

**PROPOSTA DE SEQUÊNCIA DIDÁTICA PARA
INTRODUÇÃO DA COMPUTAÇÃO E DO
PENSAMENTO COMPUTACIONAL NO ENSINO
FUNDAMENTAL II BASEADA NAS METÁFORAS
DE PERSPECTIVAS CULTURAIS**

OTO BRAZ ASSUNÇÃO

**PROPOSTA DE SEQUÊNCIA DIDÁTICA PARA
INTRODUÇÃO DA COMPUTAÇÃO E DO
PENSAMENTO COMPUTACIONAL NO ENSINO
FUNDAMENTAL II BASEADA NAS METÁFORAS
DE PERSPECTIVAS CULTURAIS**

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação do Instituto de Ciências Exatas da Universidade Federal de Minas Gerais como requisito parcial para a obtenção do grau de Mestre em Ciência da Computação.

ORIENTADOR: RAQUEL OLIVEIRA PRATES
COORIENTADOR: ELAINE SOARES FRANÇA

Belo Horizonte, MG

Maio de 2021

© 2021, Oto Braz Assunção.
Todos os direitos reservados.

Braz Assunção, Oto

Proposta de Sequência Didática para Introdução da
Computação e do Pensamento Computacional no Ensino
Fundamental II baseada nas Metáforas de Perspectivas
Culturais / Oto Braz Assunção. — Belo Horizonte, MG, 2021
xxi, 210 f. : il. ; 29cm

Dissertação (mestrado) — Universidade Federal de Minas
Gerais

Orientador: Raquel Oliveira Prates

1. Pensamento Computacional. 2. Computação na Educação
Básica. 3. Metáforas de Perspectivas Culturais. 4. Interação
Humano-Computador. I. Título.

[Folha de Aprovação]

Quando a secretaria do Curso fornecer esta folha,
ela deve ser digitalizada e armazenada no disco em formato gráfico.

Se você estiver usando o `pdflatex`,
armazene o arquivo preferencialmente em formato PNG
(o formato JPEG é pior neste caso).

Se você estiver usando o `latex` (não o `pdflatex`),
terá que converter o arquivo gráfico para o formato EPS.

Em seguida, acrescente a opção `approval={nome do arquivo}`
ao comando `\ppgccufmg`.

Se a imagem da folha de aprovação precisar ser ajustada, use:
`approval=[ajuste] [escala] {nome do arquivo}`
onde *ajuste* é uma distância para deslocar a imagem para baixo
e *escala* é um fator de escala para a imagem. Por exemplo:
`approval=[-2cm] [0.9] {nome do arquivo}`
desloca a imagem 2cm para cima e a escala em 90%.

Agradecimentos

Primeiramente, sou grato aos meus pais por todo o apoio e suporte e por sempre me encorajarem durante toda a minha vida.

Agradeço à minha orientadora, Raquel Oliveira Prates, por todo o incentivo e paciência ao me guiar e aconselhar nos inúmeros períodos de incerteza enfrentados por mim durante o mestrado. Igualmente agradeço à Elaine Soares França, que aceitou coorientar este trabalho e facilitou imensamente a sua condução. Este trabalho não poderia ter sido realizado sem o apoio e empenho de vocês. Muito obrigado!

Agradeço aos meus colegas do laboratório PENSi. À Gabriela, colega que trabalhou junto comigo durante a etapa preliminar do meu trabalho na classificação dos recursos didáticos. Ao Francisco e a Tati por me ajudarem no primeiro semestre do mestrado. Obrigado Ana Paula pelos inúmeros conselhos e dicas que facilitaram a minha vida acadêmica e também pelas caronas. À Débora pelos diversos momentos de humor e descontração no laboratório. A todos do PENSi, Amanda, Caio, Érica, Fabrício, Joana, Paulo e Ulisses, obrigado pelas conversas e pelo tempo que passamos juntos dentro e fora do laboratório.

Sou grato também a todos os meus colegas da UFMG, especialmente ao Lucas Saraiva, amigo que esteve presente e com quem pude contar inclusive antes do início do mestrado, e ao Celso, com quem estudei junto durante muitas horas para PAA.

Ao Centro Pedagógico da UFMG por permitirem que conduzíssemos a nossa pesquisa e ministrássemos três GTDs de Pensamento Computacional na escola.

Aos 27 alunos que participaram dos GTDs ministrados e aos seus responsáveis por permitirem que eles participassem da pesquisa.

À Nathalia Campos por me ajudar na condução dos GTDs e atividades no Centro Pedagógico.

Ao CNPq pelo suporte financeiro que recebi através da bolsa de estudos: 132134/2018-7.

Resumo

A contínua inserção da tecnologia digital nas diversas áreas de atuação da sociedade faz com que as competências do Pensamento Computacional (PC) se tornem cada vez mais necessárias nos dias de hoje. É argumentado em todo o mundo que a Computação deve ser abordada de alguma forma na educação básica a fim de preparar jovens que atuarão em um mundo dinâmico e que deverão lidar com diversos tipos de problemas e situações, inclusive aqueles que não existem atualmente. No Brasil também há o reconhecimento da importância destas habilidades não apenas por pesquisadores da área e pela Sociedade Brasileira de Computação, mas também pelo Ministério da Educação que passou a abordar tanto a Computação quanto o PC na nova Base Nacional Comum Curricular. Dado este reconhecimento, este trabalho tem como objetivo a concepção de uma sequência didática para apoiar professores no ensino da Computação para alunos do Ensino Fundamental II e fomentação do desenvolvimento das habilidades de PC nestes alunos através do aprendizado e prática da lógica de programação. Assim, ao tratarmos o Pensamento Computacional como uma cultura, classificamos recursos didáticos para o ensino do PC no que diz respeito a suas complexidades à luz das Metáforas de Perspectivas Culturais. Utilizamos esta classificação para selecionarmos alguns dos recursos para serem aplicados na prática em salas de aula. Conduzimos então uma Pesquisa-Ação com alunos do Ensino Fundamental II estruturada em três estudos para avaliarmos os recursos didáticos selecionados. No Estudo I, conduzimos um curso de PC no semestre 2019/1 com 16 alunos. No Estudo II, este curso foi revisado e ministrado novamente no semestre 2019/2 com outros 12 participantes. O Estudo III foi uma continuação com 8 alunos do Estudo I interessados em continuar aprendendo sobre a Computação e PC no semestre 2019/2. Com base no que aprendemos com os três estudos, elaboramos uma sequência didática composta por um total de 12 aulas para professores interessados em ensinar o PC para alunos do Ensino Fundamental II.

Palavras-chave: Pensamento Computacional, Computação na Educação Básica, Metáforas de Perspectivas Culturais, Interação Humano-Computador.

Abstract

Computational Thinking (CT) skills have become more desirable nowadays considering digital technologies are becoming more prevalent in different fields of study and work performed by society. It is argued globally that Computer Science should be addressed in K-12 curricula in order to prepare students who will be working in a dynamic world and dealing with different types of problems and issues, including those which have yet to exist. In Brazil, the importance of CT skills is acknowledged by researchers, the Brazilian Computer Society, and also by the Ministry of Education, which has addressed computer science and CT in the latest National Common Curriculum Base. Thus, this work has the ultimate goal of creating a didactic sequence to support educators in teaching Computer Science to middle school students and fomenting the development of their CT skills through the practice of coding. To that end, we considered CT a foreign culture unfamiliar to middle school students to classify CT teaching resources being used in Brazil in the light of the Cultural Viewpoint Metaphors. Then, we used the classification as a basis to select teaching resources to be applied in practice. Finally, we conducted an Action-Research, which was divided into three studies, with middle school students to analyze the resources we selected. In Study I, we taught an introductory CT course to 16 students in the first semester of 2019. In Study II, this course was improved, and then we taught this modified course to a new group of 12 students. Study III was a continuation with 8 participants of the first study that showed interest in learning more about computer science and CT. Finally, we applied everything we have learned in the three studies to create a didactic sequence comprised of 12 lessons for educators interested in introducing computer science and CT to middle school students.

Keywords: Computational Thinking, Computer Science in K-12, Cultural Viewpoint Metaphors, Human-Computer Interaction.

Lista de Figuras

1.1	Visão geral da pesquisa	5
2.1	Estrutura do CRTC. Fonte: [CIEB, 2017]	17
2.2	Exemplo de orientações para o 6º ano do EF: Fonte: [CIEB, 2017]	18
3.1	Metáforas de Perspectivas Culturais	27
4.1	Base para a sequência didática à luz das MPCs	32
4.2	GTDs conduzidos no ano de 2019	37
5.1	Contando os Pontos: exemplo de representação do número 9	45
5.2	Scratch Blocks	47
5.3	LPV do Lego Mindstorms. Fonte: https://www.funcodeforkids.com/lego-mindstorms-ev3-compere-block-in-detail/	47
5.4	Exemplo da LPV do Kodu Game Lab. Fonte: http://www.kodugamelab.com/	48
5.5	Ferramentas e atividades selecionadas	49
5.6	Desafios 1 e 2 do Estacionamento Algorítmico	50
5.7	Exemplos de tarefas a serem decompostas	51
5.8	Versão modificada do Programando em Papel utilizada na Pesquisa	51
5.9	LightBot: exemplo de desafio e sua solução	52
5.10	Interface do Scratch	54
6.1	Recursos do GTD de Pensamento Computacional I	58
6.2	Recursos do GTD de Pensamento Computacional II	59
6.3	Frequência de uso do computador durante a semana	60
6.4	Atividades mais feitas no computador	60
6.5	Avaliação das ferramentas e atividades	70
6.6	Dificuldade das ferramentas e atividades	75
6.7	Exemplo de pergunta utilizada no GTD de PC I 2019/1	79

6.8	Anime seu Nome: exemplos de scripts criados pelos alunos	83
6.9	Exemplos de scripts criados pelos alunos: Jogo do Pong e Jogo de Pular . .	83
6.10	Avaliação das ferramentas e atividades	92
6.11	Avaliação das ferramentas e atividades	97
6.12	Avaliação das ferramentas e atividades	114
6.13	Estudo III: Dificuldade das Aulas	118
A.1	Indicação visual se as lições foram completadas com o menor número de blocos ou não	163
A.2	Comandos que os alunos utilizarão para manipular as variáveis	182

Lista de Tabelas

5.1	Recursos didáticos associados ao Viajante Doméstico	42
5.2	Recursos didáticos associados ao Observador à Distância	44
5.3	Recursos didáticos associados ao Visitante em Tour Guiado	44
5.4	Recursos didáticos associados ao Viajante Doméstico	46
5.5	Recursos didáticos associados ao Estrangeiro sem Tradutor	48
5.6	Visão geral da sequência didática	55
6.1	Aulas conduzidas no GTD de PC I 2019/1	61
6.2	Aulas conduzidas no GTD de PC I 2019/2	86
6.3	Aulas conduzidas no GTD de PC II 2019/2	108
6.4	Visão geral das aulas do curso de PC	127

Sumário

Agradecimentos	ix
Resumo	xi
Abstract	xiii
Lista de Figuras	xv
Lista de Tabelas	xvii
1 Introdução	1
1.1 Motivação e Objetivos	3
1.2 Condução da Pesquisa	4
1.3 Resultados e Contribuições	5
1.4 Organização da Dissertação	7
2 A Computação e o Pensamento Computacional na Educação Básica Brasileira	9
2.1 Iniciativas em Salas de Aula	9
2.1.1 Iniciativas com a Computação Desplugada	9
2.1.2 Iniciativas com Ferramentas Digitais	11
2.2 Diretrizes Curriculares	14
2.2.1 Base Nacional Comum Curricular	14
2.2.2 Diretrizes para Ensino de Computação na Educação Básica . . .	15
2.2.3 Currículo de Referência em Tecnologia e Computação	16
2.3 Materiais de Apoio ao Ensino da Computação e Pensamento Computacional	18
2.3.1 Livros da Série Computação Fundamental	18
2.3.2 Programaê!	22

2.3.3	Sequências didáticas propostas por pesquisadores	23
3	Fundamentação Teórica	25
3.1	Engenharia Semiótica	25
3.2	As Metáforas de Perspectivas Culturais	26
3.3	O Pensamento Computacional como uma Cultura	29
4	Metodologia	31
4.1	Geração da Sequência Didática	31
4.1.1	Classificação de Recursos Didáticos à Luz das MPCs	32
4.1.2	Seleção de Recursos Didáticos Didáticos	34
4.2	Desenho da Pesquisa Ação	35
4.2.1	Caracterização da Pesquisa	35
4.2.2	Contexto da Pesquisa	36
4.2.3	Coleta de Dados	38
5	Geração de Sequência Didática para Introduzir a Computação e o Pensamento Computacional	41
5.1	Classificação de Recursos Didáticos Comumente Empregados no Brasil	41
5.1.1	Viajante Doméstico	42
5.1.2	Observador à Distância	43
5.1.3	Visitante em Tour Guiado	43
5.1.4	Estrangeiro com Tradutor	45
5.1.5	Estrangeiro sem Tradutor	48
5.2	Ferramentas e Materiais Seleccionadas para Aplicação em Salas de Aula	49
5.3	Proposta Inicial da Sequência Didática	55
6	A Pesquisa-Ação	57
6.1	GTDs de Pensamento Computacional	57
6.2	Participantes	59
6.3	Estudo I	61
6.3.1	Descrição das Aulas	61
6.3.2	Avaliação da Experiência	69
6.3.3	Discussão e Lições Aprendidas	80
6.4	Estudo II	85
6.4.1	Descrição das Aulas	85
6.4.2	Avaliação da Experiência	91
6.4.3	Discussão e Lições Aprendidas	102

6.5	Estudo III	107
6.5.1	Descrição das Aulas	108
6.5.2	Aulas que não puderam ser Conduzidas	113
6.5.3	Avaliação da Experiência	113
6.5.4	Discussões e Lições Aprendidas	120
6.6	Consolidação da Sequência Didática	125
7	Conclusão	133
7.1	Contribuições	134
7.2	Limitações	136
7.3	Trabalhos Futuros	136
	Referências Bibliográficas	139
	Apêndice A Roteiro das Aulas	151
	Apêndice B Questionário de Perfil	199
	Apêndice C Questionário de Avaliação	201
	Apêndice D Roteiro: Grupos Focais 2019/01	203
	Apêndice E Roteiro: Grupos Focais 2019/02	205
	Apêndice F Roteiro: Entrevista com alunos que não continuaram par- ticipando	207
	Apêndice G Roteiro: Entrevista com os alunos que continuaram par- ticipando	209

Capítulo 1

Introdução

O Pensamento Computacional (PC) ganhou a atenção do mundo em 2006 após Wing [2006] ter publicado o artigo *Computational Thinking*, onde ela argumenta sobre o quão importante e requisitada esta habilidade seria na sociedade contemporânea. Segundo Wing, ensinar e fomentar as habilidades do PC na Educação Básica se tornaria tão essencial quanto a leitura, escrita e a Matemática. [Conforme apontado por ela](#), ser capaz de pensar, abstrair problemas e solucioná-los como cientista da Computação é uma habilidade que todos devem buscar desenvolver e aprimorar para que sejam capazes de atuar e se sucederem nesta sociedade altamente dependente da tecnologia.

Wing definiu o PC como “*processos cognitivos envolvidos na formulação de problemas e representação de suas soluções tal que um agente computacional consiga processá-las de forma eficiente*” [Wing, 2011]. Conforme indicado em [Lye & Koh, 2014], algumas organizações tais como a *International Society for Technology in Education* (ISTE) e *National Research Council* (NRC) consideram que o PC não é dependente da programação, podendo ser exibido sem que haja necessariamente a criação de ferramentas digitais. Já a Sociedade Brasileira de Computação afirma que o PC é “*a capacidade de sistematizar, representar, analisar e resolver problemas*” [SBC, 2017]. O PC é comumente decomposto em um conjunto de pilares, considerados como as suas principais competências. Barr & Stephenson [2011] consideram que elas são: *Coleta de Dados, Análise de Dados, Representação de Dados, Decomposição de Problemas, Abstração, Algoritmos e Procedimentos, Automação, Paralelização e Simulação*. Já a SBC definiu três competências mais gerais: *abstração, automação e análise*.

Embora ainda existam diversas discussões e argumentações a respeito das definições de PC e suas competências, professores, pesquisadores, instituições de ensino e governos têm concordado com o que foi escrito por Wing. Hoje, 15 anos depois, observamos que a visão de Wing [2006] no que diz respeito ao PC vem se concretizando,

uma vez que a Computação e tecnologias digitais ainda continuam se inserindo mais e mais na vida das pessoas, inclusive na vida daquelas que não trabalham com a Computação diretamente. A inserção da Computação na Educação Básica deve ser algo a ser considerado por governos e instituições de ensino a fim de preparar a população para solucionar e lidar com problemas complexos existentes enfrentados pela sociedade e também com problemas que ainda surgirão. Vários países como os Estados Unidos, Alemanha, Austrália, Coreia do Sul, Inglaterra e Japão já integraram ou têm feito o processo de integração do ensino da Computação na Educação Básica a fim de educar as crianças no que diz respeito às tecnologias digitais e sua utilização de forma ética e segura, além de promover o desenvolvimento das habilidades do PC para a resolução de problemas [Heintz et al., 2016; SBC, 2017]. Conforme indicado no trabalho de Heintz et al. [2016], alguns países criaram disciplinas próprias para o ensino de PC e as incluíram em seus respectivos currículos da Educação Básica; em outros, o PC foi incorporado às disciplinas regulares já existentes (e.g.: Matemática, Biologia, Física, Química) de tal forma que a Computação seja ensinada e introduzida ao mesmo tempo que os tópicos tradicionais. Desta forma, os alunos podem aprender sobre a Computação e utilizá-la para resolver problemas e abordar assuntos das disciplinas regulares, consequentemente, levando ao desenvolvimento do PC.

No Brasil, embora a Computação ainda não tenha sido integrada à Educação Básica, a importância que a Computação e o PC têm e terão para a população já é reconhecida pela Sociedade Brasileira de Computação (SBC) e pelo Ministério da Educação (MEC). Em 2017, a SBC publicou um documento intitulado *Referenciais de formação em Computação: Educação Básica* [SBC, 2017] com o objetivo de promover a necessidade da inclusão da Computação na Educação Básica brasileira:

“Não é mais possível imaginar uma sociedade na qual os indivíduos não necessitem conhecimentos básicos de Computação, tão importantes para a vida na sociedade contemporânea quanto os conhecimentos básicos de Matemática, Filosofia, Física, ou outras ciências.” [SBC, 2017, pág. 1].

Além de argumentar sobre a importância da área da Computação para a sociedade brasileira, a SBC detalha e apresenta a área da Computação e indica habilidades esperadas a serem desenvolvidas em cada uma das etapas da Educação Básica¹. O PC também passou a ser abordado e mencionado na nova Base Nacional Comum Curricular do Brasil [MEC, 2020] como um dos eixos da Computação [no contexto da Educação](#)

¹Educação Infantil (0 a 5 anos), Ensino Fundamental (Anos Iniciais ou Ensino Fundamental I: 6 a 10 anos e Anos Finais ou Ensino Fundamental II: 11 a 14 anos) e Ensino Médio (15 a 18 anos)

[Básica](#), juntamente com os eixos da Cultura Digital e Mundo Digital. Além de ser mencionada a importância do PC, algoritmos e fluxogramas para os alunos, a matemática e os seus processos de aprendizagem são apresentados como uma das formas de desenvolver o PC na etapa do Ensino Fundamental. E na etapa do Ensino Médio é discutida a importância da Computação para os jovens que se inserirão em uma sociedade que constantemente utiliza e utilizará a tecnologia digital em seus afazeres.

1.1 Motivação e Objetivos

Apesar do reconhecimento da importância da Computação na Educação Básica, o Brasil está muito atrasado se comparado aos países mencionados previamente. O ensino da Computação e PC na Educação Básica brasileira se dá primariamente através de iniciativas conduzidas em pesquisas e estudos impulsionados pelo crescimento do interesse pela introdução da Computação na Educação Básica nos últimos anos [Santos et al., 2018; Oliveira et al., 2019]. Conforme visto por Santos et al. [2018], o foco no Brasil ainda está em envolver estudantes com a Computação e lhes apresentar os benefícios que o aprendizado da Computação pode lhes trazer. Assim, tais estudantes teriam menos resistência quando a Computação fosse inserida oficialmente na Educação Básica brasileira.

A indisponibilidade de profissionais adequados para ministrar disciplinas da Computação em escolas públicas será um problema que o Brasil enfrentará caso a Computação passasse a ser uma disciplina regular e obrigatória na Educação Básica [Santos et al., 2018]. Santos et al. [2018] também menciona que professores que estariam ensinando a Computação não seriam qualificados, pois eles teriam pouco ou nenhum conhecimento sobre o PC. [Financeiramente, instituições particulares não teriam grandes problemas em contratar licenciados em Computação para ministrar tais aulas; contudo, a realidade não seria a mesma para escolas públicas.](#)

Neste contexto, com este trabalho buscamos propor uma sequência didática para professores, inclusive aqueles pouco familiarizados com a área, introduzirem a Computação e o PC a alunos do EF II que não tiveram a oportunidade de aprender sobre a Computação nos anos iniciais do Ensino Fundamental. Nossa ideia é propor um conjunto de aulas, recomendações e disponibilizar materiais para apoiar os professores na condução das aulas. Nos fundamentamos nas Metáforas de Perspectivas Culturais (MPCs) [Salgado, 2011a], uma ferramenta epistêmica da Engenharia Semiótica que pode ser utilizada para auxiliar no processo de decisão de como uma nova cultura pode ser apresentada aos usuários em diferentes níveis de intensidade. Utilizamos uma

classificação de recursos didáticos à luz das MPCs no que diz respeito a como eles introduzem a Computação e o PC [Oliveira et al., 2019] para selecionarmos ferramentas (i.e.: plataformas online, programas de computador, jogos digitais) e atividades desplugadas a serem utilizadas no ensino de tal forma que a cultura da Computação e do PC sejam introduzidas e apropriadas pelos alunos de maneira gradual. Assim, os objetivos específicos do trabalho são os seguintes:

1. Classificar recursos didáticos para o ensino da Computação à luz das Metáforas de Perspectivas Culturais.
2. Selecionar recursos didáticos para o ensino da Computação com base nesta classificação de tal forma que a Computação seja introduzida de maneira gradual.
3. Gerar proposta de sequência didática inicial utilizando os recursos didáticos selecionados.
4. Realizar uma pesquisa-ação junto ao Centro Pedagógico da UFMG para aplicar a sequência didática em salas de aula.
5. [Avaliar a utilização da sequência didática e aperfeiçoar as aulas e atividades da sequência.](#)
6. Analisar as experiências em salas de aula tanto em termos da adoção das MPCs como guia, quanto em termos dos materiais adotados e aspectos práticos da aplicação com a turma.
7. Gerar versão final da sequência didática para professores ensinarem a Computação e o PC no EF II.

