problemas elaborados pelo professor através da programação.

4.2. Principais projetos desenvolvidos

Para uma melhor organização dos principais projetos e o entendimento do contexto dos resultados, esse tema foi dividido em três momentos importantes do curso: O primeiro contato dos estudantes com a plataforma através do desenvolvimento de histórias animadas, a produção de Objetos Digitais de Aprendizagem pelos professores e a utilização do *Scratch* pelos estudantes no desenvolvimento de jogos ou de programas através de problemas propostos.

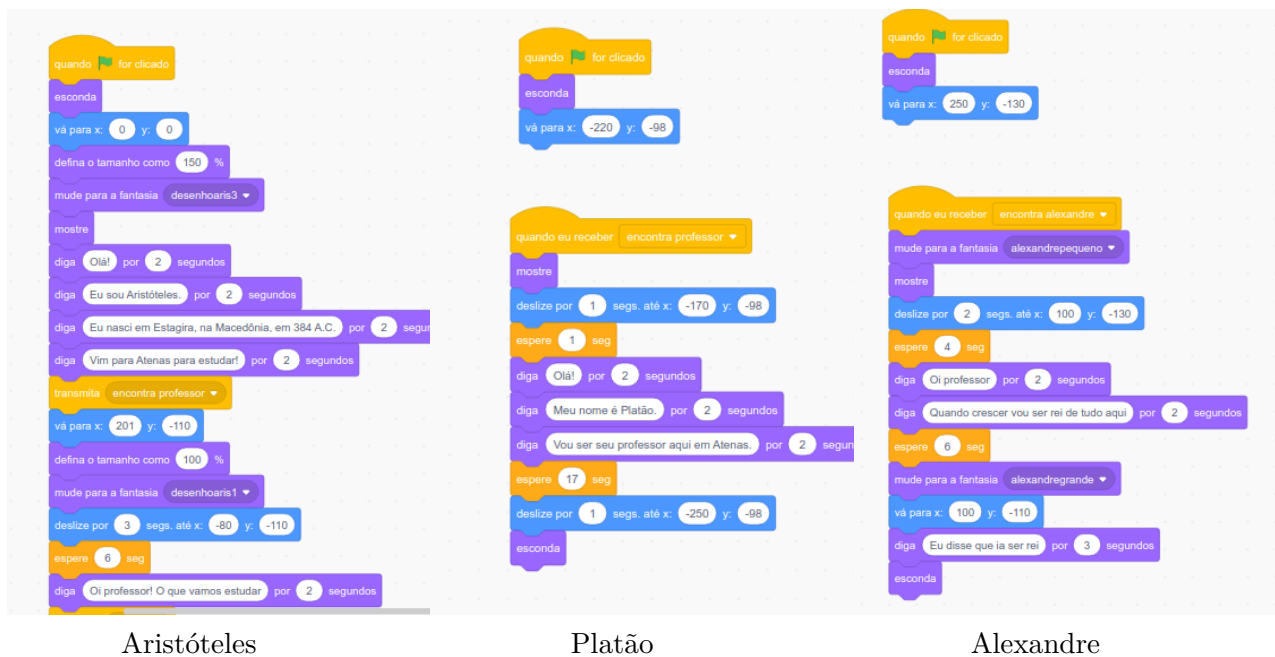
A primeira atividade dos professores com seus estudantes, o desenvolvimento da história animada, teve uma ótima repercussão entre os professores que além de contarem algumas histórias para seus estudantes, puderam perceber como seus estudantes podem interagir com essas histórias. Os professores de anos iniciais juntaram-se em grupo e usaram estas histórias como forma de fixação de leitura e interpretação de texto, já que alguns de seus estudantes acabaram de ser alfabetizados e estavam ainda tendo dificuldades em compreender o que estavam lendo. Os que tinham mais facilidade em interpretação de texto, segundo os professores, se motivaram a também escreverem histórias e as estavam desenvolvendo tanto no site do *Scratch* como em forma de quadrinhos em seus cadernos. Os professores que atuavam em salas de recursos usaram outras abordagens para contar suas histórias, usando estímulos como sons de animais e objetos que encontraram em *sites* de efeitos sonoros para desenvolver uma melhor interação com os seus estudantes.

