

Prática do Pensamento Computacional e da Língua Inglesa utilizando o Scratch: uma sequência didática

Reinildo Souza da Silva, Claudia Pinto Pereira
Universidade Estadual de Feira de Santana (UEFS)
Feira de Santana, Bahia, Brasil
reinildo@live.com, claudiap@uefs.br

RESUMO

É crescente o número de iniciativas de Pensamento Computacional (PC) na escola em diversos países. Entretanto, o Brasil ainda carece de ações que implementem tal prática. O ensino de programação e da língua inglesa, já inclusos no currículo da educação básica, podem ser uma alternativa para a prática do PC. Este trabalho apresenta uma sequência didática de aulas de programação em Inglês, com Scratch, para estudantes do Ensino Fundamental I, aplicando a metodologia de Ensino de Segunda Língua Baseado em Conteúdo. Este design de experimento foi resgatado dos planos de aula de 2018, que foram reestruturados e reclassificados de acordo com a Taxonomia de Bloom revisada, para a proposição de uma sequência didática. A partir dela e de sua reestruturação, foi possível analisar a evolução dos níveis hierárquicos do domínio cognitivo e da complexidade dos projetos desenvolvidos, além de mostrar que é possível praticar uma segunda língua ou qualquer outro conteúdo enquanto se aprende programação.

PALAVRAS-CHAVE

Pensamento Computacional, Scratch, Aprendizagem de Computação, Programação de Computadores, Ensino Fundamental

1 INTRODUÇÃO

O processo educacional está em constante evolução para acompanhar as demandas dos cenários social, econômico e tecnológico. Ao longo dos anos, muitas metodologias foram criadas para tornar o processo de aprendizagem mais ativo e significativo, tais como a aprendizagem baseada em problemas, a aprendizagem baseada em projetos, a aprendizagem entre times e a sala de aula invertida [15]. Com o objetivo de formar estudantes com habilidades e conhecimentos considerados essenciais para o século XXI, foi criada uma Base Nacional Comum Curricular, também conhecida como BNCC [5], que visa assegurar aos alunos o direito de aprender um conjunto fundamental de conhecimentos e habilidades comuns, dentre as quais está a habilidade em uma segunda língua.

Segundo o levantamento feito pela British Council [7], 95% da população brasileira não fala Inglês, o que representa um percentual muito expressivo. A pesquisa mostra ainda que, da população acima

de 16 anos de idade, apenas 5,1% afirmam que têm algum conhecimento da Língua Inglesa e destes, apenas 1% apresenta algum grau de fluência. O inglês foi escolhido como segunda língua pela BNCC pelo seu caráter internacional devido à sua influência e relevância ao redor do mundo, pois “propicia a criação de novas formas de engajamento e participação dos alunos em um mundo social cada vez mais globalizado e plural” [5, 241].

As habilidades sugeridas pela BNCC e a prática de uma segunda língua podem ser trabalhadas juntamente com o desenvolvimento do pensamento computacional (PC), que é entendido como uma abordagem de resolução de problemas que pode ser automatizada e aplicada em diversas disciplinas [2]. Neste caso, o conceito de algoritmos e afins e as habilidades do PC seriam os conteúdos trabalhados, e a língua inglesa o instrumento utilizado para tal. É possível, por exemplo, criar um algoritmo em inglês, associando o aprendizado da língua inglesa aos conhecimentos necessários para a definição de uma solução algorítmica.

O Ensino de Segunda Língua Baseado em Conteúdo é exatamente uma abordagem na qual o aprendizado do conteúdo é prioridade e é ensinado em segunda língua em vez de focar em uma ementa [21]. Este trabalho conjunto do PC e da prática de uma segunda língua pode proporcionar ludicidade e criatividade, ampliando os estímulos dados aos estudantes e permitindo maior interação entre eles e a cultura digital.

Há uma carência de propostas de currículos para o ensino de programação para crianças em uma segunda língua, identificada a partir de uma revisão bibliográfica de literatura inspirada em alguns procedimentos de mapeamento sistemático. Em pesquisas realizadas nos sistemas de buscas de artigos científicos, os trabalhos encontrados, geralmente, foram realizados fora do Brasil. Nas bases brasileiras, poucos resultados foram encontrados, e os que aparecem não são para o ensino de programação em um contexto de imersão no inglês [13] [18] [24]. O objetivo desta revisão, que será apresentada em detalhes na Seção de Métodos, foi o de encontrar artigos relevantes que abordassem experiências e metodologias para aplicação do PC na educação básica.

Neste cenário, nasce a proposta deste artigo, que visa apresentar e descrever uma sequência didática de aulas de programação (em inglês) com *Scratch* para alunos dos 2º e 5º anos do Ensino Fundamental I (de 7 e 10 anos de idade respectivamente) em uma escola privada da cidade de Feira de Santana-BA, que tem um programa de educação em duas línguas. O objetivo da apresentação dessa sequência didática da aplicação do PC com o *Scratch* e a língua inglesa é o de mostrar uma alternativa curricular de se praticar uma segunda língua enquanto se aprende programação ou outro conteúdo.

Fica permitido ao(s) autor(es) ou a terceiros a reprodução ou distribuição, em parte ou no todo, do material extraído dessa obra, de forma verbatim, adaptada ou remixada, bem como a criação ou produção a partir do conteúdo dessa obra, para fins não comerciais, desde que sejam atribuídos os devidos créditos à criação original, sob os termos da licença CC BY-NC 4.0.

EduComp'22, Abril 24-29, 2022, Feira de Santana, Bahia, Brasil (On-line)

© 2022 Copyright mantido pelo(s) autor(es). Direitos de publicação licenciados à Sociedade Brasileira de Computação (SBC).

Esta sequência didática foi feita a partir de uma análise documental dos planos de aulas realizadas em 2018 no colégio onde a pesquisa foi feita e, a partir desse resgate, foi utilizada a Taxonomia de Bloom revisada para verificar se a sequência de objetivos condizia com os níveis hierárquicos do domínio cognitivo. Os resultados mostraram que, nas turmas do 2º ano (iniciantes), a maioria dos verbos usados nos objetivos de aprendizagem estavam concentrados nos primeiros níveis da hierarquia do domínio cognitivo; enquanto que nas turmas do 5º ano foi possível explorar os verbos do domínio cognitivo de forma mais distribuída, com destaque no nível “Aplicar”, do domínio cognitivo.

2 TRABALHOS RELACIONADOS

Costa et al. [9] realizaram um estudo em Portugal, no qual o Inglês como Segunda Língua era uma disciplina obrigatória no 3º ano do Ensino Fundamental I, em um período em que as escolas foram convidadas a iniciar um projeto preliminar de Introdução à Linguagens de Programação, usando Scratch. O trabalho relatou a abordagem utilizada nas aulas e algumas conclusões iniciais acerca do estudo em andamento, cujo objetivo era identificar os níveis de satisfação e receptividade, bem como a performance dos estudantes (em relação à aquisição linguística de acordo com o CEFR¹) pode efetivamente ser aperfeiçoada explorando tal abordagem inovadora e avaliando até que ponto o PC e a Programação podem ser usados em conjunto para o ensino-aprendizagem de uma língua estrangeira.

Sarasa-Cabezuelo [25], da Universidade Complutense de Madrid, na Espanha, descreve em seu artigo o potencial da linguagem de programação Scratch como ferramenta pedagógica no ensino de segunda língua. O autor fornece diversas estratégias e exemplos de implementação para suprir as necessidades que surgem nesse tipo de ensino, tais como aquisição léxica e gramatical, compreensão auditiva, compreensão escrita, pragmática e cultural. O trabalho destaca que o uso do Scratch ajuda não somente os estudantes a aprender segundas línguas, mas também serve para fazê-los se familiarizar de maneira intuitiva e divertida com programação e as novas tecnologias.