1.2 Condução da Pesquisa

Dado os objetivos descritos na seção anterior, o nosso trabalho foi dividido em duas partes conforme ilustrado na Figura 1.1: geração da sequência didática e condução da pesquisa-ação.

Na primeira parte, nós geramos uma sequência didática para ser aplicada na prática e avaliada. Para isso, nós, primeiramente, classificamos recursos didáticos (e.g.: ferramentas digitais, atividades desplugadas, livros, linguagens de programação, etc.) sendo adotados para o ensino da Computação e PC no Brasil. As MPCs foram utilizadas para os classificarmos quanto à complexidade e conhecimentos necessários para que alunos do EF II sem conhecimentos sobre a Computação consigam utilizá-las. Em

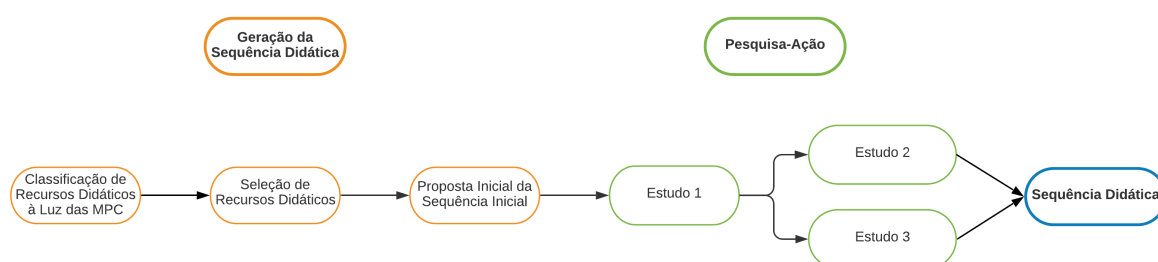


Figura 1.1: Visão geral da pesquisa

seguida, dentre os recursos classificados, selecionamos alguns para serem incluídos na sequência didática. Por fim, nos fundamentamos nas MPCs para propor a versão inicial da sequência didática utilizando os recursos selecionados.

Na segunda parte do trabalho, conduzimos uma Pesquisa-Ação estruturada em três estudos para avaliarmos e aprimorarmos a sequência didática. No primeiro estudo ministramos para 16 participantes o Grupo de Trabalho Diferenciado (GTD)² de PC I 2019/1, curso de caráter introdutório abordando conteúdos básicos da lógica de programação. Utilizamos os resultados deste estudo para melhorarmos o GTD de PC I e, assim, no segundo semestre de 2019 realizamos o segundo estudo ao ministrarmos o GTD de PC I 2019/2 para um novo grupo de alunos (12 participantes). Os mesmos recursos didáticos foram utilizados no GTD de PC I 2019/2; contudo, as atividades e aulas conduzidas foram revisadas. Por fim, o terceiro estudo, o GTD de PC II 2019/2, foi feito com 8 dos 16 alunos que participaram no primeiro semestre. Este novo GTD foi uma continuação do GTD de PC I 2019/1, pois os alunos desse GTD demonstraram interesse em continuar participando e aprender mais sobre a Computação. Finalmente, elaboramos a sequência didática final para a introdução da Computação e desenvolvimento do PC para o EF II a partir dos resultados dos três estudos e lições aprendidas com eles.

1.3 Resultados e Contribuições

Nosso trabalho gerou resultados e contribuições tanto teóricas, quanto práticas. As contribuições científicas foram a adoção das MPCs como fundamento para: (i) classificar recursos didáticos utilizados no ensino da Computação e PC e (ii) introduzir a Computação e o PC por meio de uma sequência didática gradativa. Já as contribuições práticas são a descrição dos estudos conduzidos com os alunos do EF II, o ensino da

²Grupo de Trabalho Diferenciado. Obs.: será explicado na Seção 4.2.2

Computação e PC a estes alunos e, por fim, a definição da sequência didática para o ensino da Computação e PC no Ensino Fundamental.

No que diz respeito aos resultados e contribuições científicas, as MPCs puderam ser utilizadas no contexto do ensino da Computação e PC. Todos os recursos didáticos foram associados a pelo menos uma metáfora, sendo este um indício que as MPCs, além de serem utilizadas como ferramentas epistêmicas na área de IHC, podem também ser utilizadas para classificar recursos didáticos. Esta classificação pode, inclusive, ser utilizada na prática para apoiar educadores buscando por recursos didáticos a serem utilizados no ensino da Computação e PC. Além disso, a abordagem de selecionar ferramentas e a ordem que elas seriam empregadas com base na classificação das ferramentas de ensino à luz das MPCs de tal forma que a Computação e o PC sejam introduzidas de maneira gradual aos estudantes demonstrou ser promissora. Os alunos não tiveram grandes dificuldades em compreender e utilizar as ferramentas na ordem em que elas foram introduzidas nos cursos; contudo, eles tiveram dificuldades em realizar algumas atividades nas ferramentas, pois as atividades em si tiveram complexidade muito alta.

Os resultados e contribuições práticas do trabalho vieram da condução da pesquisa-ação. Três estudos com os alunos do 6º ano do EF II para o desenvolvimento das habilidades do PC através da programação foram conduzidos e detalhados neste trabalho. Descrevemos todas as aulas ministradas, compartilhamos os materiais utilizados e discutimos o que aprendemos com a condução de cada estudo. Este detalhamento permite que os cursos ministrados possam ser replicados por outras pessoas. Além disso, as atividades e aulas conduzidas podem também ser adaptadas e melhoradas com base na descrição das abordagens adotadas que deram errado. Os dados coletados no decorrer dos estudos indicaram que os alunos ficaram satisfeitos com as aulas, uma vez que eles gostaram de aprender sobre a Computação e de terem participado do curso embora algumas atividades e ferramentas tenham sido chatas ou difíceis para alguns deles. Finalmente, a partir dos resultados dos estudos, criamos uma sequência didática para professores que pretendem introduzir a Computação para alunos do EF II e promover o desenvolvimento de suas habilidades do PC. Nesta sequência, incluímos a descrição completa das aulas e disponibilizamos materiais e atividades, indicando como as aulas e as atividades podem ser conduzidas pelos professores. Um total de 15 aulas foram incluídas e espera-se que cada uma delas dure em média 1h e 20min.

1.4 Organização da Dissertação

O restante do trabalho foi estruturado da seguinte forma. O Capítulo 2 apresenta o cenário Brasileiro no que diz respeito ao ensino da Computação na Educação Básica. Nós discutimos iniciativas conduzidas em salas de aula, diretrizes e referenciais curriculares e, por fim, materiais de apoio ao ensino do PC. No Capítulo 3 introduzimos a teoria da Engenharia Semiótica e as Metáforas de Perspectivas Culturais, que foram usadas como base para o planejamento de uma introdução gradual à Computação e ao PC. A metodologia adotada no trabalho é detalhada no Capítulo 4. Os resultados das duas primeiras etapas do trabalho são apresentados no 5. Primeiramente, apresentamos a classificação dos recursos didáticos à luz das MPCs e, em seguida, os recursos didáticos que selecionamos para serem aplicados em salas de aula. O Capítulo 6 descreve os três estudos em salas de aula, onde aplicamos os recursos selecionados, e os seus resultados. Finalmente, apresentamos as conclusões do trabalho no Capítulo 7.

Capítulo 2

A Computação e o Pensamento Computacional na Educação Básica Brasileira

Neste capítulo discutimos o cenário da Computação e Pensamento Computacional (PC) na educação brasileira. Apresentamos estudos e iniciativas conduzidas em salas de aula com alunos e as diferentes abordagens de ensino no que diz respeito aos recursos didáticos adotados. Em seguida, apresentamos diretrizes curriculares desenvolvidas no Brasil e, por fim, materiais de apoio ao ensino da Computação e fomentação do PC.

2.1 Iniciativas em Salas de Aula

Pesquisas realizadas no Brasil sobre o ensino de PC e Computação na Educação Básica são diversas, abordando diferentes tipos de recursos didáticos e trabalhando com diferentes perfis de participantes [Santos et al., 2018]. Elas vêm sendo conduzidas desde o ano de 2001 e a partir de 2010 houve um grande aumento no interesse de pesquisadores e professores pela área. Nesta seção, apresentamos e discutimos iniciativas realizadas em salas de aula para o ensino da Computação e desenvolvimento do PC no Brasil.

2.1.1 Iniciativas com a Computação Desplugada

A infraestrutura das instituições de ensino público da Educação Básica é bastante precária no Brasil. Muitas escolas não possuem laboratórios de informática ou os possuem, porém sem ou com poucos computadores em boas condições de uso [INEP, 2017; Todos pela Educação, 2017]. Segundo o Resumo Técnico do Censo da Educação

reestruturei
a subse-
ção de
acordo
com a
sugestão
da
Mirella

Básica, conduzido pelo Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira (INEP) em 2019, o Brasil possui mais de 180 mil escolas de Educação Básica [INEP, 2017]. Dentre elas, aproximadamente 60% pertence à rede municipal, 22.9% à rede privada, 16.7% à rede estadual e 0.4% à rede federal. No entanto, a rede municipal é a que mais carece de recursos tecnológicos. Computadores de mesa e portáteis estão disponíveis para os alunos utilizarem em apenas 34.3% e 20.4% das escolas da rede municipal. Além disso, a internet está disponível para o ensino e aprendizagem em apenas 29.6% delas.

Além da carência dos recursos tecnológicos nas instituições de ensino, há o problema do desuso da tecnologia pelos professores de instituições que possuem tais recursos [Batista & Damasceno, 2019; Todos pela Educação, 2017]. No estudo conduzido por Batista & Damasceno [2019] foi identificado que além da indisponibilidade de uma infraestrutura de qualidade e manutenção dos computadores, os professores não estão devidamente capacitados para utilizarem a tecnologia no ensino, não havendo apoio pedagógico para auxiliá-los no uso dos laboratórios de informática. No estudo conduzido em [Todos pela Educação, 2017], professores que participaram da pesquisa indicaram diversos motivos pelo desuso da tecnologia, dentre eles a falta de infraestrutura adequada nas escolas, a falta de materiais de ensino digitais e modelos pedagógicos para integração da tecnologia no ensino, carga horária insuficiente para a inserção de tecnologias digitais nas aulas, dificuldade em utilizar tecnologias digitais, entre outros. Não basta que apenas os recursos estejam disponíveis em condições de uso nas instituições, a capacitação dos professores na utilização destes recursos de forma adequada a fim de beneficiar o processo de ensino-aprendizagem nas escolas também é fundamental.

Levando em consideração a realidade brasileira, diversas iniciativas envolvendo a aplicação da Computação Desplugada¹ para o ensino da Computação vêm sendo conduzidas no Brasil [Rodrigues et al., 2018; Souza & Nunes, 2019], fazendo com que ela seja uma das metodologias mais empregadas por pesquisadores [Silva et al., 2018b]. Muitas destas iniciativas empregam atividades do livro *CS UNPLUGGED* [Bell et al., 2015], uma vez que suas atividades são bem estabelecidas e consolidadas [Oliveira et al., 2019; Souza & Nunes, 2019]. Bezerra [2014], Pereira et al. [2019] e Lopes & Ohashi [2019] relatam experiências do uso exclusivo de atividades desplugadas para desenvolver o Pensamento Computacional em alunos do Ensino Fundamental, sendo que todas ou maioria das atividades aplicadas nestes estudos foram retiradas ou adaptadas do livro *CS UNPLUGGED*. Há também pesquisadores que criam seus próprios materiais e atividades desplugadas para serem aplicadas nos estudos [Martin et al., 2018; Guarda

¹Ensino da Computação sem o intermédio de computadores ou outros dispositivos digitais.

et al., 2018; Casarotto et al., 2018; Guarda et al., 2019; Silva et al., 2019; Barroso et al., 2019].

A Computação Desplugada é uma metodologia especialmente importante no Brasil. Através da Computação Desplugada é possível que os alunos aprendam sobre a Computação independentemente da escola possuir ou não laboratórios de informática em condições de uso. Ademais, a aplicação das atividades desplugadas são menos complexas e utilizam materiais do dia-a-dia, favorecendo professores com menos confiança na utilização da tecnologia em sala de aula.

2.1.2 Iniciativas com Ferramentas Digitais

Além das iniciativas empregando a Computação Desplugada, pesquisadores também conduzem iniciativas utilizando abordagens que envolvem a utilização de recursos digitais. Dentre as abordagens com tecnologias digitais mais utilizadas no Brasil, estão as Linguagens de Programação Visual, Robótica Educacional e Jogos Digitais [Oliveira et al., 2019; Silva et al., 2018b].

Ambientes de Linguagem de Programação Visual

Iniciativas empregando ferramentas que fazem uso de Linguagens de Programação Visual (LPV) são comuns no Brasil [Oliveira et al., 2019]. Segundo Valente [2016], LPVs tornam o processo de programação mais simples porque a sintaxe é definida pelo encaixe dos blocos. Portanto, a utilização de LPVs se justifica especialmente para crianças que estão começando a ter contato com a lógica de programação já que elas não têm que lidar com problemas sintáticos quando utilizam ambientes de LPV, apenas problemas lógicos.

No Brasil, a ferramenta de linguagem visual mais utilizada é o Scratch, sendo utilizado tanto como ferramenta única, quanto em conjunto com outras ferramentas [Oliveira et al., 2019]. O Scratch [Resnick et al., 2009] é uma ferramenta gratuita e permite que alunos aprendam a programação e seus conceitos por meio de uma linguagem de programação em blocos. Como jogos simples podem ser desenvolvidos facilmente na ferramenta, [propõe-se](#) o desenvolvimento de jogos no Scratch como forma de engajar as crianças no aprendizado da Computação [Aono et al., 2017; Boucinha et al., 2017; Geraldles et al., 2019]. Há também estudos que exploram a interdisciplinaridade através do Scratch ao propor aos alunos que eles criem projetos relacionados a assuntos estudados na escola ou ao utilizar o Scratch para aprimorar o processo de ensino de determinados assuntos [Riboldi & Reichert, 2019; Rocha & Basso, 2019; Garcia et al., 2019; Almeida et al., 2019b].

A plataforma *Code.org* [Kalelioğlu, 2015], a qual possui várias lições que utilizam LPV em blocos, também é amplamente utilizada no Brasil para o ensino da Computação, normalmente em conjunto com outros recursos tais como a Computação Desplugada e o Scratch [Martins et al., 2016; Castro et al., 2016; Bathke & Raabe, 2016; de Oliveira et al., 2018; Geraldles et al., 2019]. O *AppInventor*, ferramenta para criação de aplicativos para smartphones por meio de uma LPV em blocos, também vêm sendo utilizado, porém com menor frequência se comparado ao Scratch e ao *Code.org* [Bauer et al., 2017; Godinho et al., 2017; Daniel et al., 2017].

Robótica

A Robótica Educacional (RE) é um termo utilizado para se referir à utilização da robótica como instrumento de ensino, geralmente por meio das atividades de construção e programação de robôs. Segundo Raabe & Gomes [2018], a RE é uma abordagem principalmente fundamentada no construcionismo, teoria epistemológica de Seymour Papert, que diz que o conhecimento é adquirido por meio da realização de ações que resultam em um produto concreto.

A RE possibilita a criação de um ambiente de ensino-aprendizado capaz de engajar e motivar alunos, promover a cooperação e o trabalho em equipe e estimular o desenvolvimento das habilidades do Pensamento Computacional [Santos et al., 2019; Almeida et al., 2019a]. Mediante às possibilidades e vantagens oferecidas pela RE, esta abordagem de ensino vem ganhando popularidade no Brasil e já é bastante empregada em iniciativas de ensino da Computação para alunos da Educação Básica [Santos et al., 2018; Oliveira et al., 2019]. Conforme os mapeamentos sistemáticos da literatura conduzidos por Santos et al. [2018] e Oliveira et al. [2019], as duas ferramentas mais utilizadas em iniciativas com a RE são os kits de robótica da LEGO e o Arduino.

Dentre os kits de robótica da LEGO, o LEGO® MINDSTORMS®² é utilizado por muitos pesquisadores brasileiros, sendo o kit da LEGO mais frequentemente utilizado em iniciativas no Brasil. Ele é indicado para alunos de 10 anos ou mais, fazendo com que este seja um kit adequado para o ensino de alunos de diferentes faixas etárias. Além de permitir que os alunos construam e programem seus próprios robôs, o kit também vem com apostilas para auxiliar ambos professores e alunos no desenvolvimento das atividades [Maffi, 2018]. Diferentes linguagens de programação podem ser utilizadas para programar os robôs. O kit possui sua própria linguagem de programação visual para simplificar a programação para os alunos, a qual é utilizada na maioria dos trabalhos com o LEGO® MINDSTORMS® [Friedrich et al., 2012; Silva et al.,

²LEGO® MINDSTORMS® 

2016; Reis et al., 2018; Maffi, 2018]. No entanto, os robôs da LEGO também podem ser programados com outras linguagens de programação como no trabalho de Almeida et al. [2017], que utilizou o linguagem Python para programá-los.

Além dos kits de robótica, o Arduino, placa com um microcontrolador de fácil programação e integração com dispositivos, vem sendo utilizado como alternativa, uma vez que é de baixo custo (diferentemente dos kits de robótica da Lego) [Silva, 2018; Victal & Cândido, 2019]. Como feito nos trabalhos de Silva [2018], Silva & Javaroni [2018] e Braga et al. [2018], a utilização do Scratch for Arduino (S4A) é uma abordagem comumente utilizada para tornar o processo do aprendizado por meio da robótica mais simples para alunos mais jovens. O S4A é uma extensão do Scratch, que possibilita que as placas Arduino sejam programadas por meio da linguagem de blocos do Scratch. Assim, os benefícios das LPV são combinados com os da robótica, facilitando e tornando lúdico o processo de programação da placa. No entanto, há também relatos de iniciativas onde a linguagem de programação textual C foi utilizada mesmo com participantes jovens [Silva et al., 2016; Costa et al., 2017; Victal & Cândido, 2019].

Jogos Digitais

Dois tipos de jogos digitais são comumente adotados nos estudos brasileiros: jogos de raciocínio lógico e jogos para o ensino de programação e Computação [Oliveira et al., 2019]. Ambos são geralmente utilizados nas fases iniciais dos estudos para desenvolver o raciocínio lógico antes de transicionar para a utilização de ambientes de programação mais complexos.

Jogos de lógica são uma forma de desenvolver capacidades do Pensamento Computacionais tais como a abstração e raciocínio lógico [Martins, 2016]. No Brasil, diversos destes jogos de lógica vêm sendo empregados em estudos envolvendo o ensino da Computação e do Pensamento Computacional. A maioria dos jogos relatados nestes estudos eram jogos disponibilizados no portal Racha Cuca (e.g.: Missionários e Canibais, O Lobo, a Ovelha e a Couve, Teste de Einstein, Pinguins numa Fria) [Martins, 2016; Boucinha, 2017; Marques et al., 2017; Silva et al., 2017]. Outros exemplos de jogos digitais de lógica são o BloxorZ, que foi utilizado no estudo conduzido por Raabe et al. [2017], e o Desafio com Palitos aplicado por Santana & Santos [2018].

Dentre os jogos que têm o objetivo de ensinar conceitos de programação, o LightBot é um dos mais empregados em estudos realizados no Brasil [Oliveira et al., 2019]. Em [Raabe et al., 2017], o LightBot foi uma das ferramentas digitais empregadas em aulas práticas de uma disciplina de Pensamento Computacional. No trabalho de Conforto et al. [2018], ele foi utilizado para introduzir a lógica de programação em uma

oficina com alunos dos anos finais do Ensino Fundamental e do Ensino Médio. O LightBot foi empregado em um curso de extensão realizado por Souza [2019] com alunos do 8º e 9º anos. Vale destacar que a versão do LightBot utilizada nos estudos é uma versão grátis disponibilizada como parte da iniciativa Hora do Código³. Além dele, há também relatos de pesquisadores que aplicaram o The Foos [Gomes & Alencar, 2015; Falcão et al., 2015], Minecraft [Silva et al., 2018a] e Swift Programming [Rino, 2018].

2.2 Diretrizes Curriculares


Em vista da contínua evolução e inovação tecnológica e a inserção de tecnologias digitais na vida social, acadêmica e profissional das pessoas, organizações vêm elaborando diretrizes para a inclusão da Computação na Educação Básica. Existem inúmeras diretrizes internacionais tais como aquelas propostas pelo CSTA (*Computer Science Teachers Association*)⁴ e o ISTE (*International Society for Technology in Education*)⁵; contudo, nosso foco nesta seção são diretrizes e referenciais nacionais. Nós apresentamos como a Computação e PC estão sendo abordados na Base Nacional Comum Curricular (BNCC) [MEC, 2020], as Diretrizes para Ensino de Computação na Educação Básica criadas pela SBC [2018] e, por fim, o Currículo de Referência em Tecnologia e Computação elaborado pelo CIEB [2017].


2.2.1 Base Nacional Comum Curricular

A Base Nacional Comum Curricular (BNCC) é um documento concebido por diversos especialistas de diversas áreas do conhecimento. O documento contém normas que instituem e definem o conjunto de competências, conhecimentos e habilidades que alunos da Educação Básica devem desenvolver e aprender ao longo de seus anos escolares. O Pensamento Computacional, a Computação e as Tecnologias Digitais de Informação e Comunicação (TDICs) passaram a ser abordados nas etapas do Ensino Fundamental e Ensino Médio da versão mais recente da BNCC [MEC, 2020].