Os resultados mais significativos desse primeiro contato com os estudantes vieram dos relatos dos professores que atuam nos Anos Finais e no Ensino Médio, pois seus estudantes foram estimulados a contar as suas próprias histórias dentro da plataforma. Para melhor direcionar essas histórias, os professores foram orientados que pedissem aos estudantes para pesquisar um fato histórico ou um personagem relacionado ao conteúdo trabalhado em sala. Desta forma a programação dessa história envolveu outras disciplinas e motivou os estudantes a aprenderem o conteúdo e também a representá-lo.

Dentre esses trabalhos o que mais se destacou foi desenvolvido pelo professor de Ciências na modalidade Anos Finais, cujo conteúdo foi a “História da Física”: Os seus

estudantes foram separados em grupos e escolheram os personagens históricos que desenvolveram a Física como Ciência. Pesquisaram quais as contribuições desses personagens e transformaram essa narrativa em uma programação no site do *Scratch*. Usando como exemplo a história de Aristóteles narrada por um destes estudantes e analisando a programação realizada, pode-se perceber como esta programação envolve várias disciplinas (vide o conjunto de Figuras 14).

Figura 14 – Programação da História de Aristóteles



Fonte – <https://scratch.mit.edu/projects/378647189/>

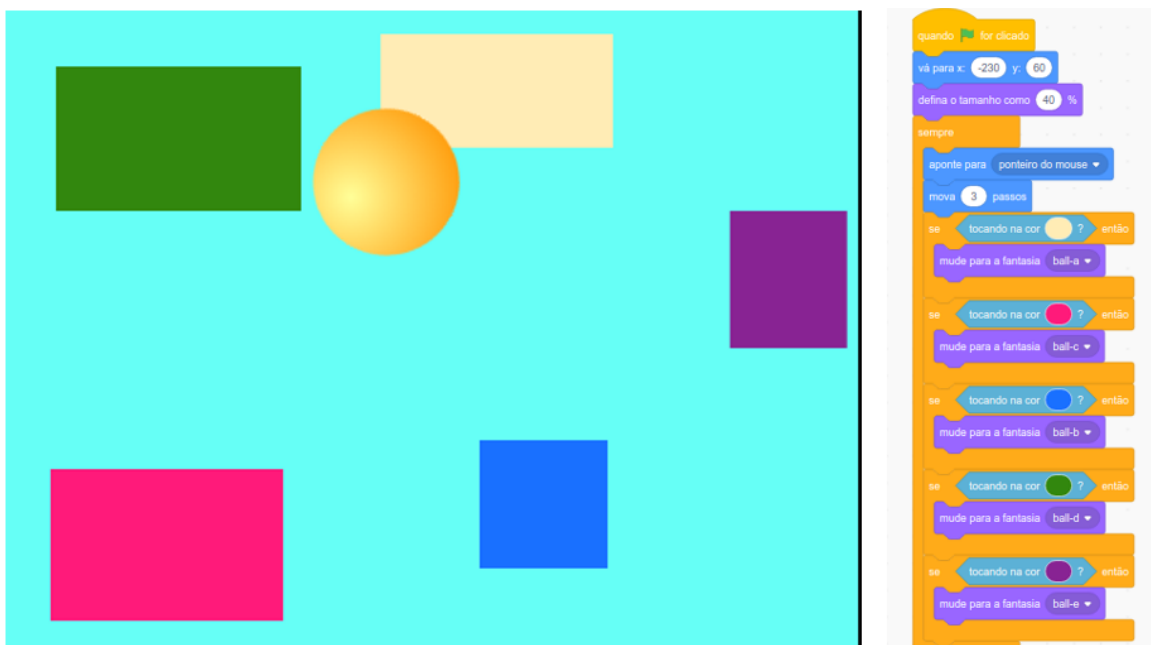
Envolvendo a disciplina de História, o estudante usou três personagens: Platão, Alexandre e o próprio Aristóteles. No campo da Matemática, a primeira aparição de Aristóteles aparece um pouco maior que no restante da narrativa, e vemos na programação que o estudante usou a ideia de porcentagem para este aumento e, além disso, para posicioná-lo usou o sistema do Ecrã do *Scratch*, que é baseado no Plano Cartesiano. Em Filosofia, usou os pensamentos filosóficos da época para mostrar como foram as contribuições de Aristóteles para a Física, para a Óptica e para a própria Filosofia. Ainda temos o estudo da Língua Portuguesa, que perpassa por toda a história, tanto na narração como nos diálogos das personagens.