Também na Espanha, Moreno-León e Robles [22] mostram como é possível ensinar programação de computador como uma ferramenta educacional que pode ser um recurso útil para aumentar tanto os resultados acadêmicos quanto a motivação dos estudantes pela disciplina inglês. Os resultados de suas pesquisas mostram que os grupos que trabalharam com atividades de programação, ao invés de métodos tradicionais, obtiveram melhores rendimentos. Além disso, os participantes do estudo afirmaram que sentiram a programação como uma influência positiva, não somente para aprender inglês, como também para desenvolver outras habilidades como trabalhar em equipe e como “aprender a aprender”.

Veerasamy e Shillabeer [28], do Instituto Real de Tecnologia de Melbourne, no Vietnã, consideram as linguagens de programação, que têm sua sintaxe em inglês, como item fundamental para o ensino de ciência da computação, pois, através delas, os estudantes

são capazes de entender o vocabulário das palavras chaves da linguagem de programação e as mensagens de erros mostradas pelo compilador.

O presente estudo, de maneira similar aos já mencionados, relata a experiência com o uso do Scratch para o ensino de programação em um contexto de prática de uma segunda língua. Diferentemente dos trabalhos mencionados, este apresenta uma sequência didática cujos objetivos de aprendizagem são elencados de acordo com a Taxonomia de Bloom revisada, que ajuda ao educador analisar melhor a evolução do desenvolvimento cognitivo dos estudantes.

3 FUNDAMENTAÇÃO TEÓRICA

3.1 Pensamento Computacional

Em 2006, o termo Pensamento Computacional (PC) começa a ganhar notoriedade quando Jeannette Wing o define como um conjunto de atitudes e habilidades aplicáveis por qualquer pessoa que quer aprender e usar, não somente por cientistas da computação [29]. Em 2014, Wing atualiza o conceito como um conjunto de métodos para a resolução de problemas que envolvam expressar suas soluções de modo que possam ser executados por processadores de informações - humanos ou computadores [30]. A autora compara o PC como uma capacidade tão importante quanto as de ler, escrever e realizar cálculos matemáticos e que, por isto, deveria ser considerado também como habilidade analítica de toda criança.

Barr e Stephenson [2] apresentam formas de contemplar o PC nos ensinos primários e secundários. A abordagem de resolução de problema do PC pode ser automatizada e aplicada em diversas disciplinas. Os autores elencam 9 habilidades que definem o PC: (1) Coleção de dados, (2) Análise de Dados, (3) Representação de Dados, (4) Automação, (5) Paralelização, (6) Simulação, (7) Decomposição, (8) Abstração e (9) Pensamento Algorítmico.

Não há um consenso sobre qual é o conjunto de habilidades que formam o PC, alguns autores trazem conceitos semelhantes aos de Barr e Stephenson [2] e outros diferentes. Csizmadia e outros [10] também abordam a (1) Abstração, a (2) Decomposição e o (3) Pensamento Algorítmico, mas adicionam (4) Generalização e (5) Avaliação.

Brackmann [4] resume as habilidades do PC denominando-as de “Quatro Pilares do Pensamento Computacional” (as mesmas contidas em um dos eixos propostos na Base Nacional Comum Curricular brasileira [5]), sendo eles: (1) **Decomposição**: envolve identificar um problema complexo e quebrá-lo em pedaços menores e mais fáceis de gerenciar; (2) **Reconhecimento de Padrões**: problemas podem ser analisados individualmente com maior profundidade, identificando outros parecidos que já foram solucionados anteriormente; (3) **Abstração**: foca apenas nos detalhes que são importantes, enquanto informações irrelevantes são ignoradas, e (4) **Algoritmos**: passos ou regras simples que podem ser criados para resolver cada um dos subproblemas encontrados.

3.2 Ensino de Segunda Língua Baseado em Conteúdo

O Ensino de Segunda Língua Baseado em Conteúdo (ESLBC), de acordo com Lyster [21], é uma abordagem didática em que um conteúdo não linguístico, incluindo disciplinas como Ciências, Matemática ou Programação, é ensinado aos estudantes através de um

¹Common European Framework of Reference for Languages (Quadro Europeu Comum de Referência para as Línguas) - <https://www.cambridgeenglish.org/exams-and-tests/cefr>

idioma que não é o seu nativo, de modo que, enquanto eles estão aprendendo o conteúdo curricular, também estão aprendendo uma outra língua.

Snow, Met, and Genesee (*apud* Lyster [21]) explicam que o desenvolvimento linguístico e o cognitivo andam lado a lado, enquanto que os métodos tradicionais tendem a separá-los. ESLBC é frequentemente referenciado como uma abordagem “dois em um”, pois os alunos destes programas aprendem o conteúdo de uma disciplina e a segunda língua ao mesmo tempo [20]. Há estudos que mostram que a maioria das pessoas que aprendem uma segunda língua através da imersão, de fato, tem êxito em dominar também o conteúdo como se estivessem aprendendo na sua língua mãe, sem prejuízos à primeira língua [26] [16].

3.3 Scratch

Scratch² é uma das ferramentas mais populares mundialmente para a prática da aprendizagem criativa. É uma linguagem de programação visual baseada em blocos (que remetem às peças de Lego) desenvolvida pelo *Lifelong Kindergarten Group*, grupo de pesquisa liderado por Mitchel Resnick no MIT Media Lab, organização que dá continuidade às ideias construcionistas do Logo [12].

A plataforma ajuda crianças não somente a aprender a programar, mas também proporciona um ambiente educacional com uma missão mais ampla. Permite que jovens de diversos contextos tenham a oportunidade de pensar criativamente, raciocinar sistematicamente e trabalhar colaborativamente [23]. Está disponível tanto na versão *on-line* quanto na *off-line*. Recentemente, com a versão Scratch 3.0, é possível acessá-lo também em dispositivos móveis.

3.4 Taxonomia revisada de Bloom

A Taxonomia de Bloom [3] é uma classificação de objetivos educacionais. Durante a elaboração de um currículo, professores usam diferentes termos para expressar os objetivos de aprendizagem, por exemplo, “compreender”, “internalizar conhecimento”, dentre outros. A Taxonomia de Bloom permite que essa classificação de objetivos de aprendizagem seja feita de forma padronizada considerando um conjunto de três modelos hierárquicos em níveis de complexidade e especificidade, a partir dos domínios: cognitivos, emocionais e psicomotores da aprendizagem.

Na versão original da taxonomia, o domínio cognitivo, mais frequentemente utilizado, é composto por seis categorias (em negrito) e os verbos recomendados em cada uma, a seguir [8]: **(1) Conhecimento**: apontar, definir, enunciar, inscrever, marcar, recordar, relatar, repetir, nomear, sublinhar; **(2) Compreensão**: descrever, discutir, esclarecer, examinar, explicar, expressar, identificar, localizar, narrar, reafirmar, traduzir, transcrever; **(3) Aplicação**: aplicar, demonstrar, dramatizar, empregar, ilustrar, interpretar, praticar, traçar, usar; **(4) Análise**: analisar, calcular, classificar, comparar, contrastar, criticar, debater, diferenciar, distinguir, examinar, provar, investigar, experimentar; **(5) Síntese**: articular, compor, constituir, coordenar, criar, dirigir, reunir, formular, organizar, planejar, propor, esquematizar, e **(6) Avaliação**: apreciar, avaliar, eliminar, escolher, estimar, julgar, ordenar, preferir, selecionar, validar, valorizar.