Na etapa do Ensino Fundamental, o Pensamento Computacional é abordado como um conjunto de habilidades que potencialmente podem ser desenvolvidas por meio dos processos matemáticos de resolução de problemas, investigação, desenvolvimento de projetos e da modelagem [MEC, 2020]. Por exemplo, considera-se que as habilidades do PC possam ser uma adquiridas por meio de atividades de aprendizado de Álgebra

³Iniciativa para promover o ensino da Ciência da Computação através de atividades com 1 hora de duração: <https://hourofcode.com/br>

⁴Diretrizes do CSTA 

⁵Diretrizes do ISTE 

(unidade temática da BNCC) que envolvam a transformação de problemas descritos através de linguagem natural em outras linguagens (e.g.: fórmulas, tabelas e gráficos). Os algoritmos e fluxogramas também são tidos na BNCC como importantes conteúdos associados ao PC que podem ser estudados na disciplina da Matemática, sendo que a linguagem algorítmica é similar a linguagem algébrica e que a identificação de padrões é uma habilidade associada à álgebra e também ao PC.

A BNCC frisa que a Computação é especialmente importante na etapa do Ensino Médio, pois jovens estão em constante contato com a tecnologia digital, atuando tanto como consumidores, quanto como produtores. Além da importância que a Computação e as TDICs têm para a sociedade contemporânea, é destacada a necessidade de abordá-las na etapa do Ensino Médio a fim de preparar jovens que estão prestes a atuar em uma sociedade que está mudando constantemente. Assim, nesta etapa, o PC é apresentado como um dos eixos da Computação, juntamente com o Mundo Digital e a Cultura Digital e são definidas um conjunto de habilidades envolvendo a utilização das tecnologias digitais tais como: buscar dados e informações de forma crítica, apropriar-se de linguagens da cultura digital, utilizar ferramentas digitais para diversas tarefas (e.g.: compreender e produzir conteúdos, propor e implementar soluções, identificar e analisar problemas, etc.).

2.2.2 Diretrizes para Ensino de Computação na Educação Básica

A Sociedade Brasileira de Computação (SBC) idealiza a inserção da Computação na Educação Básica brasileira e, neste contexto, a SBC considera a Computação como uma ciência composta por três eixos: Pensamento Computacional, Mundo Digital, e Cultura Digital. O Pensamento Computacional envolve a abstração, análise e automação de problemas. O Mundo Digital engloba a codificação, processamento e distribuição de dados. Por fim, a Cultura Digital diz respeito à fluência digital, ética digital e Computação e sociedade.

Em 2017, a SBC publicou um documento intitulado “*Referenciais de Formação em Computação: Educação Básica*” [SBC, 2017], onde são abordados, explicados e clarificados diversos tópicos referentes à área da Computação, dentre eles os três eixos da Computação e a importância que ela tem e terá para a sociedade contemporânea. Além disso, também são indicadas quais habilidades da Computação devem ser desenvolvidas em cada uma das etapas.

Este documento foi então revisado, reformulado, e complementado em 2018. O documento revisado, “*Diretrizes para ensino de Computação na Educação Básica*”

[SBC, 2018], cobre todos os tópicos abordados na versão de 2017, porém com mais detalhes. Diferentemente da versão inicial, a etapa da Educação Infantil não é abordada nesta nova versão, que detalha habilidades mais específicas para o Ensino Fundamental I, Ensino Fundamental II e para o Ensino Médio. Ademais, foram adicionadas definições de algumas terminologias tais como “Tecnologia da Informação e Comunicação (TIC)”, “Fluência Digital”, “Pensamento Computacional”, as quais podem não ser conhecidas por pessoas que não tenham o conhecimento da área da Computação. Outro grande diferencial desta nova versão foi a inclusão de um mapeamento das competências específicas da Computação em competências listadas na Base Nacional Comum Curricular (BNCC).

A SBC não especifica quais ferramentas e materiais os professores devem utilizar ou como eles devem utilizá-los para desenvolver e ensinar as habilidades da Computação aos alunos. Conforme especificado no documento, estes critérios devem ser definidos por cada instituição de ensino, uma vez que as metodologias de ensino empregadas e os recursos disponíveis em cada uma delas não são os mesmos. Além disso, associar ferramentas e materiais aos conteúdos seria um grande desafio devido à constante e rápida modificação e aprimoramento da tecnologia digital nos dias de hoje. Estas especificações acarretariam em uma constante reformulação destas diretrizes. Portanto, a experiência e conhecimento da área da Computação tornam-se aspectos ainda mais relevantes para professores que pretendem se basear nas diretrizes da SBC para ensinar a Computação.

2.2.3 Currículo de Referência em Tecnologia e Computação

O Currículo de Referência em Tecnologia e Computação (CRTC) foi concebido pelo Centro de Inovação para a Educação Brasileira (CIEB) como uma proposta curricular para nortear escolas e seus professores na inclusão da tecnologia digital e Computação nos currículos. O CRTC engloba duas etapas da Educação Básica brasileira: a Educação Infantil e a Educação Fundamental (Ensino Fundamental I e Ensino Fundamental II).

A realidade brasileira no que diz respeito à infraestrutura das instituições de ensino pública é bastante precária se comparada àquela dos países de primeiro mundo. Logo, este aspecto deve ser levado em consideração durante o processo de elaboração de currículos e propostas metodológicas para inclusão da Computação na Educação Básica brasileira. Neste sentido, o CIEB indica que *“a fim de propor um currículo inovador e ao mesmo tempo compatível com a realidade escolar brasileira, foram pesquisadas diversas referências internacionais e nacionais”* [CIEB, 2017]. Além dos referenciais

teóricos propostos pela SBC e a BNCC, o CIEB se baseou nos seguintes documentos: *Currículo da Cidade de São Paulo: Tecnologias para Aprendizagem*, *Componente Curricular de Tecnologia da Austrália*, *Next Generation Science Standards (NGSS) dos Estados Unidos* e o *National Curriculum for Computing do Reino Unido*.

A estrutura do currículo, com base na divisão da Computação feita pela SBC, é composta por três eixos principais: (i) cultura digital, (ii) tecnologia digital e (iii) pensamento computacional. A cada um desses eixos estão associados conceitos, que podem ser vistos como subdivisões que abordam temas mais específicos relacionados a eles. A Figura 2.1, disponibilizada no currículo, demonstra a sua estrutura completa:



Figura 2.1: Estrutura do CRTC. Fonte: [CIEB, 2017]

O currículo define um total de 147 habilidades derivadas dos dez conceitos apresentados na figura acima. Estas habilidades representam o que é esperado que os alunos desenvolvam ao longo dos anos. Os dez conceitos do currículo são trabalhados em todos os anos escolares abordados pelo currículo. A Figura 2.2, retirada do currículo, indica as orientações para professores trabalhando o conceito de Algoritmos do eixo Pensamento Computacional com turmas do 6º ano do Ensino Fundamental.

Além das habilidades do currículo que os alunos devem desenvolver, professores têm a sua disposição orientações a respeito de como trabalhar a habilidade com os alunos, como avaliar se a habilidade foi ou não desenvolvida por eles, habilidades e competências da BNCC associadas ao conceito, e materiais de referência. Ademais, para cada ano escolar, são indicados o nível de maturidade da escola e do professor necessários para que o conceito seja trabalhado. O nível de maturidade do professor está relacionado ao nível de conhecimento dos professores sobre a informática e a Com-



Figura 2.2: Exemplo de orientações para o 6º ano do EF: Fonte: [CIEB, 2017]

putação. Por outro lado, o nível de maturidade da escola representa quão inclusa a tecnologia está nas práticas escolares. É levada em consideração não somente a presença ou não da tecnologia digital, mas também como a tecnologia digital é utilizada nos processos de ensino e aprendizagem.

2.3 Materiais de Apoio ao Ensino da Computação e Pensamento Computacional

Visto que instituições como o MEC, a SBC, e o CIEB têm reconhecido a importância da Computação para alunos e publicado referenciais e diretrizes para o ensino da Computação na Educação Básica brasileira, é essencial que materiais de apoio sejam elaborados para guiar o processo de ensino-aprendizagem. Nesta seção nós apresentamos livros didáticos, a iniciativa Programaê e algumas sequências didáticas elaboradas por pesquisadores brasileiros.

2.3.1 Livros da Série Computação Fundamental

Os livros da série “Computação Fundamental”⁶ são livros didáticos publicados online por Santana et al. [2019] para o ensino da Computação em cada um dos anos do Ensino Fundamental II. Os livros foram elaborados levando em consideração os eixos

⁶Computação Fundamental ↗

temáticos abordados no *K-12 Computer Science Standards* [Seehorn et al., 2011]. Eles são compostos por uma série de atividades para professores utilizarem para o ensino da Computação. Todos os quatro livros foram estruturados em 4 unidades, cada uma contendo sete ou oito aulas, totalizando, para cada ano, 30 aulas com duração prevista de duas horas. Os autores publicaram duas versões do livro para cada ano do EF II: uma a ser utilizada pelos professores e a outra a ser utilizada pelos alunos. A versão dedicada aos professores contém a proposta de ensino do livro, os objetivos, conteúdo, atividades, resultados de aprendizagem e materiais de cada aula. Nele também são feitas as descrições detalhadas de como os professores devem conduzir cada aula. Já a versão dos alunos possui os conteúdos a serem lidos e atividades escritas a serem realizadas pelos alunos. Além disso, ao fim de cada aula, o livro do aluno contém um espaço para ser utilizado como diário de bordo para que os alunos registrem suas experiências e impressões sobre as aulas. Em determinadas aulas, onde são requeridos conhecimentos mais avançados da área da Computação, são descritos e explanados os tópicos relevantes em ambos os livros para que os professores e seus alunos consigam juntos desempenhar o seu papel no processo de ensino-aprendizagem de forma eficiente. Em seguida, apresentamos, para cada um dos quatro livros publicados pelos autores, os seus eixos de aprendizagem, suas unidades e os principais conteúdos abordados nelas.

“*Computação e Eu*” foi o primeiro livro da série publicado pelos autores. O livro foi elaborado para ser utilizado com turmas do 6º ano do Ensino Fundamental [Santana et al., 2019]. O “*Computação e Eu*” define quatro eixos de aprendizagem para os alunos do 6º ano: (i) Pensamento Computacional, (ii) Colaboração, (iii) Práticas de Programação e Computação e (iv) Computadores e Dispositivos de Comunicação. Para cada eixo são definidas um conjunto de capacidades a serem desenvolvidas pelos alunos, as quais são referidas como resultados de aprendizagem no livro. Os quatro eixos e seus respectivos resultados de aprendizagem são explorados nas quatro unidades do livro com maior ou menor intensidade consoante com o assunto abordado em cada unidade. A primeira unidade introduz a Computação aos alunos, abordando temas tais como a representação de números, imagens e hardware. Na segunda unidade é abordado o uso do computador para tarefas comuns (e.g.: manuseio do mouse, digitação, buscas online, editores de textos). A terceira unidade introduz a programação aos estudantes. Para isso, são utilizadas tarefas a serem executadas no Scratch, ferramenta que faz o uso da ludicidade e programação em blocos. Por fim, a quarta unidade tem o objetivo de levar os alunos a aprimorarem a programação aprendida até então. Alguns dos conteúdos abordados nesta unidade são a depuração, paralelismo, métodos de ordenação, e blocos de comando mais complexos no Scratch.

O segundo livro da série elaborado para alunos do 7º ano foi intitulado “*Computa-*

ção e Comunidade”. Ele é estruturado da mesma forma que o livro anterior, possuindo quatro unidades; no entanto, neste livro são definidos cinco eixos de aprendizagem que são explorados ao longo das unidades. Os quatro primeiros eixos são os mesmos do livro “*Computação e Eu*” (i.e.: Pensamento Computacional, Colaboração, Prática de Programação e Computação e Computadores e Dispositivos de Comunicação). O quinto eixo introduzido neste ano (i.e.: Impactos Comunitários, Globais e Éticos) aborda a influência social que a tecnologia exerce bem como os bons e maus usos dela [Araújo et al., 2019]. Na *Unidade I - Desenhando*, são abordados comandos para desenhar figuras geométricas no Scratch, utilizando princípios similares aos daqueles idealizados por Solomon & Papert [1976] e a linguagem LOGO desenvolvida por ele. Além dos comandos para desenhar, são abordados conteúdos da matemática (e.g.: plano cartesiano e ângulos), da Computação (e.g.: criptografia), redes sociais e bullying e *cyberbullying*. Na *Unidade II - Brincando com a Programação* são explorados outros comandos e funcionalidades no Scratch, a história da Computação e os seus impactos na sociedade e tópicos mais avançados da Computação (e.g.: algoritmos de busca e inteligência artificial). A *Unidade III - Criando Jogos (Parte 1)* continua trabalhando com a programação através do desenvolvimentos de jogos 2D no Scratch. Com exceção da sexta aula, onde os alunos aprendem sobre ferramentas de criação de apresentação de slides, em cada uma das aulas da terceira unidade os alunos desenvolvem um jogo específico. Por fim, a *Unidade IV - Criando Jogos (Parte 2)* tem o mesmo objetivo da unidade anterior, iniciando com a criação de três jogos nas três primeiras aulas e finalizando com os alunos projetando os seus próprios jogos na quarta e quinta aula. Ao fim da unidade, as ferramentas para criação de slides são novamente abordadas e utilizadas para que os alunos criem suas apresentações sobre os seus jogos.

Na proposta do “*Computação e Sociedade*”, livro destinado às turmas do oitavo ano do Ensino Fundamental II, os autores consideram que os alunos estão deixando de exercer atividades em pequenos grupos (e.g.: família e escola) e passando a interagir em grupos sociais mais amplos e diversos devido a influência da internet e redes sociais [Santana et al., 2020]. Os eixos de aprendizagem abordados neste livro são os mesmos eixos definidos no livro do sétimo ano: (i) Pensamento Computacional, (ii) Práticas de Computação e Programação, (iii) Colaboração, (iv) Computadores e Dispositivos de Comunicação e (v) Impactos Comunitários, Globais e Éticos. A primeira unidade do livro, *Trabalhando com Colaboração*, explora a colaboração de usuários em diferentes tipos de ferramentas e contextos (e.g.: ferramentas de e-mail, Google Docs, Google Slides, Google Drive e redes sociais). Aspectos éticos relacionados às redes sociais e às informações disponibilizadas e compartilhadas nessas redes também são abordados. A *Unidade II - Trabalhando com Robôs* faz o uso da robótica educacional ao trabalhar com


o robô mBot⁷ e a ferramenta mBlock⁸ para programar comportamentos do robô através da linguagem de programação em blocos. Diferentes funcionalidades e comportamentos do robô são exploradas em cada uma das aulas desta unidade (e.g.: movimentação, LEDs, sensores, etc.). Na *Unidade III - Manipulação de Imagens*, os alunos continuam a explorar a programação através da ferramenta GPBlocks⁹ para criação de filtros e efeitos de imagens. Por fim, a linguagem de programação Python é abordada na *Unidade IV - Desenhando com Python*. Conceitos similares a aqueles abordados na *Unidade I - Desenhando* do livro “*Computação e Comunidade*” são explorados através de uma biblioteca Python para criação de figuras geométricas.

O último livro da série, que é dedicado aos alunos do 9º ano do Ensino Fundamental II tem o nome de “*Computação e o Mundo*”. A proposta deste livro é que os alunos tenham experiências mais aprofundadas com a Computação através da resolução de problemas utilizando software e hardware e que eles relacionem estas experiências e conhecimentos adquiridos com o mundo que vivem [Araujo et al., 2020]. Ele aborda os mesmos cinco eixos de aprendizagem abordados nos livros do 7º e 8º anos. Assim como os três primeiros livros da série, o “*Computação e o Mundo*” também foi estruturado em quatro unidades. Na *Unidade I - Simulações*, os alunos exploram a utilização da matemática para criação de modelos computacionais que explicam fenômenos da natureza. O Scratch é a ferramenta utilizada nas aulas para criação dos modelos. As aulas da *Unidade II - Manipulação de Imagens* têm como objetivo aprofundar o conhecimento dos alunos sobre a linguagem de programação Python através da manipulação de imagens. Nestas aulas, a IDE JES (Jython Environment for Students)¹⁰ é utilizada para que os alunos manipulem propriedades das imagens e desenvolvam filtros. A *Unidade III - Programação em Python* aborda a utilização da linguagem de programação Python para implementação de diferentes tipos de projetos tais como o jogo da forca, jogo da velha, sistema de enquetes, sistema de criptografia, etc. Em cada uma das oito aulas da unidade um projeto diferente é implementado. Por fim, a *Unidade IV - Arduino* aprofunda a terceira unidade do livro do 8º ano com atividades em que os alunos devem construir circuitos utilizando o Arduino e programá-los através da ferramenta mBlock, que também é abordada na terceira unidade do livro anterior.

⁷mBot 

⁸mBlock 

⁹GPBlocks 


¹⁰JES 

2.3.2 Programaê!

O Programaê! é uma iniciativa que visa disseminar o ensino da lógica de programação e cultura digital para estudantes. A plataforma¹¹ disponibiliza duas visões principais para as pessoas que a estão acessando: uma para pessoas interessadas em aprender e outra para pessoas interessadas em ensinar. Desta forma, a plataforma pode ser utilizada por ambos alunos e professores.

A página dedicada aos alunos indica diversos tipos de atividades que podem ser acessadas gratuitamente pelos alunos. Elas são divididas em quatro categorias, as quais indicam o propósito das atividades. As cinco categorias ou propósitos são as seguintes: *Para Começar*, *Para Criar*, *Para Jogar*, *Para Mergulhar nos Códigos*. As atividades indicadas aos alunos nas três primeiras categorias estão presentes na plataforma Code.org, Scratch ou fazem parte da iniciativa Hora do Código. A última categoria contém atividades avançadas disponíveis na plataforma e envolvem o aprendizado de linguagens de programação convencionais. Três delas, abordando a criação de banco de dados, de páginas web e de desenhos e animações utilizando JavaScript estão disponíveis na plataforma Khan Academy. Além disso, há a indicação do Silent Teacher que é um jogo com diversas perguntas para que os alunos aprendam sobre a lógica de programação e as suas regras unicamente através da exploração e da tentativa e erro, não contendo nenhum tipo de explicação ou tutorial.

A seção da plataforma dedicada aos professores dispõe de diversas atividades e materiais de apoio para guiar os professores que estão pretendendo conduzir as atividades em suas aulas. A plataforma estrutura as atividades em três partes: Ensino Fundamental I, Ensino Fundamental II e Ensino Médio. Para cada uma, é indicado um conjunto de atividades multidisciplinares que alinham a lógica de programação a disciplinas regulares lecionadas pelos professores. Na etapa do Ensino Fundamental I, há seis atividades disponíveis que abordam o *cyberbullying* como tema transversal, a matemática (e.g.: frações, geometria, unidades de medida), a educação física (e.g.: coordenação motora, lateralidade), geografia (referências espaciais, tipos de moradia) e língua portuguesa (concordância verbal e nominal). Dentre as seis atividades, cinco são completamente desplugadas e apenas aquela que aborda o *cyberbullying* contém aulas com atividades plugadas envolvendo a criação de uma narrativa digital. Nove atividades estão disponíveis para os alunos do Ensino Fundamental II, sendo abordados os seguintes tópicos: matemática (e.g.: geometria espacial), a cultura digital como tema transversal, a língua inglesa (e.g.: noções básicas de comunicação, oralidade, leitura e escrita), a língua portuguesa (e.g.: gênero textual), ciências (e.g.: nutrição, pirâmide

¹¹Programaê! 

alimentar, reino Fungi, espécies unicelulares e pluricelulares), arte (e.g.: processos criativos, história em quadrinhos) e a cultura brasileira. As atividades destinadas aos alunos do Ensino Fundamental são em sua grande maioria plugadas (7 atividades) e apenas duas delas são completamente desplugadas. Por fim, seis atividades estão disponíveis para serem conduzidas com alunos do Ensino Médio, abordando temas transversais (e.g.: conservação do meio ambiente, segurança ao navegar na internet e redes sociais), biologia (e.g.: corpo humano, relações ecológicas), química (e.g.: relações químicas, ligações químicas) e física (e.g.: movimento uniforme, movimento uniforme variado). Cinco das atividades do ensino médio são plugadas e envolvem a utilização do Scratch e outros programas tais como simuladores. A atividade *Robô-escova e o tabuleiro "QUIZ"*, que envolve a criação de um robô que se movimenta através do vibracall (motor responsável pela vibração em celulares), é a única completamente desplugada.

As atividades e conteúdos oferecidos pelo Programaê! são interessantes porque permitem o ensino de Computação junto ao ensino e conteúdo de disciplinas escolares regulares. Assim, professores lecionando as disciplinas regulares abordadas nestas atividades podem utilizá-las em salas de aula para que a lógica de programação seja traga para a Educação Básica e para que os seus alunos desenvolvam as habilidades do Pensamento Computacional enquanto aprendem sobre conteúdos relevantes de suas disciplinas. Contudo, elas podem ser difíceis de serem combinadas com o planejamento das aulas feito pelos professores e substituírem aulas tradicionais que abordariam assuntos equivalentes. As atividades propostas na plataforma foram planejadas para serem executadas em múltiplas aulas e a carga horária disponível para os professores utilizarem pode não ser suficiente para abordar todo o conteúdo proposto. As atividades mais curtas da plataforma têm duração de 2 e 3 aulas e a maior parte das atividades têm duração de mais de 3 aulas, podendo chegar a até 10 aulas de 50 minutos cada.

2.3.3 Sequências didáticas propostas por pesquisadores

Diversas pesquisas envolvendo o PC têm sido feitas no Brasil, sendo que a maioria delas são relatos de experiência e estudos de caso [Santos et al., 2018]. Nestas pesquisas, similarmente ao nosso trabalho, sequências didáticas elaboradas pelos pesquisadores para o desenvolvimento do PC por meio da lógica de programação são aplicadas e avaliadas.