Após esse primeiro momento, os professores desenvolveram Objetos Digitais de Aprendizagem para apresentar aos estudantes.

Os professores que atuavam em salas de recurso desenvolveram dois programas para ajudar os estudantes que tinham dificuldades motoras e não conseguiam ter o domínio do *mouse*.

O primeiro programa (vide Figura 15) foi desenvolvido para que o estudante pudesse movimentar uma bola que flutuava e mudava de cor ao encostar em algum elemento no palco.

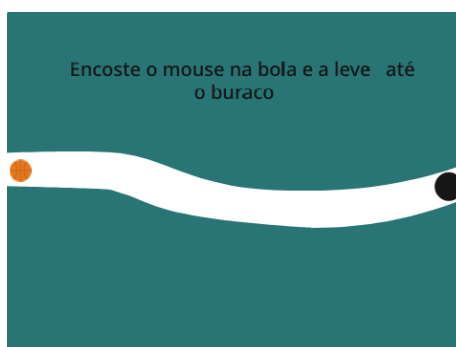
Figura 15 – Palco e Programação da bola que muda de cor



Fonte – <https://scratch.mit.edu/projects/333029656/>

O segundo programa tinha como objetivo conduzir uma bola com o mouse até o buraco através de um caminho não podendo tocar nas bordas do caminho.

Figura 16 – Palco da Condução de bola



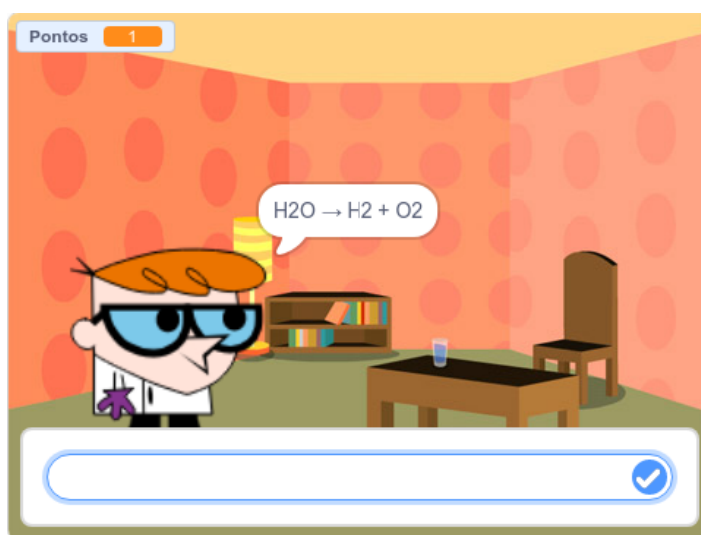
Fonte – <https://scratch.mit.edu/projects/400103790/>

Esses professores de sala de recursos desenvolveram esses projetos para estudantes com necessidades especiais de movimentos com o objetivo de ajudá-los a adquirir uma melhor coordenação motora para o uso do computador não só como meio de comunicação e informação, mas também como ferramenta de aprendizagem, já que ao conseguir manusear

o mouse, este estudante terá capacidade de usar programas e aplicativos no computador com mais independência.

Um outro Objeto Digital de Aprendizagem relevante foi desenvolvido por um professor de Química para o Ensino Médio.