Na versão revisada da taxonomia [19], todas as categorias foram expressas como verbos em vez de substantivos, conforme mostra

a Figura 1. Conhecimento foi alterado para Lembrar/Recordar, os demais se mantiveram, exceto por Síntese que foi renomeada para Criar. Além disso, Criar tornou-se o nível mais alto no sistema de classificação, trocando de lugar com Avaliar. A versão revisada agora é nessa ordem: Recordar, Compreender, Aplicar, Analisar, Avaliar, Criar.



Figura 1: Taxonomia revisada. Fonte: [27]

Araújo et. al [11] apontam que a taxonomia de Bloom tem um papel fundamental como instrumento para a análise e a avaliação do ensino-aprendizagem e que pode também ser aplicada no ensino de programação com Scratch. A taxonomia serve como um recurso de referência para orientar o ensino a partir de uma hierarquia crescente de habilidades cognitivas. Os autores supracitados elaboraram e disponibilizaram um plano de curso, planos de aulas e cadernos de atividades elaborados segundo a taxonomia de Bloom.

4 MÉTODOS

Esta pesquisa tem caráter exploratório, pois permite, através do levantamento bibliográfico, uma maior proximidade entre o pesquisador e o tema pesquisado, com vistas a torná-lo mais explícito ou a construir hipóteses [17]. Pode ser classificada como *ex-post facto*, visto que objetiva investigar acontecimentos a partir de fatos passados, ou seja, o pesquisador não possui o controle direto sobre as variáveis independentes, pois suas manifestações já ocorreram [14]. Neste trabalho, especificamente, os autores já aplicaram o ensino de conteúdo (Programação) a partir do uso da língua inglesa e do *Scratch*, e fará, portanto, a descrição e a análise dos procedimentos já realizados e de alguns novos com base na análise documental dos planos de aulas e projetos. Isto significa que os módulos foram abstraídos inicialmente dos planos de aula anteriores e depois refinados à luz da taxonomia revisada de Bloom para explicitar os objetivos de aprendizagem.

Para embasar as propostas metodológicas deste trabalho, compreender as possibilidades e as estratégias possíveis de serem replicadas, foi feita uma revisão bibliográfica de literatura inspirada em alguns procedimentos de mapeamento sistemático. O objetivo da revisão foi o de encontrar artigos relevantes que abordassem experiências e metodologias para aplicação do PC na educação básica.

Foram considerados trabalhos publicados no período de 5 anos (de 2014 a 2019), nas plataformas de pesquisas e eventos científicos brasileiros e internacionais, em português e em inglês. As

²<https://scratch.mit.edu>

buscas foram realizadas nas seguintes bases: Revista Brasileira de Informática na Educação, Simpósio Brasileiro de Informática Na Educação, *Workshop* de Informática na Escola, *Workshops* do Congresso Brasileiro de Informática na Educação, Revista Eletrônica de Iniciação Científica em Computação, *Google Scholar*, e *Association for Computing Machinery Digital Library*, que indexa publicações de eventos internacionais como *Special Interest Group on Computer Science Education* (SIGCSE), *Innovation and Technology in Computer Science Education* (ITICSE) e *International Computing Education Research* (ICER). Foram definidas as seguintes *strings* de busca em qualquer parte do texto:

- (1) Primeiro grupo de String (GS1): (“pensamento computacional OR “computational thinking” OR “aprendizagem criativa” OR “creative learning”) AND (programação OR programming OR coding OR scratch) AND (“ensino fundamental” OR “elementary school” OR crianças OR children OR kids).
- (2) Segundo grupo de String (GS2): (programação OR computer programming OR scratch) AND (inglês OR “segunda língua” OR bilíngue OR “duas línguas” OR english OR “second language” OR bilingual OR “dual language”).
- (3) Terceiro grupo de String (GS3): “segunda língua baseada em conteúdo” OR “content based language teaching”.

A fim de selecionar os estudos que tivessem relação com a pesquisa, definiram-se os seguintes critérios de inclusão: CI1 - estudos que falem sobre Pensamento Computacional OU Aprendizagem Criativa; CI2 - estudos que falem sobre programação para crianças do ensino fundamental de escolas públicas ou privadas; CI3 - estudos que falem sobre educação em duas línguas; CI4 - estudos que abordem metodologias ou estratégias para o ensino de programação para crianças.

Os critérios de exclusão considerados foram: CE1 - publicações que não estejam acessíveis para leitura; CE2 - publicações em que o idioma seja diferente dos definidos; CE3 - publicações que não contenham informações completas sobre o mesmo (autor, ano de publicação, etc); CE4 - publicações que estejam fora do contexto educacional; CE5 - estudos que não contenham resumo; CE6 - estudos com datas anteriores a 5 anos; CE7 - estudos duplicados; CE8 - publicações que não satisfaçam a nenhum critério de inclusão; CE9 - estudos que fogem do domínio da proposta deste trabalho. Após a aplicação destes filtros, 25 artigos foram selecionados e lidos integralmente para dar suporte teórico a este trabalho.

Para melhor compreensão dos procedimentos metodológicos, serão abordados os aspectos relativos ao perfil dos estudantes, organização didática pedagógica e análise dos projetos desenvolvidos.

4.1 Perfil dos estudantes

Em relação ao lócus da pesquisa, as aulas aconteceram entre os meses de fevereiro e novembro de 2018, em uma escola privada, situada na cidade de Feira de Santana, Bahia. O estabelecimento de ensino tem um programa de Educação em Duas Línguas (Português como língua nativa e Inglês como segunda língua). Os alunos têm um momento semanal no período do contraturno, no qual aprendem e praticam os conceitos de Educação Financeira, Matemática e Programação. Todas essas aulas são 100% em inglês e têm o ESLBC como prática. Isto quer dizer que durante as aulas de Programação,

por exemplo, a criança estará vendo um conteúdo de Computação em Inglês e os docentes trabalhando ambos em uma única aula.

Os estudantes envolvidos nestas atividades foram todos aqueles matriculados nos 2º e 5º anos do Ensino Fundamental I (participação discente obrigatória), com idades de 7 e 10 anos, respectivamente. Cada série contou com 4 turmas de, aproximadamente, 25 alunos em cada uma. Cada aula tinha duração de 50 minutos. A principal ferramenta utilizada para o ensino-aprendizagem de programação foi o *Scratch 2.0 offline*. Quanto ao gênero dos participantes, do total de 98 alunos do 2º ano (separados em 4 turmas), 40,8% eram do sexo masculino e 59,2% do sexo feminino. Do total de 97 alunos do 5º ano (separados em 4 turmas), 55,7% eram do sexo masculino e 44,3% do sexo feminino.

4.2 Organização Didática Pedagógica

O ano letivo iniciou-se em fevereiro e foi até o final de novembro, e foi dividido em 4 unidades. Cada unidade teve, aproximadamente, 10 aulas. Os temas dos projetos foram escolhidos tendo como base o currículo interdisciplinar de cada série, de forma que um projeto pudesse explorar o conteúdo de ciências ou matemática e inglês, por exemplo. A decisão das temáticas contou com a colaboração da coordenação pedagógica e/ou professores das outras disciplinas. A elaboração do planejamento das aulas foi feita baseando-se na Taxonomia de *Bloom* revisada para categorizar os objetivos de aprendizagem de forma padronizada.

Durante as aulas, o professor reunia os estudantes sentados no centro da sala, mais próximo da tela onde o projeto era apresentado. Nos minutos iniciais da aula, era introduzido o projeto que seria iniciado naquela aula e continuado nas semanas seguintes. O professor fazia uma demonstração do funcionamento do projeto e convidava alguns alunos para experimentá-lo. Antes de iniciar a explicação do que seria feito naquele dia, era feita uma contextualização do que os alunos conseguiam enxergar naquele projeto através de perguntas que o relacionassem com os conteúdos das outras disciplinas. Toda a comunicação era sempre feita em inglês. Caso não soubessem determinada palavra, poderiam perguntar como se pronunciaria a palavra em inglês, através da pergunta “*How do you say () in English?*”.