O trabalho de Almeida & Junior [2020] relatou a experiência da aplicação de uma sequência didática de quatro aulas para o desenvolvimento do PC com alunos do 4º ano do Ensino Fundamental. Os autores detalham para cada aula os conteúdos abordados, seus objetivos, as atividades conduzidas pelos professores e as atividades conduzidas

pelos alunos. Nesta sequência didática proposta foram adotadas atividades desplugadas, a plataforma Hora do Código e o Scratch. Em [Duarte et al., 2017], uma sequência didática de seis aulas foi apresentada e aplicada com alunos do 6º, 7º e 8º anos. Um tema específico da Computação (e.g.: algoritmos, sequenciamento de passos, laços de repetição, depuração, condicionais) é trabalhado em cada aula. Na proposta dos autores, em cada aula o tema é primeiramente introduzido e formalizado teoricamente para que então ele seja praticado através de atividades desplugadas e no Scratch. Uma sequência didática de seis aulas abordando temas específicos da Computação em cada uma delas também foi elaborada no trabalho de Carlos et al. [2018]. Similarmente, na proposta dos autores, os conceitos são primeiramente introduzidos teoricamente e depois praticados; no entanto, os autores dividiram a parte prática em dois momentos: (i) prática inicial por meio de uma dinâmica desplugada e, por fim, (ii) prática por meio de atividades de programação realizadas em uma ferramenta digital (e.g.: Code Studio, Scratch, etc.).

Em muitos dos trabalhos, tais como os que foram apresentados previamente, é relatado e descrito como foi o processo de preparação e condução das aulas, as atividades, os resultados obtidos e lições aprendidas com as experiências. Contudo, os autores não disponibilizam diretamente os recursos didáticos adotados nas experiências (e.g.: roteiros para condução das aulas e atividades, apresentações de slides, folhas de atividades, exemplos prontos de projetos a serem criados pelos alunos, etc.). A disponibilização destes materiais tornaria estes estudos mais replicáveis para outros pesquisadores e facilitaria para que professores, especialmente aqueles sem muita experiência com a Computação, adotem as abordagens e recursos didáticos utilizados.

Neste trabalho, similarmente aos materiais de apoio apresentados previamente, temos como objetivo a criação de uma sequência didática para a introdução da Computação e do PC. Diferentemente das diretrizes da SBC e do CRTCC, que foram criados considerando que os alunos terão a oportunidade de aprender sobre a Computação desde a Educação Infantil e Ensino Fundamental I, o nosso foco é apenas o Ensino Fundamental II, especificamente alunos destes anos que não aprenderam sobre a Computação e o PC nos anos anteriores ao EF II. Conforme será explicado no Capítulo 4, estamos adotando uma nova abordagem para criação da sequência didática ao nos fundamentar nas Metáforas de Perspectivas Culturais (MPCs), introduzidas no próximo capítulo, para definir recursos didáticos a serem adotados na sequência para que a Computação e PC sejam introduzidos a alunos gradualmente.

Capítulo 3

Fundamentação Teórica

Nos fundamentamos nas Metáforas de Perspectivas Culturais (MPCs) para conduzir este trabalho. Assim, neste capítulo nós, primeiramente, introduzimos a Engenharia Semiótica, a qual é uma teoria de onde as MPCs se originaram. Em seguida, apresentamos as MPCs como ferramenta epistêmica e apresentamos e detalhamos as cinco metáforas que fazem parte dela. Por fim, apresentamos o contexto do Pensamento Computacional como uma cultura e o que as MPCs representam neste contexto.

3.1 Engenharia Semiótica

A Engenharia Semiótica (EngSem) é uma teoria explicativa da área de Interação Humano-Computador (IHC) fundamentada na Semiótica (área de estudo dos signos e significação) [de Souza, 2005]. A EngSem considera que a interação com sistemas computacionais é uma conversa entre o projetista da interface e os seus usuários. Essa comunicação é denominada metacomunicação e ela ocorre por meio do próprio sistema, especificamente pela sua interface. A metacomunicação é um processo regido por três atores: designer, usuário e sistema, também denominado preposto do designer.

O termo *Designer* é comumente utilizado por pesquisadores da Interação Humano-Computador (IHC). Aqui, ele não se refere a apenas uma pessoa, mas a toda a equipe responsável pelo processo de projeto (e desenvolvimento) dos sistemas. A metacomunicação só acontece unidirecionalmente do designer para o usuário, pois o usuário não pode recorrer ao designer enquanto ele está interagindo com o sistema. Portanto, considera-se que o sistema está representando o designer no processo de metacomunicação, sendo o seu **preposto**. Compreende-se a interface do sistema como uma metamensagem que o designer, apesar de estar ausente, transmite ao usuário, o informando das decisões de design em relação a quem o sistema se destina, a quais

objetivos o sistema permite o usuário atingir e a como interagir com o sistema para que os objetivos sejam atingidos.

A propriedade do preposto do designer em transmitir aos seus usuários a solução de design é denominada comunicabilidade. Ela está diretamente relacionada à qualidade do sistema. Para que sistemas possuam alta comunicabilidade, é importante que a interpretação do sistema pelos usuários esteja condizente com as intenções do designer que o projetou.

A EngSem vem propondo um conjunto de ferramentas epistêmicas a fim de fornecer meios para apoiar os designers no entendimento e reflexão sobre a solução sendo projetada ou seu impacto nos usuários finais. O ferramental disponibilizado tem por objetivo fomentar a reflexão durante o processo de design, permitindo que os designers tenham uma melhor visão do problema, experimentem soluções e as avaliem. As Metáforas de Perspectivas Culturais são uma dessas ferramentas epistêmicas.

3.2 As Metáforas de Perspectivas Culturais

Salgado [2011b] se fundamentou na Engenharia Semiótica e criaram as Metáforas de Perspectivas Culturais (MPC), as quais “são uma ferramenta conceitual criada por cujo propósito é informar e guiar o projeto e avaliação da interação sempre que comunicações explícitas sobre diversidade cultural fazem parte da intenção de *design*.” [Salgado et al., 2011, pág. 3, tradução do autor]. Logo, as MPCs são consideradas uma ferramenta epistêmica, uma vez que elas levam os designers a considerar aspectos culturais em seus processos reflexivos sobre os artefatos que eles estão desenvolvendo.

As metáforas indicam que a interação com o sistema computacional pode ser vista como uma “viagem” e que os seus usuários são “viajantes”. Assim, considera-se que as pessoas utilizando determinado sistema estejam em uma espécie de jornada, onde elas estariam explorando uma cultura desconhecida, o sistema. Um total de cinco metáforas fazem parte da ferramenta: Viajante Doméstico, Observador à Distância, Estrangeiro em Visita Guiada, Estrangeiro com Tradutor e Estrangeiro sem Tradutor. A Figura 3.1 apresenta as metáforas e o contínuo de exploração cultural:

As metáforas nas extremidades do continuum, o *Viajante Doméstico* e o *Estrangeiro sem Tradutor* representam respectivamente o total isolamento e imersão cultural do usuário. Dessa forma, a mediação cultural, ou seja, o contraste entre a cultura nativa e a cultura estrangeira não está presente em nenhum destes cenários. O propósito da MPC *Viajante Doméstico* é o de não expor os usuários do sistema a elementos estrangeiros. Por outro lado, a MPC *Estrangeiro sem Tradutor* é adotada quando o objetivo

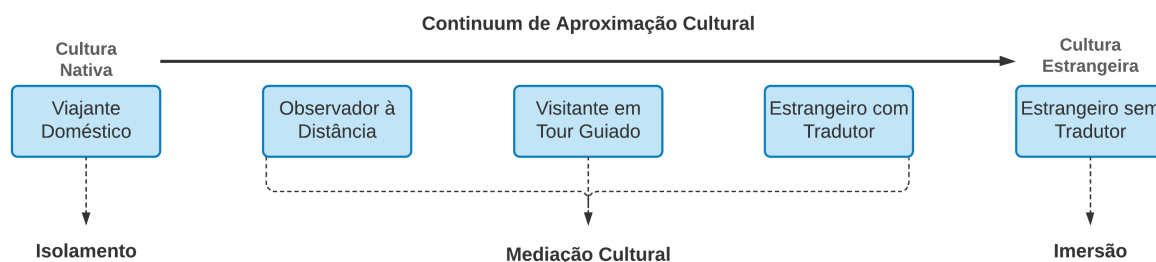


Figura 3.1: Metáforas de Perspectivas Culturais

é permitir com que os usuários se imerjam completamente na cultura estrangeira. As três metáforas no centro do continuum (i.e.: Observador à Distância, Visitante em Tour Guiado, Estrangeiro com Tradutor) há a mediação cultural e a cultura estrangeira é apresentada ao usuário de diferentes maneiras e está presente em diferentes níveis de intensidade.

A primeira metáfora, **Viajante Doméstico**, faz alusão a uma pessoa que está completamente isolada em sua própria cultura. Logo, essa pessoa não está sendo exposta explicitamente a nenhum tipo de linguagem ou prática estrangeira. Ao utilizá-la, o designer faz com que a cultura estrangeira fique “escondida” aos olhos dos usuários, fazendo com que eles interajam apenas com elementos de sua própria cultura. Por exemplo, um site de culinária que apresenta receitas de pratos (inclusive estrangeiros) em português como se eles fossem pratos do Brasil, utilizando descrições, terminologias, unidades de medida e ingredientes utilizados em receitas brasileiras.

Na metáfora do **Observador à Distância**, os usuários, inseridos em sua própria cultura, são expostos à cultura estrangeira de maneira superficial. Os elementos da cultura estrangeira são apresentados aos usuários apenas como informações sucintas, permitindo que eles aprendam um pouco sobre a cultura estrangeira. Um exemplo de design guiado por essa metáfora é um site de culinária estrangeira que, embora seja escrito em português, dá a oportunidade que os usuários aprendam sobre os nomes originais dos pratos ou sobre os países de onde os pratos se originaram. A diferença dessa MPC para a anterior é a disponibilização de algumas informações sobre a cultura estrangeira para os usuários que estão interessados em aprender sobre elas.

A terceira MPC, o **Viajante em Tour Guiado**, proporciona aos usuários uma mediação cultural mais profunda que a etapa anterior. Nesta fase, os usuários estão mais próximos da cultura estrangeira e conduzem algumas práticas dessa cultura. Contudo, essas práticas ainda são mediadas por elementos da cultura nativa dos usuários com o propósito de contrastar ambas as culturas e apresentar aos usuários uma interpretação da cultura estrangeira. Ainda no contexto de sites de culinária, um exemplo de design guiado por essa MPC é a apresentação dos nomes dos pratos e também

dos ingredientes em ambas línguas nativa e estrangeira, a utilização de unidades de medida tanto no formato utilizado na cultura nativa, quanto no formato utilizado na cultura estrangeira ou a disponibilização de explicações e imagens detalhando pratos estrangeiros.

A metáfora **Estrangeiro com Tradutor** é adotada quando a intenção é dar a oportunidade para que os usuários do sistema experimentem a cultura estrangeira de maneira similar aos nativos dessa cultura. Contudo, estes usuários têm à sua disposição a tradução textual da língua estrangeira para suas línguas maternas. Sites que disponibilizam receitas traduzidas de pratos estrangeiros da mesma forma que eles seriam apresentados aos nativos da cultura estrangeira é um exemplo de utilização dessa MPC. Embora os usuários tenham acesso a tradução textual, as terminologias e unidades de medida utilizadas são estrangeiras e não há explicações adicionais sobre a cultura estrangeira e os pratos.

Finalmente, a quinta e última MPC, o **Estrangeiro sem Tradutor**, representa o oposto da primeira MPC e indica que o usuário está completamente imerso na cultura estrangeira. A utilização dessa metáfora em um sistema computacional tem como o propósito promover uma imersão cultural em que usuários utilizarão unicamente a linguagem da cultura estrangeira e conduzirão práticas dessa cultura da mesma forma que os nativos dela fariam. No contexto de sites de receitas de internacionais, esta MPC estaria presente em sites que apresentam as receitas da forma como ela seria apresentada aos nativos da cultura estrangeira, ou seja, sem oferecer nenhuma tradução textual e explicações adicionais aos usuários.

Vale ressaltar que as MPCs foram propostas com o objetivo de apoiar o designer de sistemas interativos a refletir sobre como apresentar uma cultura estrangeira ao usuário. [No entanto, a natureza das MPCs permitem que elas sejam aplicadas em uma grande variedade de contextos \[Silva et al., 2013; Leitão et al., 2017; Oliveira et al., 2018\]. Neste trabalho, estamos originalmente propondo e investigando o uso das MPCs não como uma ferramenta epistêmica, mas como um guia para se apresentar de forma gradual uma cultura estrangeira ao usuário.](#) Para isso, não selecionamos uma das metáforas, mas sim o seu conjunto, que representa um contínuo entre uma visão de isolamento até a imersão cultural. No contexto deste trabalho, consideramos o Pensamento Computacional como a cultura estrangeira a ser apresentada, como explicaremos na próxima seção.

3.3 O Pensamento Computacional como uma Cultura

Conforme indicado no trabalho conduzido por Oliveira [2020], o PC pode ser considerado uma cultura que possui os seguintes aspectos: conceitos de programação, linguagens de programação textual e visual, raciocínio lógico e raciocínio criativo. Neste trabalho, adotamos essa visão com o objetivo de utilizarmos as MPCs como fundamento para classificarmos recursos didáticos para o ensino da Computação e PC, e em seguida, selecionarmos um conjunto desses recursos para serem aplicados na prática tal que o PC fosse introduzido gradualmente, como uma cultura, a alunos do Ensino Fundamental II. Assim, consideramos o PC como uma cultura estrangeira e que a cultura nativa dos alunos fosse uma cultura sem experiência com a Computação e sem habilidades do PC desenvolvidas. Em seguida, descrevemos o significado das cinco MPCs ao considerarmos o PC e a Computação como uma cultura estrangeira.

O **Viajante Doméstico** (VD) indica que os alunos não estão sendo expostos diretamente a tópicos da área da Computação ou conduzindo atividades que requerem conhecimentos dessa área. Alguns exemplos de atividades utilizadas no ensino de PC no Brasil que podem ser caracterizadas nessa metáfora são os jogos de lógica ou jogos de tabuleiro, os quais os alunos têm a capacidade de jogar sem que eles tenham as habilidades do PC devidamente desenvolvidas. Essas atividades normalmente são utilizadas para estimular o raciocínio lógico e o pensamento criativo dos alunos.

Conforme apresentado previamente, a metáfora do **Observador à Distância** (OD) é uma metáfora que indica que a cultura estrangeira está sendo apresentada primariamente como informação e menos como prática. Assim como no VD, as práticas conduzidas são familiares aos alunos, porém eles estão conscientes de que estão em contato com a cultura do PC. Por exemplo, explicar aos alunos que os passos seguidos por eles ao solucionar determinado jogo de lógica é um algoritmo.

Atividades e ferramentas associadas à terceira MPC, o **Visitante em Tour Guiado**, são as atividades mais simples dentre aquelas analisadas que permitem que os alunos conduzam práticas e tenham experiência com a cultura estrangeira diretamente. As ferramentas e atividades associadas à essa metáfora dão a oportunidade para que os alunos consigam desenvolver o PC por meio de atividades práticas, porém com limitações se comparadas às práticas conduzidas por nativos da cultura estrangeira. Por exemplo, atividades em que os alunos devem construir algoritmos um conjunto restrito de comandos de movimento para conduzir um personagem até determinada localização.

A metáfora do **Estrangeiro com Tradutor** representa a experiência da cultura estrangeira de maneira equivalente à experiência que os nativos dessa cultura têm, porém ainda com intermediação da tradução para facilitar o processo. Portanto, no contexto do PC e Computação, essa MPC está associada a ferramentas com potencial similar às ferramentas utilizadas por pessoas familiarizadas com a Computação e com habilidades do PC desenvolvidas e que oferecem aos usuários mecanismos de auxílio tais como a apresentação da linguagem de programação de forma visual (ao invés da textual).

Por fim, o **Estrangeiro sem Tradutor** indica a exposição à cultura estrangeira e condução das práticas dessa cultura sem nenhum tipo de mediação cultural para apoio. Consequentemente, as ferramentas e atividades representadas por essa MPC são as mesmas utilizadas pelas pessoas da cultura estrangeira. No Brasil, dentre as ferramentas associadas à MPC do estrangeiro sem tradutor, as IDEs (i.e.: Ambientes de Desenvolvimento Integrado) são as mais comumente utilizadas a fim de proporcionar uma experiência equivalente àquelas com que pessoas que trabalham com a Computação e que estão familiarizadas com a área (e.g.: profissionais de TI) têm.

Neste trabalho, nosso objetivo final é propor uma sequência didática baseada nas MPCs, considerando o PC como uma cultura estrangeira. Nesta sequência, as atividades se iniciam com atividades associadas ao Viajante Doméstico e, gradativamente, novos conceitos são apresentados e novas atividades são realizadas, de forma que o aluno percorra o *continuum* das metáforas até chegar à metáfora Estrangeiro sem Tradutor. No próximo capítulo, apresentaremos a metodologia adotada para a geração desta sequência didática.

Capítulo 4

Metodologia

Neste capítulo, apresentamos a metodologia adotada em nosso trabalho. Ele foi dividido em duas grandes etapas: (i) geração da sequência didática para o ensino da Computação e PC e (ii) condução de uma pesquisa-ação para avaliar e aprimorar a sequência.

4.1 Geração da Sequência Didática

Nós propomos a utilização das MPCs como fundamento para definir sequências didáticas para gradativamente introduzir a Computação a alunos do Ensino Fundamental II e fomentar o PC. A Figura 4.1 ilustra essa proposta, que está estruturada em cinco etapas, representando as MPCs e indicando o quão próximo da cultura estrangeira estariam os recursos didáticos aplicados em cada etapa. Os recursos didáticos utilizados nas etapas posteriores permitem que os alunos experimentem e pratiquem a cultura estrangeira com maior intensidade que os recursos utilizados nas etapas anteriores.

No que diz respeito ao perfil dos alunos aos quais essa proposta tem como público, consideramos alunos do Ensino Fundamental II (EF II) com pouca ou nenhuma experiência prévia em Computação. Primeiramente, muitos dos conteúdos a serem trabalhados e explorados, especialmente nas últimas etapas da proposta, requerem maior maturidade em relação à Matemática e ao pensamento lógico. Em segundo lugar, o contexto abordado pelas MPCs vai desde o isolamento na cultura nativa até a imersão completa na cultura estrangeira. Analogamente, a sequência didática à luz das MPCs também parte deste mesmo contexto de isolamento cultural, ou seja, alunos sem experiência com a Computação. Este contexto de isolamento cultural é especialmente importante para nós uma vez que o Brasil está atrasado quanto à inclusão da Computação na Educação Básica como detalhamos no Capítulo 2. Espera-se que no futuro

atualizei a descrição dos recursos das 2 primeiras MPC

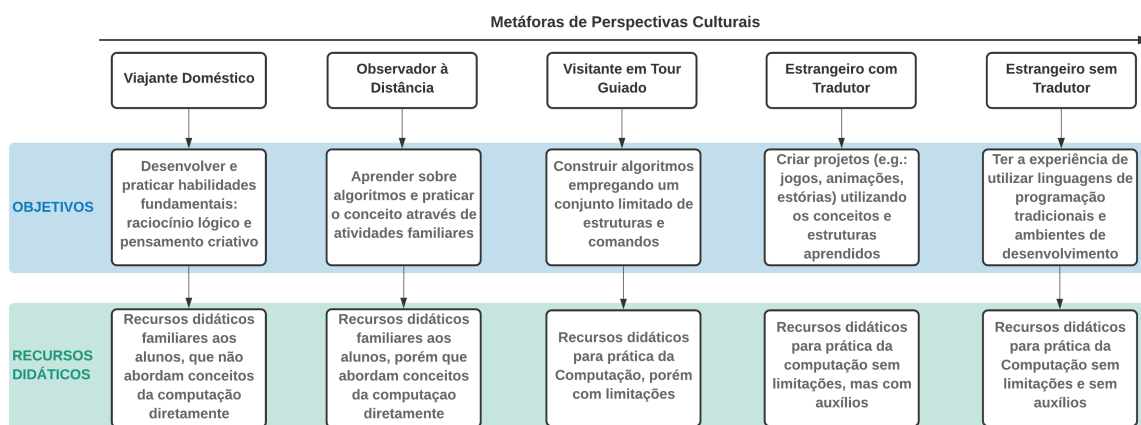


Figura 4.1: Base para a sequência didática à luz das MPCs

esta realidade mude com os esforços da SBC, do MEC, pesquisadores e professores que buscam fazer com que a Computação seja parte da realidade dos estudantes desde os anos iniciais do Ensino Fundamental. No entanto, enquanto isto ainda não é realidade no Brasil, é necessário considerarmos os alunos que estão cursando os anos finais do Ensino Fundamental ou Ensino Médio e que não tiveram a oportunidade de ter contato com a Computação ao decorrer de suas vidas escolares.

Com a finalidade de definir as atividades da sequência didática a serem realizadas em cada etapa da proposta, nós realizamos uma classificação de recursos didáticos empregados no Brasil para o ensino da Computação e PC a alunos do Ensino Fundamental. Esta classificação foi, em seguida, utilizada como base para selecionarmos quais recursos didáticos utilizaríamos em cada uma das cinco etapas da sequência didática. Na seção seguinte, apresentamos os procedimentos metodológicos adotados para condução da classificação e seleção dos recursos.