Figura 17 – Palco do *Quiz* Reações e Balanceamento



Fonte – <https://scratch.mit.edu/projects/334853158/>

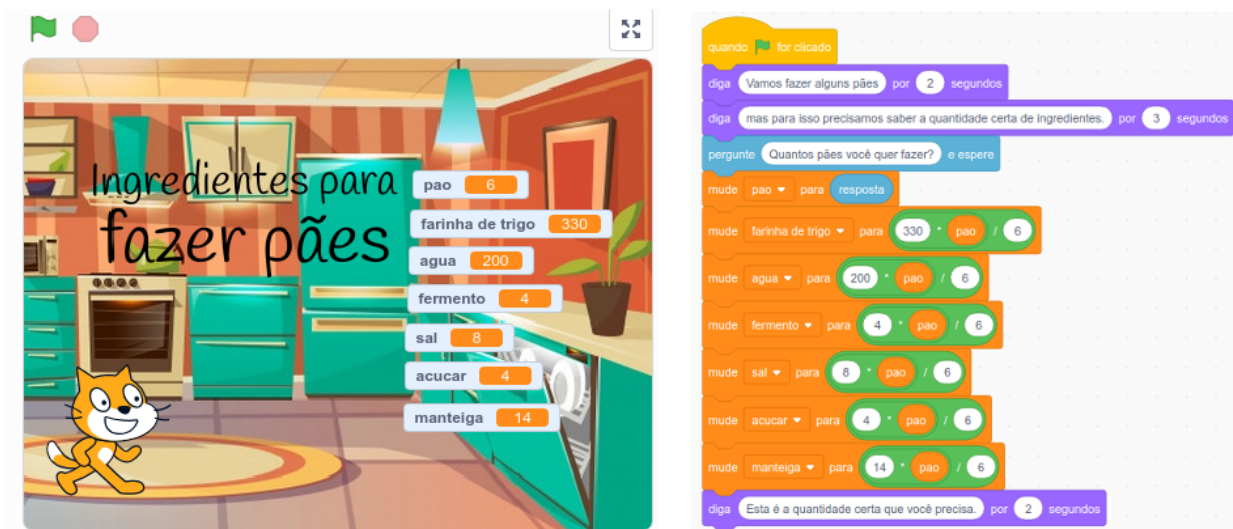
Este é um jogo em que o estudante interage com o personagem através de perguntas e respostas sobre quais são os tipos de equações químicas e qual o balanceamento correto em determinadas equações. Ao iniciar, o estudante escolhe uma entre duas opções: Distinguir qual é o tipo de equação que irá aparecer ou fazer o balanceamento de alguma equação. Este projeto foi desenvolvido com a premissa de que o estudante já tenha conhecimento do que é balanceamento e dos tipos de reações químicas e, com essas perguntas e respostas, ele pode fixar esses conceitos.

Os professores mostraram interesse maior em produzir seus próprios Objetos Digitais de Aprendizagem pois, desta forma, poderiam focar essas produções em conteúdos mais complexos ou em dificuldades específicas de cada estudante. Observou-se também que o professor começou a sentir-se mais seguro não só com o *Scratch* mas também com as suas produções, porque, como são autorais, pode ser melhor administrada pelo autor e o seu conteúdo pode estar mais próximo do que é desenvolvido em sala de aula.

O terceiro momento do curso consistia em incentivar os estudantes a desenvolverem seus próprios algoritmos e programas. Mesmo sendo o objetivo principal o estudo dos conteúdos desenvolvidos dentro de sala de aula, os estudantes poderiam construir jogos, criar animações, histórias ou qualquer outro programa que envolvesse esse conteúdo. Para tanto, os professores poderiam elaborar situações problemas ou propor aos estudantes alguma outra forma de estudo a ser transformada em programação.

Dentre as situações problemas propostas, destaca-se o trabalho realizado pelos professores de Matemática dos Anos Finais relacionado a razão e proporção. O problema consistia em saber qual a quantidade de cada um dos ingredientes necessários para fazer determinada receita. Para desenvolverem os programas, os estudantes foram separados em grupos e cada grupo escolheu uma receita para ser estudada. Um dos programas desenvolvidos calcula a quantidade de ingredientes necessários para fazer pão.

Figura 18 – Palco e Programação da Receita de Pão



Fonte – <https://scratch.mit.edu/projects/401180286/>

Neste programa (vide Figura 18), a pessoa informa qual a quantidade de pães quer fazer e o programa calcula a quantidade de ingredientes necessários para a receita.

Mesmo com uma programação simples e com poucos comandos, percebe-se como o estudante usou os conceitos de desenvolvimento do Pensamento Computacional apresentados. Em primeiro lugar, houve uma coleta dos dados iniciais, onde o estudante pesquisou quais ingredientes eram necessários para a produção de certa quantidade de pães. Após a coleta de dados, analisou a quantidade de cada um dos ingredientes e, através de cálculos envolvendo proporção, automatizou soluções por meio de pensamento algorítmico, generalizando estas quantidades de ingredientes para qualquer quantidade de pães necessários. Para chegar a essa solução, o estudante não só usou as regras de proporção, que era o conteúdo inicial a ser estudado, mas também aprendeu a pesquisar e a analisar os textos da receita.