Em seguida, era explicado o conjunto de instruções que eles usariam para programar determinado personagem. Durante este momento, o professor pedia para que alguns estudantes explicassem algum comando que já fosse de conhecimento prévio. Desta forma, os alunos revisariam os assuntos de programação e/ou *Scratch*, enquanto praticavam a língua inglesa. Finalizada a parte de explicação da atividade do dia, os alunos dirigiam-se para seus respectivos computadores. Como eles utilizavam a versão *offline* do *Scratch*, era importante lembrar onde o projeto havia sido salvo. Para tanto, cada computador tinha uma numeração. Para evitar que os alunos sentassem sempre próximos dos mesmos grupos, os notebooks eram distribuídos na sala de forma aleatória no início de cada aula.

Os estudantes dispunham de 25 a 35 minutos para o desenvolvimento do projeto. Durante este período, eles podiam consultar os *scripts* projetados na tela, fazer perguntas ao professor, consultar ou ajudar os colegas e, em algumas aulas, até desenvolver uma espécie de monitoria. O aluno que acabasse com muita antecedência poderia caminhar pela sala e dar assistência aos colegas ou desenvolver

projetos próprios. Este também era o momento em que o professor tirava as dúvidas individualmente.

4.3 Análise dos projetos desenvolvidos

Os projetos desenvolvidos nos 2º e 5º anos foram catalogados em um portfólio e classificados a partir dos critérios: Objetivo de Aprendizagem; Conteúdo de *Scratch*; Habilidades do Pensamento Computacional; Conceitos e Práticas Computacionais; e Tópicos da Língua Inglesa. A análise e a classificação dos projetos foram feitas pelo autor e revisadas pela coautora deste trabalho.

Vale destacar que, dentre os itens listados acima, o tópico “Objetivo de Aprendizagem” foi feito a partir de uma análise documental dos registros de planos de aula elaborados em 2018 enquanto professor. Entretanto, foi necessária uma reformulação dos mesmos à luz da Taxonomia revisada de Bloom para esta pesquisa, prática que não havia sido adotada anteriormente. Os demais itens, entretanto, foram todos classificados no tempo presente em uma perspectiva de pesquisador para este trabalho.

Em cada unidade, foram executados, em média, 3 projetos. Cada projeto levou cerca de 3-5 aulas para ser concluído. Cada aula tinha duração de 50 minutos e, para cada turma, acontecia semanalmente.

4.3.1 Objetivo de Aprendizagem. Este item diz respeito à identificação do que se deseja que os estudantes aprendam em cada aula de cada um dos projetos. Os verbos destacados com uma cor indicam em qual nível da Taxonomia revisada de Bloom ele se enquadra: Recordar, Compreender, Aplicar, Analisar, Avaliar ou Criar, conforme mostra a Figura 2.



Figura 2: Níveis de aprendizagem da Taxonomia revisada de Bloom.

As cores de cada verbo foram escolhidas pelos autores deste trabalho de forma aleatória. O objetivo desta identificação foi o de associar mais claramente os objetivos de aprendizagem ao modelo de classificação da Taxonomia proposta por Bloom, facilitando a percepção dos níveis de complexidade dos objetivos planejados e desejados, a partir desta organização hierárquica (Taxonomia Bloom).

Todos os verbos escolhidos desta Taxonomia fazem parte do nível Cognitivo, que trata da aprendizagem intelectual, dessa forma, os domínios Afetivos e Psicomotor não foram analisados neste trabalho. Ainda nos Objetivos, foi inserido um código para categorizar o tipo de conteúdo trabalhado com os estudantes: CP - Conteúdo de Programação; CI - Conteúdo Interdisciplinar, contemplando o Inglês e as outras disciplinas; CS - Conteúdo de *Scratch*.

4.3.2 Conteúdo de Scratch. À medida em que as crianças desenvolvem seus projetos no *Scratch* e buscam pelos comandos, elas se envolvem com conceitos computacionais que são comuns em muitas linguagens de programação [6], como também com contextos que não são de programação. Neste sentido, o item *Conteúdo de Scratch* foi criado na tabela de sequência didática de projetos para separar os conceitos, os blocos de comandos e as funcionalidades específicas do *Scratch* das demais habilidades computacionais.

Embora o ambiente de programação aborde vários elementos que podem ser classificados como assuntos de programação, ele possui itens específicos da plataforma que devem ser ensinados como, por exemplo, a criação e edição de atores e cenários, ferramentas de desenho e texto, efeitos sonoros, entre outros. Apesar deste trabalho tratar de aulas dadas em inglês, os conjuntos de blocos utilizados nos projetos serão descritos em português para facilitar a replicação das ideias por outros educadores que tenham interesse.

4.3.3 Habilidades do Pensamento Computacional. Neste item, foi utilizado o conjunto de habilidades listados por Brackmann [4]:

- (1) decomposição;
- (2) reconhecimento de padrões;
- (3) abstração;
- (4) algoritmos.

Em cada projeto, foi possível notar que pelo menos uma das habilidades foi explorada; em alguns, todas as habilidades foram utilizadas. Outros autores [10] [1] elencam um conjunto maior de habilidades. Entretanto, algumas delas são bem semelhantes entre si, apesar de usarem terminologias diferentes; e algumas não se enquadram nos projetos realizados neste trabalho, como Análise de Dados [1], por exemplo.

4.3.4 Conceitos e Práticas Computacionais. Este item foi adicionado às tabelas de sequência didática de projetos como uma extensão das habilidades do PC e atributos inerentes à Programação. Brennan e Resnick [6] desenvolveram um *framework* para avaliação do desenvolvimento do PC no contexto de programação em *Scratch*. A estrutura proposta por eles está dividida em 3 eixos principais: “conceitos computacionais”, “práticas computacionais” e “perspectivas computacionais”. Os conceitos computacionais são aqueles com os quais os estudantes se envolvem enquanto estão programando. As práticas são aquelas desenvolvidas enquanto os estudantes programam. E as perspectivas são formadas por eles a partir do que percebem do mundo ao seu redor e sobre si mesmos. Este último não foi considerado para classificação dos projetos deste trabalho. Portanto, durante a análise dos projetos, foram utilizados os eixos conceitos e práticas do *framework* que são detalhados a seguir:

- (1) **conceitos:** sequência, repetições, paralelismo, eventos, condicionais, operadores e variáveis;
- (2) **práticas computacionais:** iteração, teste e depuração, reuso, abstração e modularização.

Ambos os eixos serão listados em um único item das tabelas de projetos.

4.3.5 Tópicos da Língua Inglesa. A prática do inglês foi constante durante todas as aulas. As quatro habilidades do idioma foram desenvolvidas: *reading* (ler), *writing* (escrever), *listening* (ouvir) e *speaking* (falar). Não foi feita uma mensuração de quanto cada uma delas foi trabalhada por aula, mas pode-se dizer que, em algum momento da aula, houve o exercício destas de alguma maneira: *reading*, ao ler as instruções apresentadas pelo professor e ao procurar pelos blocos de comandos no *Scratch*; *writing*, ao criar os diálogos dos personagens, ao criar variáveis e ao usar ferramentas de texto do *Scratch*; e, majoritariamente, o *listening* e o *speaking*, ao compreender e ao responder o professor e os colegas. Tendo em

vista que o treinamento dessas quatro habilidades foi constante, no item Tópicos da Língua Inglesa das tabelas de sequência didática de projetos será apresentado apenas o vocabulário utilizado em cada projeto.

5 RESULTADOS E DISCUSSÕES

Esta parte do trabalho descreve parcialmente a sequência didática (Disponível em: <https://bit.ly/2W5XHCc>) realizada durante a concepção dos planos de aula e dos projetos desenvolvidos com os estudantes e suas etapas de construção; além da evolução dos alunos durante as aulas.