4.1.1 Classificação de Recursos Didáticos à Luz das MPCs

Primeiramente, conduzimos a classificação da complexidade de recursos didáticos empregados no Brasil para o ensino da Computação e PC a alunos do Ensino Fundamental. Utilizamos as MPCs como base para classificação ao considerarmos o PC como uma cultura estrangeira sendo introduzida aos alunos do Ensino Fundamental II. O processo da classificação foi dividido

Dado que queríamos analisar os recursos didáticos e como eles estavam sendo aplicados no Brasil, nós decidimos, primeiramente, buscar por mapeamentos sistemáticos da literatura (MSL) que detalhavam os diversos estudos envolvendo o PC sendo feitos no Brasil. Utilizamos os seguintes critérios para selecionar o MSL mais adequado

ao nosso trabalho:

1. *procedimentos metodológicos devem estar bem definidos*
2. *o ano de publicação dos estudos analisados deve ser amplo, incluindo estudos recentes*
3. *dados sobre os estudos analisados devem estar disponíveis*
4. *a etapa da Educação Básica onde os trabalhos analisados foram aplicados deve ser especificada*

A partir destes critérios, selecionamos o mapeamento conduzido por Santos et al. [2018], que caracterizaram estudos sobre o PC e programação conduzidos na Educação Básica brasileira entre os anos de 2001 e 2016, a qual é uma janela de tempo bastante ampla. Os procedimentos metodológicos aplicados para a condução do mapeamento também foram bem especificados pelos autores. Além disso, eles disponibilizaram aos leitores uma planilha digital contendo informações detalhadas sobre cada um dos estudos analisados tais como as referências, dados extraídos e a etapa da Educação Básica em cada artigo focava. Portanto, o trabalho de Santos et al. [2018] permitiu que selecionássemos apenas os estudos do nosso interesse, que eram estudos conduzidos com alunos do Ensino Fundamental, os quais totalizaram 99 estudos dos 338 analisados pelos autores.

Em seguida, nós conduzimos o nosso próprio MSL utilizando procedimentos metodológicos similares àquelas adotados por Santos et al. [2018] para seleção de estudos realizados entre o período de 2017 e abril de 2019. Criamos uma *string* de busca similar àquela utilizada por Santos et al. [2018], porém especificamos por meio dela que buscávamos por estudos focados no Ensino Fundamental: (*“pensamento computacional” OR “raciocínio computacional”*) AND (*“educação primária” OR “educação secundária” OR “educação básica” OR “ensino fundamental” OR “educação infantil”*). A *string* de busca foi utilizada no Google Scholar e obtivemos um total de 597 publicações. Destas, nós excluimos publicações: (i) duplicadas, (ii) que não focavam no Ensino Fundamental, (iii) com 2 ou menos páginas e (iv) que não detalhavam as abordagens de ensino e recursos didáticos empregados. No fim, 114 publicações permaneceram após a aplicação dos critérios exclusão.

Dessa forma, um total de 223 estudos foram selecionados para leitura completa. Ao lermos estes estudos, vimos que alguns deles caíram em pelo menos uns dos critérios de exclusão definidos acima e, logo, eles foram desconsiderados para a etapa de análise.

Por fim, nós analisamos 165 estudos. A lista destes estudos pode ser encontrada no Apêndice C do trabalho de Oliveira [2020].

Durante a leitura dos artigos, fizemos um levantamento dos recursos didáticos sendo adotados para o ensino da Computação e PC no Brasil. Estes recursos foram então classificados à luz das MPCs ao considerarmos o PC como uma cultura estrangeira conforme detalhado no Capítulo 3. Cada recurso didático foi associado a pelo menos uma MPC, indicando o quão complexo eram os recursos didáticos para alunos do Ensino Fundamental II.

A classificação foi conduzida por dois pesquisadores, sendo um deles o autor deste trabalho; e o outro, a autora do trabalho [Oliveira, 2020]. Cada pesquisador separadamente leu todos os trabalhos selecionados e classificou os recursos didáticos adotados neles. Então, uma triangulação das classificações foi feita com o objetivo de remover possíveis erros na classificação. Desacordos e dúvidas no que diz respeito às classificações dos recursos didáticos foram resolvidos por meio de reuniões entre os dois pesquisadores. Os poucos casos, onde os pesquisadores não chegaram em um acordo, foram discutidos com um terceiro pesquisador a fim de decidir qual seria a classificação final do recurso didático.

Na primeira seção do Capítulo 5, apresentamos o resultado da classificação dos recursos didáticos. É apresentado para cada uma das cinco MPCs, a classificação dos recursos didáticos mais comumente adotados nos estudos selecionados por nós.

4.1.2 Seleção de Recursos Didáticos Didáticos

Nós nos fundamentamos na classificação dos recursos didáticos à luz das MPCs apresentada previamente para selecionar quais recursos educacionais utilizaríamos na sequência didática a ser aplicada na prática. Além dos recursos classificados, procuramos também por outros recursos interessantes que pudessem ser utilizados na sequência. As MPCs também foram utilizadas como base para considerarmos a complexidade destes recursos e se os utilizaríamos ou não.

Buscamos selecionar ao menos um recurso didático para cada MPC, sendo que eles deveriam estar disponíveis gratuitamente para serem utilizadas no contexto das escolas públicas brasileiras. Os detalhes e resultados dessa seleção são apresentados na segunda seção do Capítulo 5.

4.2 Desenho da Pesquisa Ação

Nesta seção, apresentamos a metodologia utilizada na segunda e última etapa do trabalho. Uma Pesquisa-Ação foi realizada nessa etapa para aplicar, avaliar e aprimorar a sequência didática inicial gerada na etapa anterior. Nós apresentamos a caracterização da pesquisa, o contexto em que ela foi conduzida e os instrumentos de coleta de dados adotados.

4.2.1 Caracterização da Pesquisa

A metodologia utilizada neste trabalho é de cunho qualitativo. Pesquisas qualitativas podem ser caracterizadas como pesquisas em profundidade de caráter interpretativo conduzidas em cenários naturais com uma amostra pequena de participantes utilizando múltiplos procedimentos e técnicas de investigação [Nicolaci-da Costa et al., 2004; Creswell, 2014].

Diferentes procedimentos geralmente empregados em pesquisas qualitativas foram utilizados neste trabalho, porém o procedimento central dos estudos conduzidos foi a Pesquisa-Ação, método que ganhou popularidade na academia devido à pesquisa sobre problemas sociais conduzida por Kurt Lewin em 1946 [Adelman, 1993]. O restante das técnicas foram utilizadas para dar suporte à pesquisa-ação ao possibilitar a coleta de diferentes tipos de dados em diferentes momentos da pesquisa.

A Pesquisa-Ação, segundo Tripp [2005], é *“uma forma de investigação-ação que utiliza técnicas de pesquisa consagradas para informar a ação que se decide tomar para melhorar a prática”*. Assim, a pesquisa-ação é uma metodologia que une ações práticas e a pesquisa. Ela não é completamente prática e nem completamente pesquisa, requerendo que pesquisadores realizem intervenções na prática e na pesquisa Tripp [2005]. Koshy [2005] sumariza e caracteriza a pesquisa-ação da seguinte forma: (i) envolve a pesquisa da própria prática, (ii) é emergente, (iii) é participativa, (iv) constrói a teoria a partir da prática, (v) é baseada na situação, (vi) pode ser útil na solução de problemas reais, (vii) trabalha com indivíduos ou grupos com o objetivo comum de aprimorar a prática, (viii) é focada no aprimoramento, (ix) envolve análise, reflexão e avaliação, (x) facilita a mudança através da investigação.

Muitas destas características citadas por Koshy [2005] em seu guia prático são apontadas por ele como vantagens de se utilizar pesquisas-ação e estas vantagens tornam a pesquisa-ação um método apropriado para condução de nossa pesquisa. Primeiramente, as pesquisas-ação acontecem em um contexto e situação específica por estarem ligadas a intervenções práticas, que em nossa pesquisa são cursos de PC mi-

nistrados em salas de aula para alunos do Ensino Fundamental. Em segundo lugar, os próprios pesquisadores podem ser participantes, não sendo necessário que eles se distanciem do contexto pesquisado. Assim, nós atuamos tanto como professores nas salas de aula, quanto pesquisadores que analisaram a experiência.

Conforme abordado anteriormente, em uma pesquisa-ação são feitas diversas iterações no ciclo de preparação, ação, reflexão e modificação. Essa postura exploratória e flexível das pesquisas qualitativas e pesquisas-ação, permite que o planejamento prévio das atividades possa ser adaptado e aprimorado simultaneamente à condução das aulas caso necessário. Em nosso contexto de trabalho, estas adaptações foram em relação à forma que tópicos foram abordados e apresentados, à dificuldade das tarefas dadas aos alunos ou às próprias ferramentas sendo utilizadas. Por isso, a utilização desse método de pesquisa está em alinhamento com um dos objetivos de nosso trabalho, que é o desenvolvimento de uma sequência didática para o ensino da Computação e PC no Ensino Fundamental II.

4.2.2 Contexto da Pesquisa

Neste trabalho foram realizados três estudos para o desenvolvimento das habilidades do Pensamento Computacional, que serão referidos como *Estudo I*, *Estudo II* e *Estudo III*. Todos os estudos foram realizados com alunos do 6º ano do Ensino Fundamental II e tiveram como *locus* o Centro Pedagógico da Escola de Educação Básica e Profissional da UFMG (CP).

O CP é uma escola pública pertencente à UFMG, que atende estudantes do Ensino Fundamental, os organizando em três ciclos de formação humana: o 1º ciclo compreende o 1º, 2º e 3º anos; o 2º ciclo corresponde ao 4º, 5º e 6º anos; e no 3º ciclo estão os três anos finais do Ensino Fundamental, 7º, 8º e 9º. Cada ano escolar é composto por duas turmas. Os estudantes são moradores de diversos bairros da cidade de Belo Horizonte ou de cidades circunvizinhas. A cada ano letivo são abertas 50 vagas para que alunos do 1º ano sejam matriculados no Centro Pedagógico. Os responsáveis interessados fazem a inscrição dos filhos e os alunos a serem matriculados são decididos por meio de um sorteio. Sendo assim, o CP recebe um público de alunos diversificado nos aspectos socioeconômicos e culturais.

Além das disciplinas regulares do CP, como arte, ciências, educação física, geografia, história, língua portuguesa, matemática, inglês e espanhol, os alunos escolhem três disciplinas por semestre sobre assuntos variados. Estas disciplinas são chamadas de Grupo de Trabalho Diferenciado (GTD). Os GTDs são ministrados semanalmente e cada aula tem duração de 1 hora e 20 minutos. As turmas nos GTDs são menores,

ocorrendo uma enturmação diferente das turmas das disciplinas regulares com alunos de diferentes turmas do mesmo ciclo de formação. A cada semestre, GTDs de diferentes temas são oferecidos pelos professores para os estudantes (e.g., atividades de teatro, música, artes integradas, contos de fadas, oficinas criativas, esportes). Dos GTDs oferecidos, os alunos podem escolher três que eles têm interesse em participar, indicando quais são a primeira, segunda e terceira opção. Assim, os professores do CP, com base nas opções escolhidas e vagas disponíveis, definem quais alunos participarão de determinado GTD.

No ano de 2019, nós oferecemos aos alunos do CP GTDs de Pensamento Computacional com o propósito de estimular o desenvolvimento das habilidades de Pensamento Computacional dos alunos por meio da programação. Conforme ilustrado na Figura 4.2, dois tipos de GTD de Pensamento Computacional foram oferecidos no ano de 2019: *GTD de Pensamento Computacional I* e *GTD de Pensamento Computacional II*.

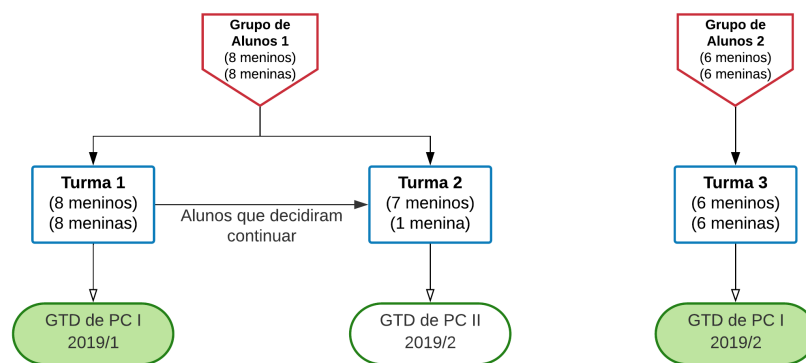


Figura 4.2: GTDs conduzidos no ano de 2019

O Estudo I aconteceu no primeiro semestre de 2019 (i.e., semestre 2019/1) com 16 alunos do 6º ano por meio do GTD de PC I 2019/1, sendo essa a nossa primeira experiência com o ensino do Pensamento Computacional. O Estudo II, conduzido por meio do GTD de PC I 2019/2, foi uma versão revisada do primeiro GTD conduzido no primeiro semestre de 2019. Ele foi conduzido no segundo semestre de 2019 com outros 12 alunos do 6º ano que tiveram interesse em participar porque não foram selecionados no primeiro semestre por falta de vagas ou porque ficaram curiosos com os comentários dos colegas que participaram da primeira turma. Por fim, o Estudo III, também conduzido no semestre 2019/2, foi uma continuação do Estudo I. Por isso, apenas os 16 alunos do primeiro grupo que participaram do primeiro estudo tiveram a oportunidade de participar deste terceiro estudo. Os 16 alunos precisaram escolher o GTD de PC II 2019/2, caso quisessem continuar participando. Desses, oito (i.e., 7 meninos e 1 menina) demonstraram interesse em continuar participando.

4.2.3 Coleta de Dados

A pesquisa qualitativa tem como uma de suas características a flexibilidade e aplicação de diferentes métodos e técnicas em um mesmo estudo. O processo de coleta de dados dos estudos conduzidos ocorreram em diferentes momentos e envolveram uma variedade de instrumentos e métodos de coleta. Durante as aulas, adotamos a observação participante e gravações de vídeos para coleta de acontecimentos de interesse da pesquisa. Questionários foram utilizados para obtenção de dados mais objetivos tais como perfil dos alunos e avaliação das atividades. Para a obtenção de dados mais qualitativos (e.g.: opiniões, justificativas, sentimentos), grupos focais e entrevistas foram realizados com os alunos. Em seguida apresentamos e detalhamos os instrumentos de coleta de dados utilizados e os dados coletados através deles.

Observação Participante

A observação participante é uma técnica de coleta de dados comumente utilizada em pesquisas etnográficas. O pesquisador conduz a observação, *in loco*, dos participantes imersos em um contexto social, fazendo registros sobre os seus comportamentos, comentários, postura, interações e atividades sendo conduzidas [Spradley, 1980]. Spradley [1980] define cinco tipos de participação que podem ser conduzidos por pesquisadores dependendo do grau de envolvimento que o pesquisador tem na condução das atividades no contexto da observação. Segundo ele, a participação pode ser inexistente, passiva, moderada, ativa ou completa.

Nos estudos conduzidos com os alunos do Centro Pedagógico assumimos o papel de participantes ativos uma vez que, além do papel pesquisadores, também participamos diretamente das aulas como professores ao planejar as atividades e ensinar os alunos. A interação com os alunos aconteceu antes e durante a condução das atividades em sala. A interação com os alunos antes da condução das atividades ocorreu em momentos de explicações de conceitos e das atividades que seriam realizadas. Já durante a condução das atividades em si, a interação ocorreu em momentos de auxílio, quando eles tinham dúvidas e necessitavam de ajuda. A observação ocorreu tanto em momentos ociosos, ou seja, quando os alunos não precisavam de nenhum tipo de auxílio, mas também durante os momentos de interação a fim de entender as dificuldades que eles estavam enfrentando e o porquê deles terem estas dificuldades.

Pesquisadores conduzindo observações participantes podem registrar o que está sendo observado simultaneamente à observação ou após eles terem deixado o local de estudo [Spradley, 1980]. Além disso, Spradley [1980] afirma que os registros devem ser objetivos e subjetivos, contendo informações precisas do que acontece e reflexões,

interpretações que o observador tem sobre tais acontecimentos. Neste sentido, os registros das observações das aulas conduzidas no Centro Pedagógico ocorreram em dois momentos. Durante as aulas e atividades o registro foi feito por meio de diários de campo, que são qualquer espaço (e.g., bloco de notas) físico ou digital onde são feitas anotações sobre o que está sendo observado [Leitão & Prates, 2017]. Após a finalização das aulas, o registro foi feito digitalmente para expandir e detalhar os registros dos diários de campo. Dessa forma, os registros durante as aulas foram mais objetivos; enquanto aqueles realizados após as aulas, mais subjetivos, explorando e interpretando o que havia sido observado.

Questionários

Questionários são comumente utilizados por pesquisadores interessados em obter um grande volume de dados em curto período de tempo [Barbosa & da Silva, 2010]. Embora não estivéssemos lidando com um grande número de participantes, nós utilizamos questionários em nossos estudos devido a necessidade de obtenção de dados pontuais de forma simples e rápida durante as aulas dos GTDs.

Criamos dois tipos de questionários para serem utilizados no estudo: Questionário de Perfil e Questionário de Avaliação de Atividade. O Questionário de Perfil foi criado para coleta de dados demográficos e sobre a experiência prévia dos alunos com computadores e dispositivos móveis. O Questionário de Avaliação de Atividade foi criado para que os alunos fizessem avaliações imediatas das atividades realizadas por eles. Por avaliação imediata, nos referimos a realização da avaliação logo após o fim da atividade ter sido conduzida nas aulas. Adotamos essa abordagem em vez de conduzirmos uma avaliação de todas as atividades ao fim do semestre para que as avaliações dos alunos fossem mais precisas e estivessem mais condizentes com a impressão real que eles tiveram destas atividades. Uma única avaliação no fim do semestre de todas as atividades conduzidas poderia ser menos precisa dado que os alunos precisariam se lembrar de experiências que ocorreram há muito tempo, influenciando as suas respostas positiva ou negativamente.

Grupos Focais

Grupo focal ou grupos de foco é um método de coleta de dados que permite a análise de aspectos tais como concordância e discordância entre os participantes do grupo ao promover um ambiente de discussão interativa mediada [Leitão & Prates, 2017].

Grupos focais semiestruturados foram realizados ao fim de ambos os GTDs de PC I para coleta de dados qualitativos referentes à opinião dos alunos sobre as au-

las, atividades, ferramentas e o GTD em geral. Apesar dos questionários de avaliação das atividades prover dados relevantes sobre as atividades, eles são superficiais, dificultando a identificação das razões que levaram os alunos a avaliarem as ferramentas de determinada maneira. Ademais, entrevistas individuais podem ser cansativas para as crianças, que acabam respondendo de uma forma mais resumida e rápida [Neves, 2013]. Logo, adotamos grupos focais com o intuito de estimular os alunos a falar mais ao envolvê-los em situações de diálogo com outras crianças.

Os registros dos grupos focais foram feitos em áudio e vídeo. A gravação de vídeo foi utilizada para capturar a linguagem corporal dos participantes e identificar os participantes que estavam falando em determinado momento, o que facilita o processo de transcrição do que foi discutido, especialmente quando vários participantes falam ao mesmo tempo. Além disso, logo após a realização dos grupos focais, nós também registramos as nossas impressões iniciais sobre os grupos e pontos interessantes discutidos pelos participantes.

Entrevistas

Entrevistas podem ser estruturadas, livres, ou semiestruturadas. Em entrevistas estruturadas, o entrevistador segue exatamente o roteiro de perguntas previamente definido, mantendo a ordem das perguntas do roteiro e não aprofundando em tópicos discutidos [Barbosa & da Silva, 2010]. Por outro lado, entrevistas livres não são guiadas por roteiros pré-definidos. As entrevistas semiestruturadas possuem características de ambas entrevistas estruturadas e livres, sendo o tipo de entrevista mais empregado em pesquisas qualitativas realizadas na área da Computação Barbosa & da Silva [2010]. Elas são mais flexíveis no sentido que permitem que o entrevistador seja mais ou menos fiel ao roteiro elaborado. Por exemplo, participantes podem mencionar tópicos que são do interesse do pesquisador ou inesperados, levando o pesquisador a abordá-los mais profundamente.

Entrevistas semiestruturadas foram planejadas para serem realizadas com os alunos que participaram do GTD de PC I no semestre 2019/1. Eles foram os únicos que tiveram a oportunidade de participar do GTD de PC II no semestre 2019/2. Assim, dois tipos de entrevistas foram elaboradas por nós. A primeira entrevista foi realizada com alunos que não participaram do GTD de PC II para entendermos os motivos pelos quais eles não quiseram participar. A segunda foi realizada no fim do GTD de PC II com os alunos que decidiram continuar participando.

Capítulo 5

Geração de Sequência Didática para Introduzir a Computação e o Pensamento Computacional

Neste capítulo, apresentamos a geração da sequência didática à luz das Metáforas de Perspectivas Culturais (MPC) para que a Computação e o Pensamento Computacional sejam introduzidos gradativamente no Ensino Fundamental II. Primeiramente, apresentamos o resultado da classificação dos recursos didáticos sendo adotados no ensino da Computação no Brasil para o ensino da Computação e Pensamento Computacional (PC) na etapa do Ensino Fundamental. Em seguida, apresentamos os recursos didáticos selecionados para serem utilizados na sequência didática. E por fim, apresentamos a visão geral da sequência didática.

5.1 Classificação de Recursos Didáticos Comumente Empregados no Brasil

Como explicado no Capítulo 4, utilizamos as Metáforas de Perspectivas Culturais (MPC) como base para a classificação dos recursos didáticos ao considerarmos que o PC fosse uma cultura estrangeira a ser introduzida a alunos sem conhecimentos sobre a Computação e o PC. Os recursos didáticos classificados por nós foram aqueles citados nos artigos revistos no MSL de Santos et al. [2018] e no nosso próprio MSL [Oliveira et al., 2019; Oliveira, 2020]. A classificação dos recursos foi feita com base na descrição feita dos recursos e de como eles foram usados nos artigos, ou seja, nós não os analisamos diretamente. Para fazer a classificação, nós levamos em consideração o

conhecimento da área da Computação necessário para que os recursos didáticos pudessem ser utilizados pelos alunos, o idioma utilizado, a interface e a forma em que eles foram aplicados e descritos nos estudos pelos autores. O fato de um recurso didático ser desplugado ou digital não foi um fator relevante para classificarmos os recursos didáticos. Portanto, há recursos digitais associados às MPCs iniciais e também recursos desplugados associados às MPCs finais do continuum. Certos recursos didáticos foram utilizados nos estudos de maneiras distintas e, portanto, alguns dos recursos foram associados a duas MPCs diferentes. Em seguida, nós apresentamos para cada uma das MPC, os principais recursos didáticos associados a elas.