Um segundo trabalho produzido por estudantes de anos finais a ser destacado é um jogo de tabuleiro que envolve um questionário estimulado pelo professor de Ciências. Envolvendo dois jogadores, o objetivo é chegar ao final do tabuleiro com a ajuda de um dado virtual e, parando em determinados lugares, o jogador deverá responder a uma pergunta envolvendo conhecimentos sobre *Misturas Homogêneas e Heterogêneas* e sobre

Estados Físicos da Matéria (vide Figura 19).

Figura 19 – Palco do jogo de Tabuleiro



Fonte – <https://scratch.mit.edu/projects/403307051/>

Para este jogo foram utilizados três *sprites* (personagens) programáveis: dois que representam os jogadores e o dado que é usado para o movimento no tabuleiro. Ao analisar a programação de cada um desses personagens, pode-se perceber características interdisciplinares relevantes além do conteúdo trabalhado diretamente pelo professor de Ciências.

A programação do dado inclui porcentagem, localização em um plano cartesiano e, apesar de ainda não calcular probabilidades, o estudante já tem contato com experimentos aleatórios (vide Figura 20)

Figura 20 – Programação do dado



Fonte – <https://scratch.mit.edu/projects/403307051/>

A programação dos outros dois personagens além de incluir porcentagem e localização no plano cartesiano, tem também a movimentação nos eixos x e y do plano usando

números positivos e negativos para orientar em qual sentido e direção esses personagens irão se movimentar (vide Figura 21).

Figura 21 – Programação dos participantes



Fonte – <https://scratch.mit.edu/projects/403307051/>

Existe também a inserção dos conteúdos de Ciências estudados na pesquisa das perguntas formuladas e suas respostas durante a movimentação dos personagens. Desta forma, o estudante, além de ser estimulado a criar um jogo autoral, adquiriu habilidades que seriam mais complexas para o seu nível de desenvolvimento, onde o professor desempenhou o papel de orientar esse progresso.

5 Conclusão

O *Scratch* se mostra uma ferramenta importante quando se busca por novas abordagens de educação mediada pelo uso das novas Tecnologias Digitais de Informação e de Comunicação de forma interdisciplinar onde o estudante é alçado a protagonista. Quando o estudante é estimulado a pensar de forma lógica e a resolver problemas através da linguagem de programação, a sua capacidade de abstrair e de fazer relações entre diversos conteúdos e conhecimentos também é estimulada e, além disso, o professor também é influenciado a se tornar cada vez mais um orientador em vez de um detentor de conhecimentos.

Mas de nada adianta ferramentas como essa estarem disponíveis se o professor não conhecer as suas potencialidades e também não for estimulado a usá-las. Foi o que se percebeu no início do curso, quando alguns professores se mostraram receosos ao serem apresentados à ideia de inserir a computação em sua prática pedagógica. Esses professores tinham a percepção de que a programação só poderia ser pensada e realizada através de comandos rebuscados dentro de uma linguagem de máquina, por isso foi importante, logo no primeiro encontro, apresentar o *lightbot*, mostrando que a programação está mais ligada ao desenvolvimento de um pensamento lógico e a resolução de problemas do que à linguagens específicas. Mesmo após esse início, alguns só conseguiram se motivar mais e a aprender um pouco mais sobre como usar o Pensamento Computacional após serem apresentados realmente ao *Scratch* e fazerem a sua primeira historinha para se apresentarem, percebendo que a programação pode ser um fator de desenvolvimento do pensamento lógico e abstrato dos estudantes.

O primeiro momento em que os estudantes foram estimulados a utilizar a plataforma seja para acompanhar ou para criar uma história, foi quando os professores já conseguiam fazer as suas próprias histórias. Os estudantes se motivaram a aprender mais sobre a programação de seus personagens e de suas histórias, de forma a conseguirem organizar os pensamentos e a planejar melhor as suas próximas ações. Essa empolgação dos estudantes em suas primeiras programações, fizeram com que os professores também ficassem animados e receptivos para o segundo momento importante do curso, que foi a construção de objetos digitais de aprendizagem, que requereu dos professores um pouco mais de planejamento. Para construir seus Objetos Digitais de Aprendizagem os professores pesquisaram, através das dificuldades de seus estudantes, quais conteúdos deveriam estar inseridos, desenvolveram algoritmos que expressavam como deveriam funcionar e fizeram a programação. Todo esse planejamento e desenvolvimento de objetos os ajudou a perceber como deveriam orientar os seus estudantes para que estes pudessem construir seus próprios programas.