Serão descritos um projeto do 2º ano e um do 5º ano, que serão utilizados como exemplos, de forma a descrever o funcionamento, seus objetivos de aprendizagem, como as habilidades do PC, os conceitos e as práticas computacionais foram explorados, bem como a prática da abordagem de ESLBC. O quadro da Figura 3 mostra um recorte de como o Levantamento de Projetos está estruturado e quais itens foram analisados em cada um deles.

Essa parte da análise e detalhamento servirá como um guia para a compreensão do portfólio de projetos feitos no *Scratch*, descritos e classificados na sequência didática de projetos. Além disso, em paralelo, será apresentada a avaliação qualitativa, a partir da experiência do pesquisador, em relação à sua percepção do processo de ensino enquanto educador, e também em relação à avaliação da complexidade cognitiva trabalhada com os projetos.

5.1 2º ano

Os estudantes do 2º ano produziram um total de 13 projetos no *Scratch* ao longo do ano. Será destacado aqui o projeto “I can” (Figura 4).

O funcionamento do projeto é relativamente simples. Nessa animação, há 3 atores: um menino, um papagaio e uma bailarina. Cada ator faz um movimento repetidamente ao ser clicado e expressa sua habilidade através de uma frase:

- Menino: “I am boy. I can walk.”
- Papagaio: “I am ballerina. I can dance.”
- Bailarina: “I am parrot. I can fly.”

Antes de descrever sobre o processo de criação de um projeto como este, é importante explicar sobre a motivação da escolha do seu tema, como e por que ele foi criado. Conforme explicado na seção Métodos, os projetos feitos durante as aulas precisavam ter relação com alguma outra disciplina.

No período de planejamento, estava previsto no currículo que as turmas do 2º ano teriam aula de inglês sobre o verbo modal “can/can’t” para expressar habilidades. Então, como forma de praticar essa estrutura gramatical, e abordar um conteúdo de maneira interdisciplinar nas aulas de programação, fez-se uso do conteúdo da aula de Inglês, mas que poderia ser de qualquer outra disciplina como ciências e matemática, por exemplo.

O projeto escolhido foi feito para ser apresentado à turma em meados da primeira unidade, momento em que os alunos ainda teriam pouca experiência com a programação e o *Scratch*. Pensando nisso, a animação teria que ser simples, mas, ao mesmo tempo, interessante e atraente para os alunos, com movimento dos personagens e um diálogo de fácil compreensão. Sendo assim, foi criado este projeto com 1 *backdrop* (cenário), 3 *sprites* (atores) e apenas 5

Figura 3: Projeto “Food Rain” e Itens analisados no levantamento dos projetos

<p>Projeto: Food Rain</p> <p>Objetivo de Aprendizagem: Demonstrar que é capaz de desenhar os próprios atores (CS). Usar comandos de repetição; manipular variáveis (inicialização, incremento e decremento); gerar números aleatórios; usar troca de mensagens entre os comandos-atores; usar condicionais e operadores (< e >) (CP). Compor novos scripts baseando-se na reutilização de códigos (CP). Reproduzir efeitos sonoros (CS). Duplicar atores (CS).</p> <p>Conteúdo de Scratch: Aparência: mude para a fantasia (); Eventos: quando eu receber (); transmita (); quando Bandeira for clicado; quando a tecla () for pressionada; quando este ator for clicado. Controle: se () então; sempre; espere () seg; crie clone de (); quando eu começar como um clone. Variáveis: mude () para (); adicione () a (). Som: toque o som (). Sensores: tocando em () ?. Operadores: número aleatório entre () e (); (<()); (>()) (relacionais menor/ maior que).</p> <p>Habilidades do Pensamento Computacional: Algoritmos; Reconhecimento de Padrões; Abstração; Decomposição.</p> <p>Conceitos e Práticas Computacionais: Eventos; Sequências; Repetições; Condicionais; Operadores; Variáveis; Paralelismo; Iteração; Abstração e Modularização.</p> <p>Tópicos da Língua Inglesa: Vocabulário: healthy and unhealthy foods.</p> <p>Descrição: O jogo consiste em comer somente os alimentos saudáveis que estão caindo em direção ao chão. Para cada item saudável, pontos são adicionados ao score. Para cada item não saudável, o jogador perde 1 ponto de vida. A partida vai ficando mais rápida e novos itens vão surgindo de acordo com a pontuação. O personagem parece um Pac-Man e é controlado usando as setas (esquerda e direita) do teclado. Ele fica abrindo e fechando a boca constantemente e reproduz um efeito sonoro ao comer um item indicando quando se ganha ou perde pontos. Os alimentos são duplicados e vão surgindo em posições aleatórias do topo e deslocam-se na vertical.</p>
--

blocos de comandos (os mesmos tipos para todos os personagens), como mostra a Figura 5). O que muda de um bloco para o outro é apenas a frase dita por cada um dos personagens. Nesta parte já é possível perceber algumas práticas do PC como algoritmos e reconhecimento de padrões.

Uma vez que o projeto tenha sido finalizado pelo professor, ele era validado pela coordenação do colégio para verificar se o conteúdo condizia com a faixa etária e com o currículo dos estudantes.

Em sala de aula, antes de iniciar a explicação de como fazer a animação, foi feita uma apresentação para os estudantes, seguida de uma contextualização do assunto trabalhado. Para isto, o professor perguntou-lhes o que viam na tela e cada estudante dizia alguma coisa em inglês de acordo com as suas percepções sobre o projeto. Esta rotina de pensamento estimulava os estudantes a fazer observações cuidadosas e interpretações profundas. Isso ajudava a



Figura 4: Animação “I can” feita no Scratch.



Figura 5: Blocos de comandos utilizados na animação “I can”.

estimular a curiosidade e a “preparar o terreno” para a investigação [31]. Essa estratégia foi utilizada neste e nos outros projetos, como forma de revisão e prática de vocabulário da língua inglesa, além de aguçar a curiosidade sobre o quê e como eles fariam a animação ou o jogo apresentado.

A Taxonomia de Bloom revisada, no domínio cognitivo, apresenta 6 níveis por ordem crescente de complexidade: recordar, compreender, aplicar, analisar, avaliar e criar. Cada um sugere uma lista de verbos que ajudam o educador a refletir a definir os objetivos de aprendizagem. Neste projeto, todos os verbos utilizados fizeram parte do primeiro nível: recordar.

- (1) **identificar** os personagens apresentados no projeto e nomeá-los (CI);
- (2) **apontar** as habilidades de cada personagem (CI);
- (3) **recordar** o processo de escolha de atores e cenários (CS);
- (4) **usar** comandos de eventos para iniciar outros comandos, de diálogos entre os atores, e de troca de fantasias (CS e CP).

Utilizar os verbos recomendados ajuda o professor a elaborar melhor os objetivos de aprendizagem, pois, além de criar uma padronização, permite validar se as etapas de aprendizagem estão seguindo a hierarquia “correta”. Para as aulas deste projeto, foi primordial que todos os objetivos estivessem no primeiro nível,

pois os estudantes estavam na primeira unidade de uma disciplina apresentada a eles pela primeira vez.

O processo de aprendizagem do conteúdo de Scratch foi além da compreensão do funcionamento de cada bloco de comando utilizado na animação. Para o 2º ano, foi necessário algum tempo para explorar o ambiente de programação. Este processo foi feito de forma incremental ao longo do ano. O que era essencial, até então, era saber como escolher os *sprites* e *backdrops*, e diferenciar entre os espaços dos *scripts* de cada um.