5.1.1 Viajante Doméstico

Em relação ao conhecimento sobre a Computação e desenvolvimento das habilidades de PC, as ferramentas e atividades associadas à metáfora **Viajante Doméstico** (VD) não requerem que os alunos tenham estes conhecimentos e habilidades para que eles possam utilizar estas ferramentas e conduzir as atividades. A Tabela 5.1 apresenta a descrição de alguns recursos didáticos associados ao VD adotados nos estudos analisados por nós.

Tabela 5.1: Recursos didáticos associados ao Viajante Doméstico

Recurso Didático	Descrição
Racha Cuca	Plataforma com diversos jogos e desafios de raciocínio lógico
Bloxorz	Jogo de raciocínio lógico cujo objetivo é movimentar um bloco corretamente para fazê-lo cair em um buraco
Sokodan	Jogo de raciocínio lógico cujo objetivo é movimentar caixas até os lugares indicados
Desafio com Palitos	Jogo em que palitos devem ser adicionados ou removidos para que determinadas formações sejam criadas. Por exemplo, adicionar 2 palitos de tal forma que a palavra “OITO” seja formada
Pacman	Também conhecido como “come-come”. O jogador deve movimentar o personagem para comer todas bolinhas enquanto escapa dos fantasmas
Snake II	O clássico jogo da cobrinha. O jogador deve fazê-la crescer ao coletar comida, evitando que a cobra colida com o seu próprio corpo ou paredes
Space Invaders	O jogador controla espaço-nave para destruir naves inimigas que aparecem no cenário
Pong 2	Jogo de tênis simplificado onde jogadores devem movimentar uma raquete para evitar que a bola passe por elas
Kahoot!	Plataforma para criação e realização de quizzes gamificados

Jogos e desafios de raciocínio lógico são comumente aplicados nos estudos conduzidos no Brasil. Alguns exemplos são desafios do site RachaCuca¹ (e.g.: Missionários

¹RachaCuca 


e Canibais, O Lobo e a Ovelha), BloxorZ², Sokoban e Desafios com Palitos³. Jogos clássicos eletrônicos (e.i.: Pacman, Snake II, Space Invaders e Pong 2) foram utilizados no trabalho de Cândido et al. [2017] como temas para realização de simulações corporais, ou seja, a execução dos jogos foi simulada pelos próprios alunos assumindo o papel dos objetos de cada jogo como forma de representar e abstrair conceitos de funções. Além disso, o Kahoot! foi utilizado em um trabalho para realização de quizzes com questões de raciocínio lógico [de Oliveira et al., 2017]. Em nenhuma das atividades citadas tópicos da Computação são introduzidos ou abordados explicitamente e os alunos conseguem entender os fundamentos das atividades e realizá-las.


5.1.2 Observador à Distância

O **Observador à Distância** (OD) foi associado a recursos didáticos abordando atividades familiares aos alunos que dão a oportunidade para que eles “enxerguem” conceitos da cultura estrangeira sendo praticadas por meio destas atividades. Por exemplo, uma atividade em que os alunos, ao ter aprendido sobre algoritmos, devem decompor uma tarefa que envolve múltiplos passos (e.g.: escovar os dentes) em ações pequenas para que eles enxerguem estes passos como um algoritmo. Vale destacar, que essa mesma atividade poderia estar associada ao **Viajante Doméstico** caso os alunos a realizassem sem ter o conhecimento dos algoritmos. Neste caso, a cultura estrangeira estaria “invisível” aos alunos embora eles estejam em contato com ela. O OD também foi associado a explicações teóricas ou demonstrações de exemplos, uma vez que o foco dessa MPC é a apresentação da cultura estrangeira como informação. A apresentação teórica de conceitos, que é geralmente feita antes da condução de atividades práticas abordando os conceitos, é um exemplo comum de utilização dessa MPC nos estudos lidos por nós. Outro exemplo de atividade associada ao OD foi a apresentação de peças teatrais para demonstrar conteúdos da Computação a alunos. Os recursos didáticos classificados como OD estão descritos na Tabela 5.2.

5.1.3 Visitante em Tour Guiado

Recursos didáticos associados à metáfora **Visitante em Tour Guiado** (VTG) são apresentados na tabela 5.3. Dentre eles, o Code.org⁴ e o livro *Computer Science Un-*

²BloxorZ 

³Desafios com Palitos 


⁴Code.org 

Tabela 5.2: Recursos didáticos associados ao Observador à Distância

Recurso Didático	Descrição
Peças teatrais	Peças de teatro para visualmente apresentar os conceitos de detecção de erros, criptografia, números binários para alunos
Classes	Explicações teóricas sobre conceitos e tópicos da Computação e do PC tal como algoritmos, decomposição, loops, e condicionais
Criação de Algoritmos	Os alunos devem criar algoritmos para situações do cotidiano: cozinhar um bolo, tomar banho, montar um barco de papel
Dinâmicas	Dinâmicas para trabalhar conceitos da Computação e PC. Ex.: Caminho do Robô, dinâmica em que os alunos devem indicar a sequência numérica para fazer um robô (simulado por um aluno) percorrer determinado caminho. Cada número representa uma movimentação diferente.

*plugged*⁵ foram os recursos mais utilizados nos trabalhos realizados no Brasil com alunos do Ensino Fundamental.

Tabela 5.3: Recursos didáticos associados ao Visitante em Tour Guiado

Recurso Didático	Descrição
Code.org	Plataforma com diversas lições para o ensino-aprendizado da Computação
CS Unplugged	Livro gratuito que contém diversas atividades desplugadas com o intuito de ensinar a Computação sem a utilização de dispositivos tecnológicos
LightBot	Jogo em que comandos devem ser utilizados para criar algoritmos para fazer um robô se movimentar pelo cenário e acender os campos demarcados
Hour of Code	Plataforma com atividades de aproximadamente 1 hora de duração para disseminação da Computação
Blocky Games	Site com jogos e atividades para o ensino da Computação através da linguagem de programação de blocos
Monster Coding	Plataforma com atividades para o aprendizado de lógica de programação e matemática
Flippy Bit	Jogo educacional para o ensino de número binários e hexadecimais
SoccerCraft	Atividade desplugada onde os alunos devem utilizar comandos para levar um jogador de futebol com a bola até o gol evitando jogadores do time adversário
The Foos	Jogo similar ao LightBot em que o jogador deve utilizar comandos para controlar um personagem que deve realizar pequenas tarefas

A plataforma **Code.org** oferece a professores e estudantes cursos com lições e atividades variadas para o ensino e aprendizagem da Computação. Os estudos analisados por nós que empregaram o Code.org exploraram atividades da plataforma com níveis de complexidade similares no que diz respeito à exposição à Computação e ao PC. A maioria deles utilizaram o Curso 2⁶, que possui lições desplugadas e lições plugadas abordando tópicos específicos tais como *sequenciamento de ações*, *laços de repetição* e

⁵CS Unplugged [!\[\]\(011c28b8f907ee8625ff950900b30c29_img.jpg\)](#)

⁶Curso 2 [!\[\]\(5a56aac5cf3ed5f6d309bbbb67f2d8c6_img.jpg\)](#)

condicionais. Nestas lições os alunos devem utilizar uma linguagem de programação em blocos para resolver desafios. Os alunos têm a sua disposição um conjunto limitado de blocos, cada um representando uma ação que um personagem pode executar, e estes blocos devem ser utilizados para criar programas que solucionarão os desafios (e.g., fazer o personagem chegar a um determinado lugar). Estas lições permitem que os alunos se aproximem mais da cultura estrangeira por meio da programação em blocos; porém, essa aproximação é superficial e limitada dado que tanto os comandos que os alunos podem utilizar quanto as coisas que eles podem fazer são muito específicas. A liberdade de criação não está presente e a solução para os problemas é “guiada” por meio dos comandos específicos disponibilizados aos alunos. Devido a estas limitações impostas por esses cursos, eles também foram associados à MPC VTG.

O livro **Computer Science Unplugged** também foi amplamente adotado nos estudos. Embora o livro contenha apenas atividades desplugadas, ele também foi associado ao VTG, pois suas atividades abordam tópicos específicos da Computação (e.g.: números binários, algoritmos de ordenação, representação de imagens, etc.), os quais não fazem parte do “mundo” dos alunos, e permitem que eles utilizem diretamente estes conhecimentos na prática. A atividade “Contando os Pontos”, que tem como objetivo ensinar como números são representados apenas através de 0s e 1s, foi a mais utilizada nos trabalhos que adotaram o Livro. Nesta atividade, os alunos manipulam cartões que possuem pontos em um lado e nada do outro. Assim como ilustrado na Figura 5.1, o lado sem nada representa os valores 0, o lado com os pontos representa os valores 1 e a quantidade de pontos representa o valor decimal equivalente a determinada posição. Dessa forma, os alunos devem simplesmente contar a quantidade de pontos para que eles façam a conversão de binário para decimal.

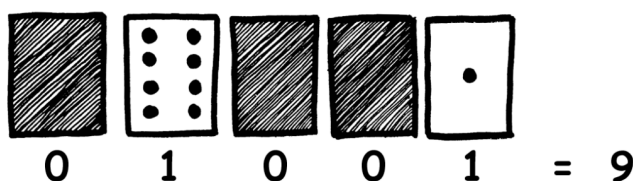


Figura 5.1: Contando os Pontos: exemplo de representação do número 9

5.1.4 Estrangeiro com Tradutor

A Tabela 5.4 apresenta os principais recursos didáticos associados ao **Estrangeiro com Tradutor** (EcT). A maioria dos estudos, em algum ponto, utilizaram ferramentas que adotam linguagens de programação visual (LPV) e que, conseqüentemente, foram associadas ao EcT. As linguagens de programação visual tornam o primeiro contato


com a programação mais simples para os alunos, fazendo que estas ferramentas sejam adequadas para os alunos do Ensino Fundamental embora elas sejam complexas e possam ser utilizadas de inúmeras formas. Dentre as ferramentas de linguagem visual, o Scratch, Arduino⁷ (programado através do Scratch), kit de robótica Lego Mindstorms⁸ e AppInventor⁹ foram as mais comumente adotadas pelos pesquisadores.


Tabela 5.4: Recursos didáticos associados ao Viajante Doméstico


Recurso Didático	Descrição
Scratch	Ferramenta para criação de animações, jogos, histórias utilizando linguagem de programação visual em blocos.
Arduino	Microcontrolador de baixo-custo com sensores e atuadores que pode ser utilizado de diferentes formas (programado através de LPVs)
Lego Mindstorms	Kit de robótica da Lego (programado através de LPVs)
App Inventor	Ambiente para a criação de aplicativos para dispositivos Android por meio de uma linguagem de programação em blocos
Kodu Game Lab	Ambiente de desenvolvimento de jogos 3D por meio de LPVs criado a fim de ensinar princípios básicos de programação para crianças
OBI	Exercícios da Olimpíada Brasileira de Informática solucionados por meio de LPVs


O Scratch foi o recurso didático com maior presença nos trabalhos analisados, sendo utilizado em 72 dos 165 estudos. O Scratch faz o uso da linguagem de programação visual Scratch Blocks¹⁰, desenvolvida a partir da LPV Blockly¹¹ da Google. Assim, como o Blockly, o Scratch Blocks é uma linguagem de programação visual em blocos, os quais devem ser encaixados uns aos outros como um quebra-cabeça para que *scripts* sejam desenvolvidos. Os blocos possuem diferentes formas, que indicam visualmente quais tipos de blocos podem ser combinados (Figura 5.2). Em nossa análise, consideramos a utilização de uma LPV análoga a tradução textual e dado que o Scratch permite que inúmeros tipos de projetos com diferentes níveis de complexidade sejam desenvolvidos, ele foi associado à MPC EcT.


Os outros recursos didáticos associados ao EcT também adotam a abordagem de traduzir a linguagem de programação textual para uma LPV e permitem que seus usuários programem e manipulem objetos de diferentes maneiras e níveis de complexidade. O Arduino pode ser programado por meio do *Scratch for Arduino* (S4A) e consequentemente foi associado ao EcT. Nos trabalhos em que o S4A foi adotado, os alunos tiveram a experiência de manipular e programar o Arduino por meio do Scratch

⁷Arduino 

⁸Lego Mindstorms 

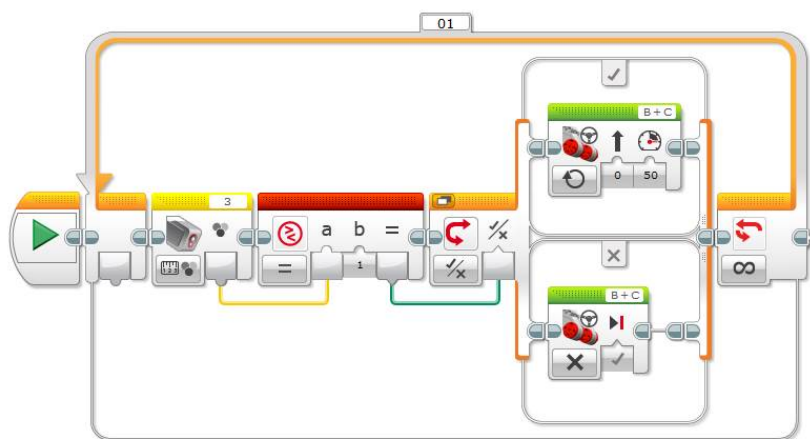
⁹AppInventor 

¹⁰Scratch Blocks 

¹¹Blockly 

**Figura 5.2:** Scratch Blocks

Blocks. Similarmente, o kit de robótica Lego Mindstorms pode ser programado utilizando o Scratch Blocks ou uma LPV própria da Lego (Figura 5.3).

**Figura 5.3:** LPV do Lego Mindstorms. Fonte: <https://www.funcodeforkids.com/lego-mindstorms-ev3-compare-block-in-detail/>

O AppInventor adota o Blockly da Google para que os seus usuários consigam desenvolver aplicativos funcionais para dispositivos Android, oferecendo uma experiência similar àquela que desenvolvedores de software normalmente teriam ao criar aplicativos. Por fim, o Kodu Game Lab adota uma LPV própria (Figura 5.4) para que jogos 3D sejam desenvolvidos.

A Olimpíada Brasileira de Informática (OBI)¹² é uma competição realizada similarmente às outras olimpíadas científicas brasileiras, sendo criada com o intuito de despertar o interesse de alunos pela Ciência da Computação. Exercícios da OBI foram utilizados em trabalhos para que alunos os resolvessem por meio de LPVs e, logo, nós também associamos a OBI (não exclusivamente) à metáfora EcT.

¹²OBI



Figura 5.4: Exemplo da LPV do Kodu Game Lab. Fonte: <http://www.kodugamelab.com/>

5.1.5 Estrangeiro sem Tradutor

No caso da metáfora **Estrangeiro sem Tradutor** (EsT), as ferramentas e atividades utilizadas pelos pesquisadores envolviam ambientes de desenvolvimento de software ou ferramentas educativas que adotavam linguagens de programação tradicionais tais como Python, C/C++, Swift e PHP. A Tabela 5.5 apresenta os principais recursos didáticos adotados nos estudos conduzidos no Brasil.

Tabela 5.5: Recursos didáticos associados ao Estrangeiro sem Tradutor

Recurso Didático	Descrição
Arduino	Microcontrolador de baixo-custo com sensores e atuadores que pode ser utilizado de diferentes formas (programado através de linguagens de programação textual).
Lego Mindstorms	Kit de robótica da Lego (programado através de linguagens de programação textual)
Linguagens de Programação Tradicionais	Linguagens de programação tradicionais sendo utilizadas para o desenvolvimento de programas simples.
OBI	Exercícios da Olimpíada Brasileira de Informática solucionados por meio de linguagens de programação textual

Recursos didáticos tais como o Arduino, o kit de robótica Lego Mindstorms e exercícios da OBI também foram associados ao EsT, além de terem sido associados ao EcT. Eles foram utilizados e manipulados de diferentes maneiras nos estudos analisados e, conseqüentemente, eles foram associadas a ambas as metáforas. Ambos o Arduino e o Lego Mindstorms podem ser programados por meio de LPVs ou linguagens de programação textuais. O Arduino pode ser programado utilizando C/C++ e também através do Scratch Blocks e o kit Lego MindStorms pode ser programado através do Scratch Blocks, sua linguagem de programação visual própria e também linguagens de programação textual como o Python. Similarmente, houve trabalhos que adotaram exercícios da OBI a serem resolvidos utilizando linguagens de programação textuais. Portanto, a utilização de LPVs para manipular estes recursos foi associada ao EcT e a utilização de linguagens de programação textuais foi associada ao EsT.

Linguagens de programação textual também foram adotadas em diversos dos trabalhos analisados. A linguagem de programação educacional Logo¹³ foi uma das linguagens de programação textuais mais empregadas nos trabalhos, sendo utilizada para a programação da movimentação de uma tartaruga de tal forma que ela desenhe figuras geométricas, fractais, ou outras coisas na tela. O K-Turtle¹⁴ e o Super Logo¹⁵ foram algumas das aplicações utilizadas para que os alunos programassem em Logo. Além do Logo, houve trabalhos utilizando Pascal, HTML 5 e CSS 3 por meio da ferramenta Construct 2¹⁶ e Python por meio do programa SyperPython¹⁷.

Na próxima seção, nós apresentamos os recursos didáticos que selecionamos, a partir da classificação apresentada previamente, para serem aplicados na pesquisa-ação conduzida com os alunos do Ensino Fundamental II.

5.2 Ferramentas e Materiais Seleccionadas para Aplicação em Salas de Aula

Os recursos didáticos para serem incluídos na sequência didática foram selecionados a partir dos recursos identificados e classificados na Seção 5.1 e de outros recursos que nós procuramos diretamente. A Figura 5.5 apresenta os recursos selecionados e suas classificações no continuum das MPC.

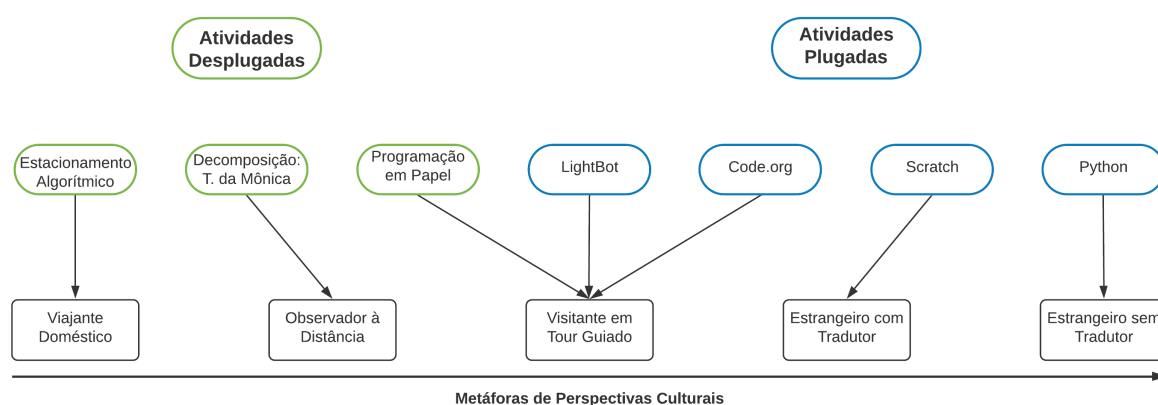


Figura 5.5: Ferramentas e atividades seleccionadas

¹³Logo [↗](#)

¹⁴K-turtle [↗](#)

¹⁵Super Logo [↗](#)

¹⁶Construct 2 [↗](#)

¹⁷SuperPython [↗](#)

Estacionamento Algorítmico

Estacionamento Algorítmico¹⁸ é um jogo desplugado cujo objetivo é retirar o carro vermelho do estacionamento sem que ele bata nos demais veículos. Os veículos podem ser movidos apenas na orientação que eles estão, ou seja, veículos na vertical podem ser movidos para cima ou para baixo e veículos na horizontal, para esquerda ou para direita. A Figura 5.6 ilustra o jogo e alguns de seus desafios:



Figura 5.6: Desafios 1 e 2 do Estacionamento Algorítmico

Ele não foi utilizado nos artigos analisados por nós; contudo, o selecionamos para o nosso trabalho, pois ele é uma atividade desplugada de lógica simples, que não requer que os alunos tenham conhecimentos sobre a Computação para que eles consigam resolver os desafios. Na atividade, os alunos também devem registrar os passos para resolver os desafios, porém mesmo essa tarefa pode ser feita sem que eles tenham consciência de que estão registrando algoritmos. Portanto, a atividade está associada ao Viajante Doméstico.

Decomposição da Turma da Mônica

A Decomposição da Turma da Mônica¹⁹ é uma atividade desplugada onde tarefas devem ser decompostas em passos menores e precisos a serem seguidos para que elas sejam completadas. Conforme ilustrado na Figura 5.7, as tarefas que devem ser decompostas pelos alunos são comuns e conhecidas por eles, ou seja, elas fazem parte da cultura nativa dos alunos, logo eles conseguem decompô-las corretamente mesmo que a níveis de detalhamento baixo.

Selecionamos essa atividade para ser aplicada após o conceito de algoritmos ser introduzido formalmente aos alunos para que eles o coloquem em prática por meio de uma atividade simples e familiar a eles. Embora a atividade em si seja familiar aos alunos, a cultura nativa não está sendo “escondida” porque eles estão cientes de

¹⁸Disponível em: <http://www.computacional.com.br/>

¹⁹Disponível em: <http://www.computacional.com.br/>



Figura 5.7: Exemplos de tarefas a serem decompostas

que estão criando algoritmos. Portanto, essa atividade está associada ao Observador à Distância.

Programando em Papel

O Programando em Papel²⁰, é a primeira atividade do Curso 2 da plataforma Code.org. A atividade consiste em utilizar comandos para desenhar figuras em um papel quadriculado (Figura 5.8). A estrela representada na figura indica a casa inicial que deve ser utilizada como referência para criar os algoritmos. No exemplo apresentado na figura abaixo, uma das possíveis soluções para o problema seria: *mover para baixo, colorir, mover para direita, mover para cima, colorir, mover para a direita, colorir, mover para a direita, mover para baixo, colorir*.

Criando Algoritmos I

Utilizando as instruções abaixo, crie algoritmos para desenhar as figuras abaixo.