Através dos relatos dos professores durante a apresentação dos projetos desenvolvidos no último encontro, percebeu-se que, quando os estudantes foram incentivados a desenvolverem jogos, histórias ou até mesmo a resolverem pequenos problemas através do *Scratch*, a postura dos professores foi mudando gradativamente, deixando de ser os detentores do conhecimento e passando a ser orientadores, facilitando a criação, a independência, o raciocínio lógico e a aprendizagem desse estudante. Além disso os estudantes estavam deixando de ser apenas usuários da tecnologia e passavam a ser criadores e desenvolvedores, interagindo com essas tecnologias de forma a produzir novos conhecimentos de forma interdisciplinar e conectada com as informações que estão dispostas ao seu redor.

Desta forma, o *Scratch* mostra-se tanto como uma plataforma de aprendizagem interdisciplinar oferecendo ao estudante novos conhecimentos e habilidades, como também pode ser usado para inserir os princípios do Pensamento Computacional em sala de aula, trazendo novas perspectivas à aprendizagem, à relação entre professores e estudantes e à relação entre o estudante e as tecnologias. Para trabalhos futuros existem ainda estudos a serem feitos sobre como os conceitos da gamificação podem auxiliar no desenvolvimento de novos projetos no *Scratch* para melhorar a aprendizagem, na compreensão dos conteúdos apresentados e no engajamento de professores e estudantes.

Referências

BRACKMANN, C. P. Desenvolvimento do pensamento computacional através de atividades desplugadas na educação básica. UFRGS - Universidade Federal do Rio Grande do Sul, Rio Grande do Sul - Brasil, 2017. Disponível em: <<https://lume.ufrgs.br/handle/10183/172208>>. Citado 3 vezes nas páginas 12, 19 e 35.

ISTE/CSTA. *Computational Thinking Leadership Toolkit*. Computer Science Teachers Association (CSTA) and the International Society for Technology in Education (ISTE), 2011. Disponível em: <<https://id.iste.org/docs/ct-documents/ct-leadership-toolkit.pdf>>. Citado 2 vezes nas páginas 17 e 18.

MORAN, J. A integração das tecnologias na educação. In: *A Educação que desejamos: novos desafios e como chegar lá*. Campinas, SP - Brasil: [s.n.], 2013. p. 89–90. Disponível em: <http://www.eca.usp.br/prof/moran/site/textos/tecnologias_eduacacao/integracao.pdf>. Citado na página 14.

NATIONAL RESEARCH COUNCIL. *Report of a Workshop on the Pedagogical Aspects of Computational Thinking*. Washington, DC: The National Academies Press, 2011. ISBN 978-0-309-21474-2. Disponível em: <<https://www.nap.edu/catalog/13170/report-of-a-workshop-on-the-pedagogical-aspects-of-computational-thinking>>. Citado na página 17.

OSTI, A. Concepções sobre desenvolvimento e aprendizagem segundo a psicogênese piagetiana. *Revista de Educação*, XII, p. 109–118, 2009. Citado na página 15.

PAPERT, S. *LOGO: computadores e educação*. [S.l.]: Editora Brasiliense, 1985. Citado 2 vezes nas páginas 15 e 16.

TAROUCO, L. M. R.; FABRE, M.-C. J. M.; TAMUSIUNAS, F. Reusabilidade de objetos educacionais. *RENOTE - Revista Novas Tecnologias para a Educação*, Porto Alegre - Brasil, 2003. ISSN 1679-1916. Disponível em: <<https://seer.ufrgs.br/renote/article/view/13628>>. Citado 3 vezes nas páginas 22, 24 e 38.

WING, J. M. Computational thinking. *COMMUNICATIONS OF THE ACM*, v. 49, p. 33–35, 2006. Citado 2 vezes nas páginas 16 e 18.