Partindo para a etapa de programação dos atores, o professor explicou cada comando individualmente, sempre demonstrando seu funcionamento no próprio programa - clicando no bloco e vendo o seu efeito no “palco” ou convidando alguns estudantes para fazer demonstrações desplugadas. Por exemplo, o bloco “when this sprite clicked” (quando este ator for clicado - Figura 5) teve seu funcionamento ilustrado convidando 3 estudantes à frente da sala. Eles ficaram de frente para os colegas e executaram alguma tarefa quando o professor os tocaram: *clap your hands* (bata palma), *jump* (pule), *turn around* (gire ao redor de si mesmo). Com isto, eles aprenderam sobre eventos e sequências de forma implícita. Alguma atividade lúdica foi feita para explicar o restante do programa.

Tais atividades lúdicas envolviam a explicação através de demonstrações com alguma ação de computação desplugada, que é uma forma de trazer algum conceito ou problema computacional sem fazer uso de computador ou equipamento eletrônico. Para ensinar sobre variáveis, por exemplo, é possível escrever uma tabela de pontuação no quadro com os nomes dos estudantes e mostrar que o valor de cada um pode ser alterado ao longo do tempo.

No projeto “I can”, foi possível trabalhar 2 habilidades do PC: (a) algoritmos e (b) reconhecimento de padrões. Os estudantes compreenderam a ideia de algoritmo ao entender o que é um comando e a execução sequenciada dos comandos (um após o outro). Inclusive, o bloco “wait 0.5 seconds” (espere 0.5 seg, Figura 5) foi utilizado para que os estudantes pudessem perceber a transição das fantasias dos personagens. Sem ele, o *loop* feito pelo bloco “forever” (sempre, Figura 5) executaria os comandos tão rapidamente que eles não poderiam ver os personagens andando, voando ou dançando. Eles tiveram também a oportunidade de fazer o reconhecimento do padrão de comandos utilizados em todos os *sprites*, e viram que seria possível fazer reusos dos mesmos, apenas modificando as frases inseridas em cada um, de acordo com o ator selecionado.

Alguns conceitos e práticas computacionais explorados no projeto já foram apresentados nos parágrafos anteriores, mas vale a pena listá-los, agrupá-los e evidenciá-los novamente: eventos, repetição, sequências, e mais dois que ainda não foram abordados, paralelismo e iteração. Os alunos puderam compreender o que é paralelismo ao ver que um ator não interrompe seus comandos quando o outro inicia; todos continuam se movimentando ao mesmo tempo. A prática de iteração foi feita durante todas as aulas do projeto, pois os alunos repetiram diversas etapas até chegarem ao resultado final.

A língua inglesa foi praticada em todos os momentos das aulas de programação, desde a chegada dos estudantes através de cumprimentos, passando pela discussão sobre o funcionamento do projeto, como desenvolvê-lo, a sua criação, ao mapear os blocos no Scratch, até os momentos em que os alunos tiravam dúvidas individuais. Seguindo a metodologia de ESLBC, o tema das aulas

deste projeto foi o verbo modal *Can* para expressar habilidades, na parte de conteúdo interdisciplinar, pois os conteúdos principais são aqueles relacionados à programação listados no parágrafo anterior. Na parte inicial de discussão, os alunos tiveram que falar alguma habilidade que eles eram capazes ou não de fazer. Por exemplo:

Estudante 1: *I can ride a bike.* (Eu sei/posso andar de bicicleta.)

Estudante 2: *I can't swim.* (Eu não sei/posso nadar.)

O papel do professor não era ensinar a forma gramatical para os alunos na aula de programação, pois isto já era feito nas aulas de inglês. Na aula de programação, eles viam um conteúdo não linguístico (*Scratch* e programação), mas tinham a oportunidade de praticar a segunda língua por meio de imersão, diálogos, leitura e escrita.

Por fim, a última etapa das aulas deste projeto seria salvar o arquivo do *Scratch*. Entretanto, neste primeiro projeto, isto ainda não foi possível, pela pouca familiaridade dos estudantes com o *software* ou, até mesmo, com o computador de forma geral. O professor e a assistente os ajudaram nesta tarefa, que foi aprendida ao longo dos outros projetos desenvolvidos até o final do período.

5.2 5º ano

Os estudantes do 5º ano produziram um total de 9 projetos no *Scratch* ao longo do ano, todos descritos na sequência didática de projetos. Será dado destaque aqui ao projeto “Food Rain” (Figura 6) com uma análise completa dele.

O funcionamento do projeto pode ser verificado no item **Descrição** apresentado no quadro da figura 3. A motivação da escolha do tema do jogo se deu em função da importância de se criar um projeto interdisciplinar a partir dos currículos de ciências e inglês, que estavam abordando o assunto “alimentos saudáveis e não saudáveis”.



Figura 6: Jogo “Food Rain”.

Na primeira aula do projeto, o professor fez uma demonstração de como o jogo funcionava e convidou alguns alunos para testá-lo. Em seguida, fez uma contextualização do tema do projeto com a participação de todos. É importante ressaltar que a comunicação entre alunos e professor era sempre em inglês. Os estudantes tiveram que

listar alguns alimentos que consideravam saudáveis e outros não saudáveis. Discutiram, também, o que aconteceria no jogo quando o personagem comesse um alimento saudável e um não saudável. Neste momento, eles perceberam e apontaram (momento de descoberta) que, para cada tipo de alimento, uma pontuação/penalidade diferente era aplicada.

Foi-lhes perguntado também sobre as posições iniciais de cada *sprite* e o seu deslocamento no cenário: “eles aparecem em uma posição fixa ou aleatória?”; “Movem-se na vertical ou na horizontal?”; “Em qual sentido?”; “Como podemos programar essas coisas?”. Após responderem esses questionamentos e terem uma visão global do projeto, deu-se início ao processo de explicação e ao detalhamento do jogo por parte do professor, seguido da efetiva criação pelos estudantes.

Os objetivos de aprendizagem deste projeto também foram definidos a partir da Taxonomia de Bloom revisada, no domínio cognitivo, e os verbos utilizados fazem parte do nível Aplicar, destacados no quadro da Figura 3 na cor amarela, de acordo com a Figura 2.

No início de todas as etapas de criação do projeto, o professor mostrava uma parte do jogo e questionava a turma sobre como seria possível fazer aquele pedaço do projeto. Devido à experiência prévia dos estudantes com o *Scratch* (no 3º e no 4º anos), esse *brainstorming* era bem produtivo, com o surgimento de diferentes ideias. Por exemplo, os alunos foram questionados sobre como desenhar o *Pac-man*. Alguns disseram que poderiam desenhar um círculo e utilizar a ferramenta de borracha para apagar/desenhar a boca; outros disseram que poderiam fazer a boca desenhando um triângulo verde da mesma cor do fundo do cenário. Apesar de o professor já ter um roteiro pronto para mostrá-los como fazer, os estudantes eram sempre estimulados a usar suas próprias ideias, desde que o professor as validasse, pois, às vezes, algumas sugestões não funcionavam para a proposta em questão.

Além de utilizarem as ferramentas de desenho para criar os próprios *sprites*, os estudantes aprenderam alguns conteúdos específicos do *Scratch* listados no item **Conteúdo de Scratch** da Tabela 3:

- (1) aparência: mude para a fantasia ();
- (2) eventos: quando eu receber (); transmita (); quando Bandeira for clicada; quando a tecla () for pressionada; quando este ator for clicado;
- (3) controle: se () então; sempre; espere () seg; crie clone de (); quando eu começar como um clone;
- (4) variáveis: mude () para (); adicione () a ();
- (5) som: toque o som ();
- (6) sensores: tocando em () ?;
- (7) operadores: número aleatório entre () e (); ()<(); ()>() (relacionais menor/maior que).