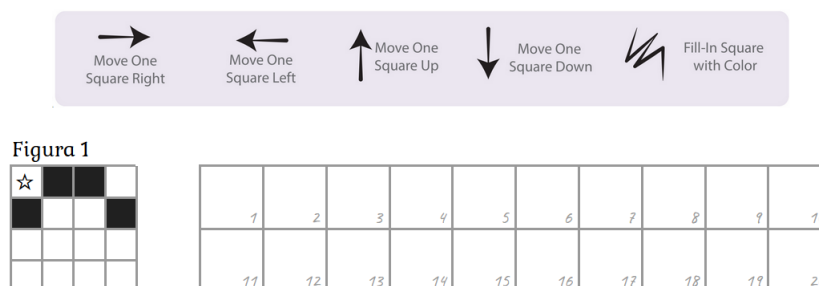


Figura 5.8: Versão modificada do Programando em Papel utilizada na Pesquisa

O Programando em Papel foi selecionado para que os alunos praticassem de forma consciente a criação e execução de algoritmos. Diferentemente da Decomposição da Turma da Mônica, o Programando em Papel não é algo familiar aos alunos do EF

²⁰Disponível em: <https://studio.code.org/s/course2/stage/1/puzzle/1>

II dado que eles devem criar e executar algoritmos para desenhar figuras em um papel quadriculado. Logo, essa atividade está associada ao Visitante em Tour Guiado.

LightBot

O LightBot²¹ é um jogo digital cujo propósito é ensinar lógica de programação. Os jogadores devem utilizar comandos (avancar, virar para direita, virar para a esquerda, pular, acender, etc.) para programar um robô que deve andar pelo cenário e acender todos os ladrilhos azuis. O jogo é composto por diversos desafios que ficam mais complexos conforme o jogador avança. Um dos desafios é ilustrado na Figura 5.9.

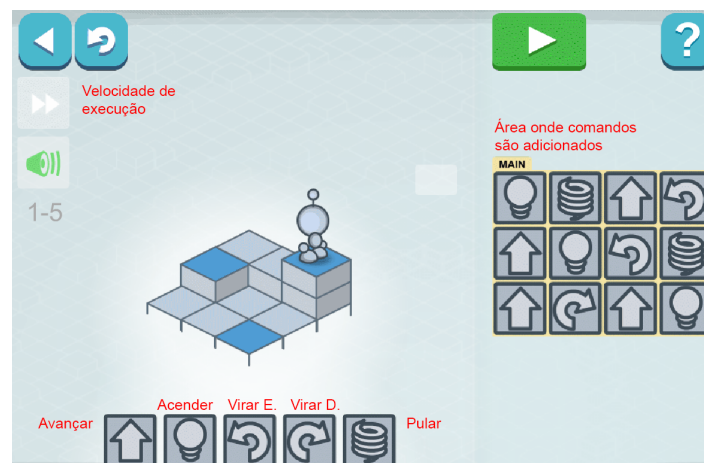


Figura 5.9: LightBot: exemplo de desafio e sua solução

O LightBot é pago e está disponível para Android e iOS. No entanto, os criadores disponibilizaram uma versão gratuita, porém limitada, que pode ser utilizada em navegadores como parte da iniciativa Hora do Código²².

Selecionamos a versão gratuita para ser utilizada no trabalho para que os alunos também praticassem a criação, execução e teste de algoritmos, porém de forma plugada e, possivelmente, mais engajadora já que ele pode ser visto como um jogo pelos alunos. Ele também está associado à MPC Visitante em Tour Guiado, visto que comandos similares aos comandos do Programando em Papel e pode abordar conceitos mais avançados de programação (e.g: repetição de ações, procedimentos e recursão).

²¹LightBot [↗](#)

²²LightBot - Code Hour [↗](#)

Plataforma Code.org

Criada em 2013, a Code.org²³ é uma organização sem fins lucrativos que busca promover o ensino da Ciência da Computação para crianças, jovens e mulheres e outros grupos com baixa representatividade na área de Computação.

Eles mantêm uma plataforma online com diversos cursos de diferentes níveis de complexidade para ensinar a Ciência da Computação. Dentre eles, quatro cursos disponíveis buscam introduzir os principais fundamentos da Ciência da Computação. O Curso 1²⁴ é um curso introdutório para alunos entre 4 e 6 anos de idade, que ainda estão começando a aprender a ler. O Curso 2²⁵, outro curso introdutório, é para alunos com mais de 6 anos de idade e que devem saber ler. O Curso 3²⁶ e Curso 4²⁷ são continuações do Curso 2 e Curso 3 respectivamente.


Consoante com o que foi apresentado na Seção 5.1, a maioria dos trabalhos no Brasil adotaram o Curso 2, que está associado ao Visitante em Tour Guiado, para introduzir a Computação no Ensino Fundamental. Similarmente, este também foi o curso que decidimos utilizar em nosso trabalho para apresentação e prática do sequenciamento de ações, condicionais e outros conceitos da Computação. A utilização da linguagem de programação em blocos nas lições do Curso 2 também facilita a transição para ferramentas associadas à metáfora seguinte, as quais também utilizam LPVs.


Scratch


O Scratch²⁸ é uma ferramenta desenvolvida e mantida pelo Lifelong Kindergarten Group no Media Lab do Instituto de Tecnologia de Massachusetts. Ele foi desenvolvido para tornar mais simples, lúdica e atrativa a experiência de programar animações, histórias, jogos e simulações por meio de uma LPV [Resnick et al., 2009].


A Figura 5.10 apresenta a interface da versão mais recente do Scratch. Diversos tipos de blocos de comandos estão disponíveis no Scratch e cada um deles é agrupado por cores de acordo com as suas funções. Por exemplo, os blocos azuis são comandos relacionados à movimentação de personagens nos projetos. Já os blocos amarelos são responsáveis pelo tratamento de eventos tais como a iniciação da execução do projeto, acionamento de teclas do teclado ou cliques do mouse, etc. As estruturas de repetição e condicionais estão agrupadas nos blocos laranjas, chamados blocos de controle.


²³Code.org 

²⁴Curso 1 

²⁵Curso 2 

²⁶Curso 3 

²⁷Curso 4 

²⁸Scratch 

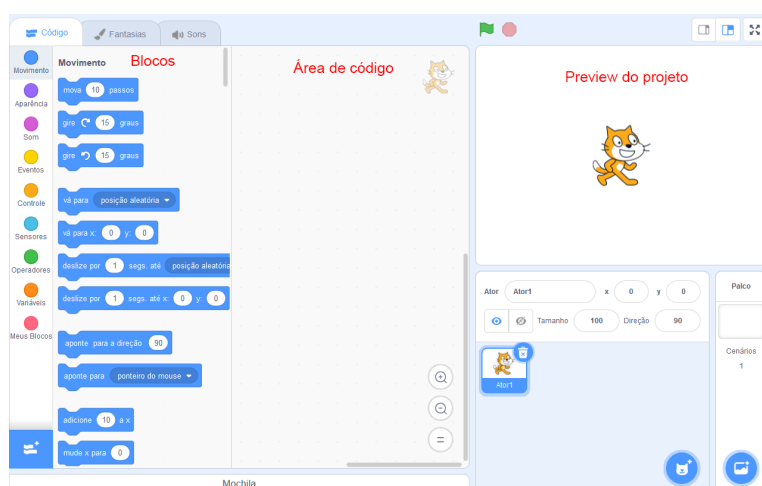


Figura 5.10: Interface do Scratch

Operações matemáticas e booleanas fazem parte dos blocos verdes (i.e.: blocos de operação). Utilizando e agrupando estes e outros blocos, os usuários podem criar *scripts* e programar personagens, cenários e assim criar vários tipos de projetos.

O Scratch é a ferramenta mais utilizada para introduzir a Computação, programação e pensamento computacional no Brasil [Santos et al., 2018; Blatt et al., 2017]. Além disso, dentre os recursos didáticos gratuitos associados ao Estrangeiro com Tradutor, o Scratch foi aquele que consideramos ser mais simples de ser utilizado e entendido por alunos sem experiência com programação. Portanto, nós o selecionamos para ser aplicado nos cursos para que os alunos pudessem criar diversos tipos de projetos, explorando diferentes conceitos da Computação.

Python

Python²⁹ é uma linguagem de programação textual bastante utilizada acadêmica e profissionalmente. Uma das características dessa linguagem é a simplicidade de sua sintaxe, tornando-o mais simples de ser entendida e, consequentemente, mais acessível para pessoas interessadas em aprender programação.

Assim como as outras linguagens de programação textuais sendo empregadas na introdução da Computação no Ensino Fundamental brasileiro, Python também está associada ao Estrangeiro sem Tradutor. A sintaxe da linguagem é simples quando comparada às demais linguagens de programação textuais. Além disso, ela é uma linguagem com diversas utilizações práticas, inclusive para pessoas que não são da área da Computação. Portanto, decidimos selecioná-la para ser utilizada após o Scratch.

²⁹Python

Kahoot!

Além das ferramentas selecionadas com base nas MPCs, também decidimos utilizar o Kahoot! para os alunos revisarem ludicamente conceitos e atividades realizadas por eles. O Kahoot!³⁰ é uma plataforma para criação e realização de *quizzes* gamificados (i.e.: kahoots), com perguntas de múltipla escolha. É necessário que haja um meio em que todos alunos possam visualizar a tela do dispositivo principal responsável por executar o Kahoot! (e.g.: projetor de vídeo), pois é nele que as perguntas e alternativas são exibidas. Já os dispositivos que os participantes utilizam, exibem as alternativas das perguntas para eles selecionarem. No próximo capítulo detalhamos os três estudos onde os recursos apresentados foram aplicados.


5.3 Proposta Inicial da Sequência Didática

Geramos a proposta inicial da nossa sequência didática sem definirmos especificamente as atividades, levando em consideração os recursos didáticos selecionados para serem utilizadas em cada etapa e os tipos de atividades a serem realizadas nestes recursos. A Tabela 5.6 apresenta a visão geral da proposta inicial da sequência didática gerada à luz das MPCs, mostrando a relação das metáforas, recursos didáticos utilizados e conteúdos abordados.

Tabela 5.6: Visão geral da sequência didática

Metáforas	Recursos Didáticos	Conteúdo
Viajante Doméstico	Estacionamento Algorítmico	Raciocínio Lógico
Observador à Distância	Explicações orais e Decomposição da Turma da Mônica	Algoritmos e sequência de passos
Visitante em Tour Guiado	Programando em Papel, LightBot e Code.org	Algoritmos, Loops, Condicionais e Depuração
Estrangeiro com Tradutor	Scratch	Criação de jogos, animações e histórias
Estrangeiro sem Tradutor	Linguagem de Programação Python	Criação de programas simples de entrada e saída

Em relação ao escopo da sequência didática no que diz respeito aos conceitos teóricos da Computação que queríamos que os alunos aprendessem. A nossa ideia foi a apresentação de conteúdos introdutórios da programação que os alunos deveriam saber para que eles conseguissem pelo menos criar projetos funcionais no Scratch na

³⁰Kahoot! 

quarta etapa da sequência. Para que projetos interessantes tais como jogos, animações e histórias fossem criados no Scratch, seria preciso que eles manipulassem pelo menos os principais blocos de evento e controle na ferramenta. Dessa forma, além do conceito de algoritmo, que pode ser trabalhado em todos os recursos, decidimos que deveríamos apresentar aos alunos os conceitos e finalidades dos laços de repetição e estruturas condicionais. Para abordar estes tópicos em algoritmos mais avançados, optamos em utilizar o Code.org, que possui lições específicas abordando estes conteúdos.

Embora tenhamos definido quais seriam os recursos didáticos e conceitos teóricos abordados na sequência, a sequência didática completa, ou seja, as atividades específicas que seriam conduzidas e como os recursos seriam utilizadas foi gerada paralelamente à condução dos cursos ministrados na pesquisa-ação. A definição específica das aulas e atividades foi realizada levando em consideração como foi a condução da aula anterior. Nós buscamos observar se os alunos estavam conseguindo entender os conceitos apresentados, se estavam conseguindo realizar as atividades propostas, o quão difícil estava sendo para que eles as realizassem e se eles estavam gostando ou não das atividades. Assim, a realização da pesquisa-ação com os GTDs nos permitiu instanciar a proposta, discriminando as atividades e conteúdos específicos a serem abordados em cada aula, conforme será apresentado no Capítulo 6.

Capítulo 6

A Pesquisa-Ação

Neste capítulo detalhamos a Pesquisa-Ação que foi realizada por meio de três estudos em salas de aula no Centro Pedagógico da UFMG (CP). Antes de conduzi-la, a pesquisa foi submetida ao Comitê de Ética em Pesquisa, que a aprovou para ser realizada (CAAE: 09959219.9.0000.5149). Primeiramente, apresentamos uma visão geral dos cursos ministrados. Em seguida, descrevemos o perfil dos alunos participantes. E por fim, para cada estudo da pesquisa, apresentamos a descrição das aulas, seus resultados e discussões e lições aprendidas com cada um deles.

6.1 GTDs de Pensamento Computacional

Conforme explicado na Seção 4.2.2, a Pesquisa-Ação foi conduzida na disciplina Grupo de Trabalho Diferenciado (GTD) oferecido no Centro Pedagógico aos seus alunos. Dois tipos de GTDs foram conduzidos: GTD de Pensamento Computacional I e GTD de Pensamento Computacional II.

O GTD de Pensamento Computacional I introduz o Pensamento Computacional e conceitos de programação tais como o pensamento algorítmico, decomposição de problemas e teste e correção de soluções. O GTD de PC I foi oferecido tanto no primeiro, quanto no segundo semestre de 2019. Os recursos utilizados incluíam tanto atividades desplugadas quanto plugadas, conforme apresentado no Capítulo 5. Para as atividades plugadas, utilizamos jogos digitais e ambientes utilizando linguagens de programação visual (LPV) a fim de tornar a experiência mais lúdica e engajadora para os alunos. A ideia inicial era tentar percorrer todo o percurso das MPCs desde o Viajante Doméstico até o Estrangeiro sem Tradutor. No entanto, no decorrer do primeiro GTD (2019-1) percebemos que não seria possível dentro do número de aulas disponíveis e fomos até a quarta MPC (i.e.: Estrangeiro com Tradutor). A MPC

Observador à Distância esteve presente no curso por meio de explicações teóricas curtas feitas no início de algumas aulas e por meio da atividade de decomposição, a qual foi aplicada apenas no GTD de PC I 2019/2. Como a MPC de Estrangeiro sem Tradutor não foi abordada, as linguagens de programação textual não foram apresentadas aos alunos. A Figura 6.1 apresenta um panorama dos recursos didáticos utilizados com os alunos. O Observador à Distância está em amarelo, pois a atividade de decomposição só foi utilizada no GTD de PC I 2019/2.

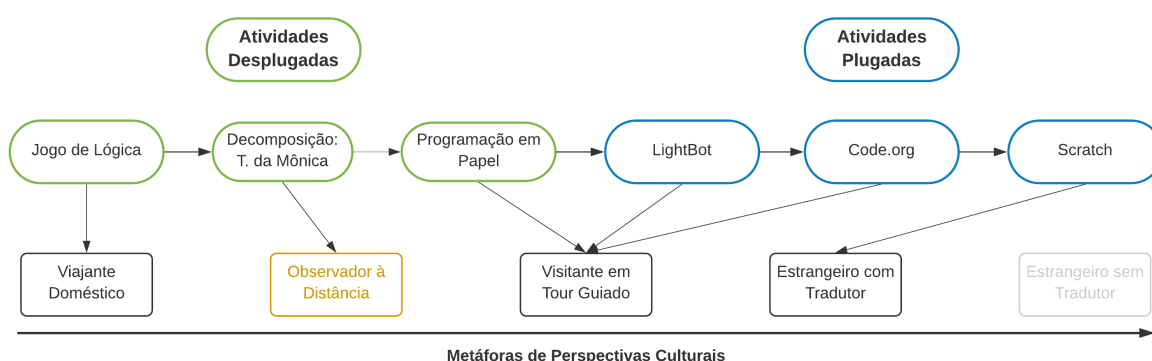


Figura 6.1: Recursos do GTD de Pensamento Computacional I

Como não foi possível chegar à MPC Estrangeiro sem Tradutor durante o GTD de PC I, o GTD de Pensamento Computacional II foi criado com o objetivo de ser uma continuação do GTD de PC I, requerendo que os participantes já tivessem conhecimento dos tópicos abordados no GTD de PC I. Logo, durante o processo de seleção dos participantes dos GTDs do CP, foi permitido que apenas os participantes do GTD de PC I do semestre 2019/1 pudessem escolher participar do GTD de PC II. Neste GTD, foi dada continuidade à utilização do Scratch com os alunos, abordando novas funcionalidades por meio de projetos mais complexos. Além da criação de projetos específicos e pré-definidos, também foi dada a oportunidade para que os alunos expressassem, definissem e criassem seus próprios projetos. No fim do GTD de PC II pretendíamos apresentar aos alunos uma linguagem de programação textual amigável como o Python. Ao fazer isso, a quinta MPC (i.e., Estrangeiro sem Tradutor) estaria presente e percorreríamos todo o continuum das MPC com o grupo de alunos que participaram desde o semestre 2019/1. No entanto, como explicaremos neste capítulo, não foi possível realizar esta última etapa com os alunos devido a imprevistos que ocorreram ao longo do semestre. Logo, no fim, utilizamos apenas o Scratch durante todo o GTD (Figura 6.2). Similarmente ao GTD de Pensamento Computacional I, a metáfora Observador à Distância esteve presente em algumas aulas do GTD de PC II ao introduzirmos conceitos (e.g.: plano cartesiano, variáveis, árvores de decisão) aos

alunos por meio de explicações teóricas.



Figura 6.2: Recursos do GTD de Pensamento Computacional II

6.2 Participantes

Um total de 27 alunos do CP participaram da pesquisa. O primeiro grupo de participantes, composto por 16 alunos, participou do primeiro GTD de PC I no semestre 2019/1 e, além disso, tiveram a oportunidade de continuar participando no GTD de PC II no semestre 2019/2. O Grupo 2 (12 alunos) participou apenas da versão revista do GTD de PC I no semestre 2019/2. No entanto, os responsáveis por um dos alunos não autorizaram a coleta de dados e, portanto, apenas 11 alunos participaram da pesquisa. No texto, os alunos são referidos utilizando os seguintes códigos:

- **Grupo 1:** G1P1 (Grupo 1 - Participante 1), G1P2, G1P3, ..., G1P16;
- **Grupo 2:** G2P1 (Grupo 2 - Participante 1), G2P2, G2P3, ..., G2P11;

O Questionário de Perfil (Apêndice B) foi aplicado nos estudos para traçar o perfil dos alunos por meio de perguntas demográficas básicas (e.g.: nome, data de nascimento) e suas experiências com a tecnologia. Ele foi disponibilizado aos alunos digitalmente através do Google Forms para que eles pudessem respondê-lo no próprio laboratório de informática. Os dados relativos às suas experiências com tecnologia envolveram a experiência com o computador, seja ele *desktop* ou móvel, e com *smartphones*. Para ambos os tipos de tecnologia, perguntamos aos alunos se eles as tinham em casa e com qual finalidade eles as utilizavam (e.g., entretenimento, estudo, pesquisas). O questionário possui mais perguntas específicas para os itens referentes aos computadores, pois as atividades do GTD seriam realizadas, em sua grande parte, por meio deles. Exclusivamente para os itens referentes aos computadores, foi questionado

se os alunos já o tinham utilizado previamente, onde eles normalmente o utilizam, a frequência que o utilizam e qual é a atividade que eles mais realizam no computador.

De acordo com os dados coletados, todos os 27 alunos participantes possuíam alguma experiência com a utilização do computador antes de iniciar o GTD, sendo que apenas dois deles não tinham computador em casa. A Figura 6.3 indica a frequência com que os participantes relataram utilizar o computador durante a semana. A grande maioria dos participantes (22 alunos) utiliza o computador menos de três dias por semana, sendo que 14 destes alunos apenas fazem o uso do computador nos fins de semana. Poucos alunos utilizam o computador todos os dias (3), de cinco a seis dias (1) e de três a quatro dias (1).

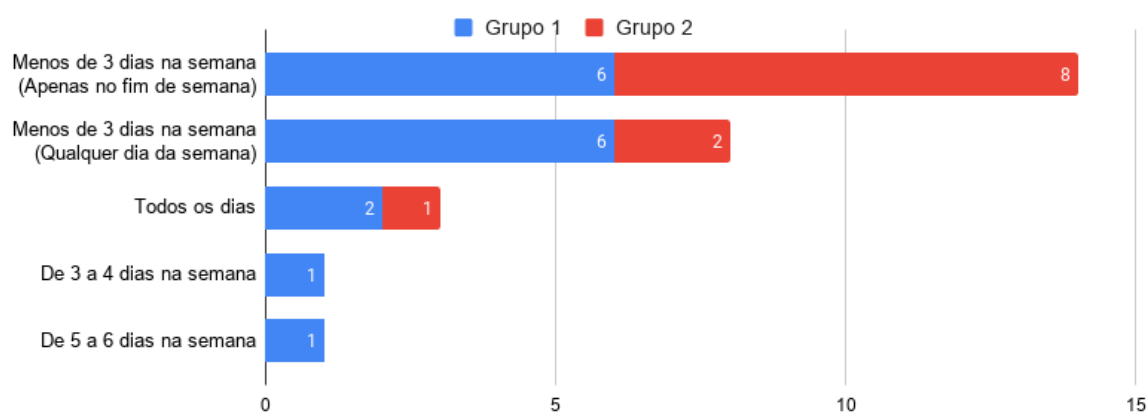


Figura 6.3: Frequência de uso do computador durante a semana

No que diz respeito às atividades que os alunos realizam no computador (Figura 6.4), fazer trabalhos de escola (8 alunos), assistir vídeos (7 alunos), e jogar (7 alunos) são as atividades mais comuns. Apenas três dos alunos utilizam o computador para estudar e dois deles para as atividades do GTD.