Em relação ao PC, os estudantes puderam praticar as 4 habilidades durante o desenvolvimento do jogo: **(1) Algoritmos**: ao entender as regras do jogo e definir os passos para a sua execução. **(2) Reconhecimento de Padrões**: percebendo que a mesma sequência de blocos utilizada para programar um *sprite* poderia ser reutilizada em outras partes do jogo (fazendo leves adaptações). Na Figura 7, é possível ver os comandos utilizados para programar o *sprite 'Apple'*. Os mesmos comandos foram utilizados para programar os demais *sprites* de comida do jogo, ajustando o valor da

variável *Score* para que cada tipo de alimento tivesse uma pontuação diferente. No bloco *glide*, também foi preciso ajustar a quantidade de segundos para fazer com que um tipo de *sprite* caísse mais rapidamente do que o outro. (3) **Abstração**: ao programar um *sprite* por vez, o estudante pôde focar apenas no funcionamento de um e ignorar os demais, até que todos estivessem programados ou que já pudessem ser testados. (4) **Decomposição**: mesmo programando apenas um *sprite*, o aluno precisava quebrar os *scripts* em partes menores para entender seu funcionamento. Por exemplo, o conjunto de blocos 1 da Figura 7 era responsável por fazer o ator aparecer em uma posição aleatória no topo da tela e deslizar lentamente para baixo. Já o conjunto de blocos de comandos 2 era responsável por gerar a pontuação quando o *sprite* tocasse no *Pacman*. Os estudantes também desenvolveram os **conceitos e práticas computacionais** listados no quadro da Figura 3.

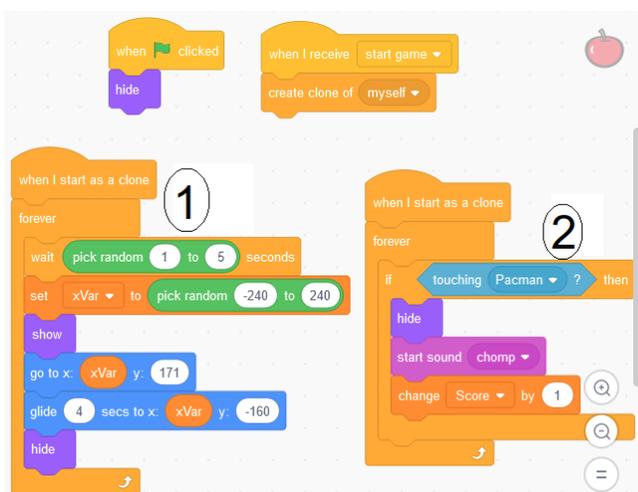


Figura 7: Parte dos blocos de comandos do jogo “Food Rain”.

A prática da língua inglesa, por meio da metodologia de ESLBC no 5º ano, se deu de forma bem semelhante nas turmas de 2º ano. O que diferiu foi o tema do projeto e o nível de complexidade da discussão por parte dos estudantes. Listamos vários nomes de alimentos e discutimos, sempre em inglês, o porquê de um tipo de comida ser saudável ou não. No próprio vocabulário de *Scratch* e programação, termos como *loops*, *random numbers*, *operators*, *variables*, *increase*, *decrease*, *broadcast*, *etc* já se apresentam em inglês. Mais uma vez, foi possível perceber que, mesmo os alunos não estando em uma aula formal de inglês, eles puderam lembrar e aprender, por imersão, uma grande quantidade de vocabulário da segunda língua.

5.3 Análise global

Foram analisados tanto os projetos do 2º ano quanto os do 5º ano, seguindo os mesmos critérios e procedimentos apresentados na seção Métodos em ambas as turmas. A sequência didática dos projetos, a evolução em complexidade, conteúdos e habilidades desenvolvidas podem ser consultados nas tabelas de sequência didática dos projetos.

Cabe aqui uma breve comparação da evolução dos níveis de complexidade dos objetivos de aprendizagem de acordo com a Taxonomia de Bloom revisada no domínio cognitivo do período letivo de cada turma. No 2º ano, é possível notar que, nos primeiros projetos, foram utilizados mais verbos dos níveis “recordar” e “compreender” (os primeiros níveis da hierarquia). Os demais, como o “aplicar”, aparecem de forma mais distribuída ao longo do ano. No 5º ano, os níveis de complexidade dos objetivos de aprendizagem tiveram ocorrências mais distribuídas dentro de um mesmo projeto e podem ser observados na sequência didática de projetos.

Na Tabela 1, observa-se que a quantidade de verbos do nível “recordar”, o primeiro da hierarquia do nível cognitivo da Taxonomia de Bloom, é predominante no 2º ano, o que faz todo sentido, pois os estudantes estão sendo iniciados na programação. Já no 5º ano, o que predomina são os verbos do nível cognitivo “aplicar”, considerando o fato de que os estudantes estão em um nível maior de desenvolvimento de programação, devido à experiência com as aulas de *Scratch* dos anos anteriores. Em relação aos demais níveis, eles foram aplicados de forma distribuída em ambas as turmas.

Tabela 1: Quantidade de verbos utilizados por nível de complexidade da Taxonomia de Bloom no domínio cognitivo

Turma	Recordar	Compreender	Aplicar	Analisar	Avaliar	Criar
2º ano	19	6	11	3	5	7
5º ano	5	7	26	6	2	5

Com relação às habilidades do PC praticadas em cada turma (Tabela 2), algoritmos e reconhecimento de padrões foram as competências mais desenvolvidas no 2º ano. No 5º ano, houve um equilíbrio das quatro habilidades, com um quantitativo um pouco menor, mas não muito distante, do reconhecimento de padrões.

Tabela 2: Habilidades do PC trabalhadas em cada turma

Turma	Algoritmos	Abstração	Decomposição	Rec. de Padrões
2º ano	8	3	3	9
5º ano	7	7	7	5

5.4 Limitações

Esta pesquisa teve início nos últimos meses de 2019 (encerramento do ano letivo). Por isto, não houve tempo para aplicação de avaliação com os estudantes daquele período. E, no ano seguinte, em 2020, iniciou-se a pandemia do novo Coronavírus. Com isto, as escolas tiveram que ser fechadas e adaptadas para o ensino remoto. Desta forma, não foi possível executar uma avaliação quantitativa da sequência didática proposta. Então, os autores deste trabalho optaram por fazer um resgate dos planos de aulas de 2018, que já estavam bem consolidados.

6 CONSIDERAÇÕES FINAIS

O trabalho faz a descrição de uma sequência didática, com um breve relato de experiência, de aulas de programação em inglês para

crianças, que envolve a prática simultânea de ambas habilidades durante as aulas, a programação como conteúdo principal e o inglês como meio para obtenção do conhecimento.

O uso da Taxonomia revisada de Bloom foi muito útil, pois, além de analisar se a sequência de objetivos de aprendizagem seguia a ordem hierárquica do domínio cognitivo, ajudou a reformular os objetivos de aprendizagem resgatados nos registros documentais e elaborar novos. Com isto, pôde-se compreender melhor pontos importantes que já vinham sendo feitos cotidianamente, mas sem uma classificação sistemática. O mesmo se aplica para os demais itens da sequência didática dos projetos, como habilidades do PC e conceitos e práticas computacionais, por exemplo. Diversas habilidades do PC já vinham sendo praticadas de forma implícita. Depois da análise, isso ficou mais evidente e destacou a relevância do trabalho realizado. Consequentemente, a perspectiva é que novos projetos sejam melhor elaborados tendo maior clareza da hierarquia a ser seguida e do conjunto de habilidades a ser desenvolvido.

Apesar de não haver uma avaliação de aprendizado dos estudantes, foi possível perceber a evolução deles, à medida que o nível de complexidade dos projetos ia aumentando. Além disso, professores e coordenação fizeram alguns relatos sobre as associações feitas pelas crianças nas demais aulas em relação ao que foi ensinado nas aulas de programação. Entretanto, não foi possível mensurar esse *feedback* dado o tipo da pesquisa. Os alunos se mantiveram motivados em boa parte das aulas (perspectiva do professor/pesquisador), mesmo utilizando somente o *Scratch* como ferramenta, pois, apesar de a plataforma ser a mesma, os projetos eram sempre com uma temática e uma complexidade diferentes.

Para trabalhos futuros, propõe-se analisar quali-quantitativamente o desenvolvimento dos estudantes em relação às habilidades do PC, como a segunda língua interfere no processo de aprendizagem e como eles se mantiveram motivados ao longo dos projetos.

REFERÊNCIAS

- [1] David Barr, John Harrison, and Leslie Conery. 2011. Computational thinking: A digital age skill for everyone. *Learning & Leading with Technology* 38, 6, 20–23.
- [2] Valerie Barr and Chris Stephenson. 2011. Bringing computational thinking to K-12: what is Involved and what is the role of the computer science education community? *Acm Inroads* 2, 1, 48–54.
- [3] B. S. Bloom, M. B. Engelhart, E. J. Furst, W. H. Hill, and D. R. Krathwohl. 1956. *Taxonomy of educational objectives. The classification of educational goals. Handbook 1: Cognitive domain*. Longmans Green, New York.
- [4] Christian P. Brackmann. 2017. *Desenvolvimento do Pensamento Computacional através de atividades desplugadas na educação básica*. Ph.D. Dissertation. Universidade Federal do Rio Grande do Sul.
- [5] Brasil. 2018. Base Nacional Comum Curricular: Educação é a Base. http://basenacionalcomum.mec.gov.br/images/BNCC_EI_EF_110518_-versaofinal_site.pdf http://basenacionalcomum.mec.gov.br/images/BNCC_EI_EF_110518_-versaofinal_site.pdf, [acessado em Jul 21].
- [6] Karen Brennan and Mitchel Resnick. 2012. New frameworks for studying and assessing the development of computational thinking. In *Proceedings of the 2012 annual meeting of the American educational research association, Vancouver, Canada*, Vol. 1, 25.
- [7] British Council. 2014. Learning English in Brazil Understanding the aims and expectations of the Brazilian emerging middle classes. https://www.britishcouncil.org.br/sites/default/files/learning_english_in_brazil.pdf https://www.britishcouncil.org.br/sites/default/files/learning_english_in_brazil.pdf, [acessado em Jul 21].
- [8] Jorge Cativo. 2021. A Taxonomia de Bloom, verbos e os processos cognitivos. *Biblioteconomia Digital*. <http://biblioteconomia digital.com.br/2012/08/a-taxonomia-de-bloom-verbos-e-os.html>
- [9] Sandra Costa, Anabela Gomes, and Teresa Pessoa. 2016. Using scratch to teach and learn English as a foreign language in elementary school. *International Journal of Education and Learning Systems* 1.
- [10] Andrew Csizmadia, Paul Curzon, Mark Dorling, Simon Humphreys, Thomas Ng, Cynthia Selby, and John Woollard. 2015. *Computational thinking - a guide for teachers*. Project Report. University of Southampton. <https://eprints.soton.ac.uk/424545/>
- [11] Ana Liz Souto Oliveira de Araújo, Pasqueline Dantas Scaico, Luiz Fernando de Paiva, Hévellyn de Moraes Rabêlo, Luan de Luna Santos, Francisco Italo Ribeiro Pessoa, Jonas Mendonça Targino, and Leonardo dos Santos Costa. 2013. Aplicação da Taxonomia de Bloom no ensino de programação com Scratch. In *Anais do Workshop de Informática na Escola*, Vol. 19, 31.
- [12] Rozelma de França and Haroldo do Amaral. 2013. Proposta Metodológica de Ensino e Avaliação para o Desenvolvimento do Pensamento Computacional com o Uso do Scratch. *Anais do Workshop de Informática na Escola* 19, 1, 179–188.
- [13] Jaqueline do Nascimento Brito, Higor Pereira de Brito, Hélio Junho de Souza Santos, and Flávia Luciana GMP de Araújo. [n.d.]. TECNOLOGIA E EDUCAÇÃO: O USO DA PROGRAMAÇÃO EM BLOCOS PARA PRODUÇÃO DO GLOSSÁRIO BILÍNGUE (PORTUGUÊS/LIBRAS).
- [14] João José Saraiva Fonseca. 2002. *Apostila de metodologia da pesquisa científica*. UFRGS Editora.
- [15] Débora Garofalo. 2018. Como as metodologias ativas favorecem o aprendizado. *Nova escola*.
- [16] Fred Genesee. 1987. *Learning through two languages: Studies of immersion and bilingual education*. Vol. 163. Newbury House Cambridge, MA.
- [17] Antonio Carlos Gil. 2002. *Como elaborar projetos de pesquisa*. Vol. 4. Atlas São Paulo.
- [18] Rafael Granada, Victor Cesário, Darcylene Domingues, Regina Barwaldt, Ricardo Rodrigues, and Cristiane Fernandes. 2017. Dicionário de termos de computação como facilitador no ensino de programação para surdos. In *Anais dos Workshops do Congresso Brasileiro de Informática na Educação*, Vol. 6, 1049.
- [19] David R Krathwohl. 2002. A revision of Bloom's taxonomy: An overview. *Theory into practice* 41, 4, 212–218.
- [20] Patsy M Lightbown and Nina Spada. 2013. *How languages are learned 4th edition-Oxford Handbooks for Language Teachers*. Oxford university press.
- [21] Roy Lyster. 2011. Content-based second language teaching. *Handbook of research in second language teaching and learning* 2, 611–630.
- [22] J Moreno-León and Gregorio Robles. 2015. Computer programming as an educational tool in the English classroom a preliminary study. In *2015 IEEE global engineering education conference (EDUCON)*. IEEE, 961–966.
- [23] Mitchel Resnick and Natalie Rusk. 2020. Coding at a crossroads. *Commun. ACM* 63, 11, 120–127.
- [24] Vinicius Santana and Rachel Ferreira. 2017. Desenvolvimento de games na Língua Inglesa: lógica e criatividade na construção do conhecimento. In *Anais dos Workshops do Congresso Brasileiro de Informática na Educação*, Vol. 6, 414.
- [25] Antonio Sarasa-Cabezuelo. 2019. Use of Scratch for the Teaching of Second Languages. *International Journal of Emerging Technologies in Learning (iJET)* 14, 21, 80–95.
- [26] Miles Turnbull, Sharon Lapkin, and Doug Hart. 2001. Grade 3 immersion students' performance in literacy and mathematics: Province-wide results from Ontario (1998-99). *Canadian modern language review* 58, 1, 9–26.
- [27] Faculdade Unina. 2021. O que É taxonomia de bloom e como ela É aplicada na educação? <https://unina.edu.br/o-que-e-taxonomia-de-bloom-e-como-ela-e-aplicada-na-educacao/>
- [28] Ashok Kumar Veerasamy and Anna Shillabeer. 2014. Teaching English based programming courses to English language learners/non-native speakers of English. *International Proceedings of Economics Development and Research* 70, 17.
- [29] Jeannette M. Wing. 2006. Computational thinking. *Commun. ACM* 49, 3, 33–35.
- [30] Jeannette M. Wing. 2014. Computational Thinking Benefits Society. *Social Issues in Computing - Academic Press 40th Anniversary Blog*.
- [31] Project Zero. [n.d.]. See, Think, Wonder - A routine for exploring works of art and other interesting things. https://pz.harvard.edu/sites/default/files/SeeThinkWonder_2.pdf