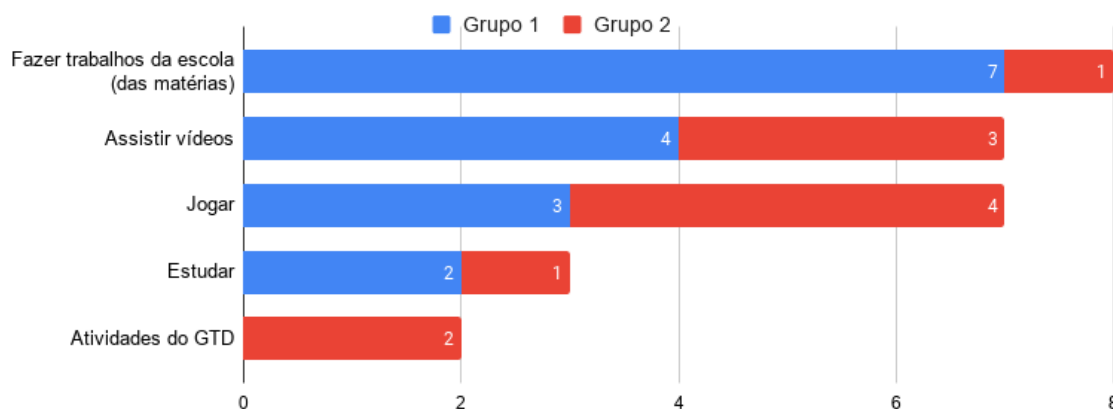


Figura 6.4: Atividades mais feitas no computador

6.3 Estudo I

O primeiro estudo realizado por nós aconteceu no primeiro semestre de 2019 por meio do GTD de PC I 2019/1. Este GTD foi lecionado para 16 alunos do Ensino Fundamental II. O número de participantes foi definido pelo tamanho da sala e disponibilidade de computadores nos laboratórios. Em seguida, nós apresentamos a descrição das aulas conduzidas, os resultados da avaliação da experiência, tanto sob a perspectiva dos alunos, quanto da nossa e, finalmente, a discussão e lições aprendidas.

6.3.1 Descrição das Aulas

A Tabela 6.1 sumariza as 14 aulas conduzidas no GTD de PC I 2019/1, fazendo a relação das aulas, metáforas, ferramentas, conteúdos e apresentando brevemente o que foi realizado. O detalhamento de como as aulas foram conduzidas é apresentado logo após a tabela. As abreviações na segunda coluna representam as MPCs associadas a cada aula: Viajante Doméstico (VD), Visitante em Tour Guiado (VTG) e Estrangeiro com Tradutor (EcT). O Observador à Distância (OD) não esteve presente neste GTD por meio de atividades, apenas através das explicações teóricas.

Tabela 6.1: Aulas conduzidas no GTD de PC I 2019/1

<i>Aulas</i>	<i>Recursos</i>	<i>Conteúdo</i>	<i>Descrição</i>
Aula 1	Estacionamento Algorítmico [VD]	Raciocínio Lógico	Apresentação do GTD de PC I, professores e alunos. Prática do raciocínio lógico ao solucionar desafios do Estacionamento Algorítmico
Aula 2	<i>Programando em Papel</i> [VTG]	Algoritmos	Introdução ao conceito de algoritmos e criação de algoritmos para pintar figuras utilizando comandos de movimento e colorir
Aula 3	LightBot [VTG]	Algoritmos	Revisão dos conteúdos das aulas anteriores e resolução dos desafios do <i>LightBot</i>
Aula 4	Code.org [VTG]	Algoritmos	Resolução dos desafios da lição <i>Labirinto: Sequência</i> para prática do sequenciamento de ações e utilização da linguagem de programação em blocos
Aula 5	Code.org [VTG]	Loops	Introdução ao conceito de loops. Resolução dos desafios da lição <i>Labirinto: Ciclos</i> para prática da identificação de ações e grupos de ações que se repetem e a utilização de <i>loops</i>
Aula 6	Code.org [VTG]	Depuração	Introdução ao conceito de depuração. Resolução dos desafios da lição <i>Abelha: Depuração</i> para prática da identificação e correção de erros
Aula 7	Code.org [VTG]	Condicionais	Introdução ao conceito de condicionais. Resolução dos desafios da lição <i>Abelha: Condicionais</i> para prática de condicionais
Aula 8	Scratch [EcT]	Introdução ao Scratch e Anime seu Nome	Revisão de todo o conteúdo visto até o momento. Apresentação do Scratch e os seus principais blocos de movimento, aparência e eventos. Condução da atividade <i>Anime seu Nome</i> , onde os alunos têm que animar as letras de seus nomes
Aula 9	Scratch [EcT]	Anime seu Nome	Finalização do <i>Anime seu Nome</i> . Realização de votação com a turma para escolhermos os três melhores projetos segundo os alunos
Aula 10	Scratch [EcT]	Jogos ou Estórias	Criação de animações, estórias e jogos simples no Scratch

Aula 11	Scratch [EcT]	Bugs no Ciclo da Água	Depuração e correção de problemas em projetos com animações sobre o Ciclo da Água no Scratch
Aula 12	Scratch [EcT]	Alteração de um jogo	Teste e modificação visual de um jogo de clicar no Scratch. No jogo, o jogador deve clicar em um personagem que fica aparecendo na tela
Aula 13	Scratch [EcT]	Alteração de um jogo	Continuação da aula anterior. Adição de novas funcionalidades ao jogo
Aula 14	Scratch [EcT]	Teste de Projetos	Teste dos jogos da turma e realização de votação com a turma para escolhermos os jogos mais legais

Aula 1: Exercitando o Raciocínio Lógico

O **Estacionamento Algorítmico**¹ foi utilizado nesta aula. Ela foi estruturada em três partes: apresentação do jogo, resolução de desafios, e registro das soluções.

Primeiramente, fizemos uma apresentação geral do GTD de PC I e dos professores que o estariam ministrando. Em seguida, apresentamos o Estacionamento Algorítmico, suas regras e como ele seria realizado em sala de aula. Para isso, utilizamos o projetor de vídeo para apresentar um exemplo simples da atividade.

Em seguida, foi dado início à atividade. Ela foi feita em duplas, onde os alunos tiveram que trabalhar em equipe para resolver os cinco primeiros desafios do nível iniciante do Estacionamento Algorítmico.

Por fim, os alunos fizeram o registro em papel das soluções dos desafios resolvidos por eles. Foi pedido que cada aluno registrasse pelo menos uma solução conforme apresentado no slide de apresentação do jogo, informando os veículos na ordem em que eles devem ser movimentados para que o desafio seja resolvido. Pedimos que os alunos testassem as soluções, seguindo o passo a passo registrado por eles, e corrigissem erros se necessário.

Aula 2: Algoritmos e Sequenciamento de Ações Desplugado

Nesta aula, o conceito de algoritmos foi apresentado formalmente e praticado pelos alunos. Utilizamos a atividade **Programando em Papel** presente no Curso 2 da plataforma Code.org para a parte prática da aula.

No início da aula, apresentamos o conceito de algoritmo aos alunos, fazendo associação com as soluções dos desafios do **Estacionamento Algorítmico** que eles tiveram que registrar na aula anterior. Situações e tarefas da vida cotidiana dos alunos também foram discutidas e decompostas em ações menores e específicas.

¹Disponível em: <https://www.computacional.com.br/#atividades>

Em seguida, foi realizada a parte prática da aula por meio de uma versão adaptada do **Programando em Papel**². Utilizamos o projetor de vídeo e slides para realizar a explicação da atividade através de exemplos simples e animações. Nós fizemos algumas modificações no Programando em Papel para adicionarmos uma maior variedade de exercícios. A atividade foi dividida em duas etapas: *Criando Algoritmos* e *Criando Figuras*.

No *Criando Algoritmos*, os alunos criaram algoritmos para desenhar as figuras indicadas. As seis primeiras figuras tiveram que ser desenhadas utilizando apenas os comandos de *movimentação* e *colorir* presentes na atividade original. As seis últimas figuras tiveram que ser desenhadas utilizando os comandos originais e um comando de repetição introduzido por nós para que eles praticassem a repetição de ações.

No *Criando Figuras*, os alunos criaram suas próprias figuras e algoritmos que as desenham. Em seguida, os alunos fizeram duplas e trocaram os algoritmos que eles criaram para que suas duplas o executassem e tentassem colorir as figuras. Por fim, foi pedido que os eles verificassem se a figura criada era a mesma que a figura original e trabalhassem juntos para corrigir possíveis erros, sendo que estes erros poderiam estar na execução incorreta do algoritmo ou nos próprios algoritmos e figuras originais.

Aula 3: Algoritmos e Sequenciamento de Ações Plugado - LightBot

Nos minutos iniciais da aula, conduzimos uma revisão abordando o que foi apresentado nas duas primeiras aulas. Utilizamos o Kahoot! para conduzirmos uma revisão lúdica e potencialmente engajadora com os alunos. O kahoot³ criado por nós continha questões sobre algoritmos e também questões com problemas do Programando em Papel conduzido na aula anterior. Discussões foram feitas com os alunos conforme cada questão fosse respondida, buscando explicar o porquê das alternativas estarem corretas ou incorretas.

Durante o restante da aula, os alunos fizeram a resolução dos desafios do primeiro nível do LightBot. Foi deixado que os alunos que conseguissem finalizar os desafios do primeiro nível, tentassem solucionar também os desafios do segundo nível.

Aula 4: Algoritmos e Sequenciamento de Ações Plugado - Code.org

Nesta aula, os alunos praticaram a criação de algoritmos por meio de uma linguagem de programação em blocos. Eles solucionaram desafios da lição **Labirinto: Sequência**⁴

²Aula 2 - Programando em Papel Adaptado [!\[\]\(7294993fb1bbe2afbeb9b19d00db5332_img.jpg\)](#)

³Aula 3 - Kahoot: Algoritmos e Programando em Papel [!\[\]\(e598ffc4fb50c546c17b948a0c264bfa_img.jpg\)](#)

⁴Aula 4 - Labirinto: Sequência [!\[\]\(bdbe5c2a9369dcb25db0198c79fa539d_img.jpg\)](#)

do Curso 2 da plataforma Code.org.

O Kahoot! foi novamente utilizado no início da aula para revisão de conteúdos. O kahoot⁵ criado tinha questões com desafios do LightBot para que os alunos pudessem revisar o que foi feito na aula anterior de forma divertida.

Em seguida, apresentamos a plataforma Code.org e sua linguagem de programação em blocos aos alunos. Os usuários e senhas dos alunos na plataforma Code.org foram disponibilizados e demos início **Labirinto: Sequência**. Nesta lição, os alunos devem levar o passarinho do Angry Birds até determinado local utilizando os comandos “avançar”, “vire à esquerda” e “vire à direita”. Para os alunos que conseguiram finalizar todos os desafios, nós disponibilizamos a lição **Artista: Sequência**⁶. Os desafios da lição do Artista são mais complexos, requerendo conhecimento sobre ângulos, pixels e interpretação de texto.


Aula 5: Aprofundando em Algoritmos - Loops


Antes de introduzirmos o tópico principal da quinta aula, conduzimos uma revisão com os alunos para lembrá-los dos tópicos e atividades realizadas até o momento. O kahoot⁷ realizado tinha questões referentes a algoritmos e às atividades que os alunos fizeram nas aulas anteriores.


Em seguida, apresentamos formalmente os laços de repetição aos alunos frisando como eles podem ser utilizados no Code.org para reduzir o número de blocos utilizados para resolver os desafios. Durante a explicação, lembramos aos alunos da repetição de ações que eles utilizaram na Aula 2 para colorir as figuras do Programando em Papel.


Para a parte prática da aula, utilizamos a lição **Labirinto: Ciclos**⁸, também presente no Curso 2 da plataforma Code.org. Os desafios desta lição têm a temática dos jogos Angry Birds e Plants VS Zombies. A lição introduz o bloco de repetição e tem como o foco a resolução de problemas utilizando a repetição de ações de maneira eficaz, ou seja, a resolução dos problemas com o menor número de blocos possível. Quando os desafios são resolvidos a ferramenta informa ao usuário se o número de blocos utilizados foi o mínimo ou não.

Os níveis da lição ficam gradativamente mais complexos, dificultando a identificação dos padrões que estão se repetindo e, conseqüentemente, a resolução dos níveis com o menor número de blocos. Nos níveis iniciais da lição, os alunos apenas devem repetir um único tipo de ação. As lições finais são difíceis de serem resolvidas com o

⁵Aula 4 - Kahoot II: LightBot 

⁶Aula 4 - Artista: Sequência 

⁷Aula 5 - Kahoot de Revisão I 

⁸Aula 5 - Labirinto: Ciclos 

número mínimo de blocos, requerendo a identificação de padrões mais complexos (i.e.: repetição de um conjunto de ações).

Aula 6: Aprofundando em Algoritmos - Depuração


Neste aula, abordamos a depuração de código por meio da lição **Abelha: Depuração**⁹ também presente no Curso 2 da plataforma Code.org. Todos os níveis da lição possuem algoritmos já construídos. No entanto, os algoritmos estão errados e devem ser corrigidos pelos alunos de tal forma que a abelha colete o néctar de todas as flores presentes no cenário. Diversos tipos de erros estão presentes ao longo da lição: utilização errônea de comandos, utilização de menos comandos que o necessário e utilização de mais comandos que o necessário. Além de erros, também há desafios com soluções corretas, porém ineficientes, que devem ser alteradas para que a solução utilize o número mínimo de blocos.


No início da aula apresentamos o tópico da análise de programas formalmente aos alunos, explicando a importância da análise de problemas, proposta de soluções para estes problemas e o teste das soluções a fim de verificar se elas estão corretas ou não. A lição foi iniciada após esta explicação teórica. Vários alunos conseguiram terminar os desafios desta lição e similarmente ao que tínhamos feito na Aula 4, disponibilizamos a lição **Artista: Depuração** para que eles não ficassem ociosos durante o restante da aula e continuassem praticando a depuração.

Aula 7: Condicionais

A última aula conduzida na plataforma Code.org abordou as estruturas condicionais pela lição **Abelha: Condicionais**¹⁰. Nesta lição são utilizados todos os blocos e conceitos que os alunos aprenderam previamente, juntamente com o bloco “SE” que é introduzido na lição. Os alunos devem utilizá-lo para fazer com que a abelha colete o néctar das flores apenas se as flores tiverem néctar. Em todos os níveis da lição a condição do bloco não é alterada, sendo que a verificação é sempre “Se néctar = 1”. Além disso, em um dos níveis, é necessário a utilização de laços aninhados para resolver o problema com o número mínimo de blocos, que é um ponto negativo da lição, uma vez que os laços aninhados só são apresentados aos alunos na lição seguinte do Curso 2.

Iniciamos a aula com uma explicação teórica sobre as condicionais para que os alunos compreendessem como o bloco “SE” funciona e conseguissem utilizá-lo correta-

⁹Aula 6 - Abelha: Depuração 

¹⁰Aula 7 - Abelha - Condicionais 

mente na lição. Em seguida, conduzimos a “Dinâmica da Palma”, descrita abaixo, com a turma para exemplificarmos o funcionamento das condicionais e fixação do conteúdo.

Dinâmica da Palma: atividade em que os alunos devem bater palmas de acordo com o número indicado pelo professor. O professor deve explicar que os alunos devem bater 1 palma SE o número falado for menor que 5. E caso, o número seja maior que 5, eles devem bater palmas no número de vezes do número falado. O condutor da dinâmica deve alternar entre número menores e maiores que 5 de forma arbitrária e observar a reação dos alunos e corrigi-los caso necessário. Deixamos o número 5 para o final da dinâmica, visto que ele não é nem menor e nem maior que 5, sendo este um caso que não cai em nenhuma das condições ditas aos alunos no começo da dinâmica.

A maioria dos alunos bateu o número de palmas corretamente para os casos iniciais. Por outro lado, na parte final da lição onde a condição “ 5 é maior que 5” é exibida, a maioria deles bateram 5 palmas, poucos bateram uma palma e não houve alunos que não bateram palmas. Para finalizar a dinâmica, questionamos a turma se 5 palmas era realmente a resposta correta, se o número 5 é menor ou maior que 5 e sobre o que deve ser feito quando nenhuma das condições é satisfeita. Durante o restante da aula, os alunos ficaram resolvendo os desafios da lição **Abelha: Condicionais**.


Aula 8: Introdução ao Scratch e Anime seu Nome


O Scratch foi utilizado a partir da oitava aula para prática e aplicação de tudo que foi abordado previamente no desenvolvimento de animações, histórias e jogos simples. Na Aula 8, apresentamos a ferramenta aos alunos e iniciamos a atividade Anime Seu Nome.

Primeiramente, conduzimos um kahoot para que os alunos revisassem tudo que aprenderam. As questões do kahoot¹¹ de revisão abordavam todos os conceitos teóricos apresentados (i.e.: algoritmos, loops, depuração e condicionais) e também desafios e problemas das atividades e ferramentas utilizadas no decorrer do GTD.

Em seguida, apresentamos o Scratch e suas principais funcionalidades interativamente pelo projetor de vídeo, mostrando o funcionamento dos principais comandos de movimento, aparência, som e eventos que os alunos utilizariam nas aulas. Antes dos alunos acessarem o Scratch, foi explicado que faríamos uma votação para decidir quais eram os melhores projetos como forma de engajar e motivar os alunos a desenvolverem projetos interessantes.

Por fim, realizamos o **Anime seu Nome**¹², que é uma das atividades de introdu-

¹¹Aula 8 - Kahoot de Revisão II 

¹²Aula 8 - Anime seu Nome 

ção ao Scratch sugeridas no próprio site da ferramenta. O objetivo da atividade é que os alunos aprendam a utilizar blocos simples de movimento, aparência, sons, eventos e controle através da criação de animações simples para as letras de seus nomes. No site do Scratch estão disponíveis a descrição da atividade, materiais de apoio para o professor conduzi-la e materiais de apoio para o aluno contendo sugestões de animações que eles podem criar. Os links para os materiais de apoio dos alunos foram disponibilizados para a turma para que os alunos pudessem testar certas animações e se inspirar.

Aula 9: Anime seu Nome (Continuação)

O Anime Seu Nome não pôde ser finalizado na aula anterior devido à falta de tempo, então alocamos a Aula 9 para que os alunos pudessem finalizá-la e para que realizássemos a votação dos melhores projetos. Após os alunos terem finalizado seus projetos, foi feito o compartilhamento deles com a turma e a votação dos melhores projetos. Para facilitar o compartilhamento e a visualização, foi criado um Estúdio da turma para que os alunos colocassem os seus projetos. Assim, pudemos acessar todos os projetos e exibí-los para a turma através do projetor de vídeo. Cada aluno ficou responsável por apresentar e explicar seu projeto à turma.

Aula 10: Criação de Jogos E Estórias

Nesta aula foi dada maior liberdade aos alunos para que eles escolhessem, dentre um conjunto de opções, o projeto que eles queriam criar no Scratch: **Crie uma Estória**¹³, **Jogo da Coleta**¹⁴, **Jogo de Pular**¹⁵, **Pega-Pega**¹⁶, **Pong**¹⁷. Todas as cinco opções também fazem parte das atividades sugeridas no site do Scratch, possuindo materiais de apoio para o condutor da atividade e para os alunos.

Um link que direcionava para uma pasta do Google Drive contendo os materiais de apoio das atividades foi disponibilizado para os alunos. Deixamos que eles explorassem os materiais por alguns minutos para que eles escolhessem qual dos projetos eles queriam criar no Scratch. A maioria dos alunos escolheu a atividade de estória, que era a primeira da lista. Os materiais de apoio de cada atividade possuíam diversos passos a serem seguidos, então decidimos separar cada um destes passos em diferentes arquivos em PDF para evitar que os alunos pulassem passos e que eles conseguissem segui-los sem grandes problemas.

¹³Aula 10 - Crie uma Estória [!\[\]\(b77aed42373877ea86d1cd8a66b06ee3_img.jpg\)](#)

¹⁴Aula 10 - Jogo da Coleta [!\[\]\(ec83030e89bfa4cb4c49248d2247e840_img.jpg\)](#)

¹⁵Aula 10 - Jogo de Pular [!\[\]\(c0509b681fe1d84fa177f5cb9a057e9b_img.jpg\)](#)

¹⁶Aula 10 - Pega-Pega [!\[\]\(78777943e070e904a742ec48e0e7f654_img.jpg\)](#)

¹⁷Aula 10 - Pong [!\[\]\(c3d6552f98f054395e95d5fd0da04e8a_img.jpg\)](#)

Aula 11: Bugs no Ciclo da Água

Devido a uma outra atividade ocorrendo na escola, a Aula 11 teve a duração de apenas 30 minutos, assim utilizamos a atividade **Bugs no Ciclo da Água**¹⁸ para que eles pudessem praticar a análise e correção de problemas através da depuração de algoritmos no Scratch.

Foram criados cinco projetos abordando o ciclo da água através de animações e explicações sobre o ciclo. Todos os cinco projetos continham pequenos erros nos algoritmos tais como explicações, ilustrações e animações sendo mostradas erroneamente. Além disso, eles tinham níveis de complexidade lógica diferentes, sendo que o primeiro projeto era o mais simples e o quinto, o mais complexo.

O ciclo da água era um conteúdo que já tinha sido explicado aos alunos na aula de ciências, porém fizemos uma revisão rápida sobre o ciclo antes de iniciarmos a atividade no Scratch. Após a revisão, disponibilizamos o estúdio aos alunos, que tiveram que executar os projetos, encontrar os seus erros e corrigi-los.

Aulas 12-13: Criando e Aprimorando um Jogo de Clicar


As aulas 12 e 13 abordaram o desenvolvimento e modificação de um jogo simples de Clicar¹⁹. No jogo, um personagem aparece em algum lugar aleatório do cenário e depois de alguns segundos desaparece. Este processo fica se repetindo e o objetivo do jogador é clicar neste personagem antes que ele desapareça. A ideia destas duas últimas aulas foi disponibilizar um projeto pronto aos alunos para que eles, a princípio, o testassem, entendessem o funcionamento e também se divertissem jogando. Em seguida, eles modificaram o jogo tanto visual quanto mecanicamente.


Na Aula 12, após disponibilizarmos o jogo aos alunos, demos alguns minutos para que eles o jogassem e testassem. No restante da aula, pequenas alterações no jogo foram feitas junto com os alunos de forma estruturada para garantir que todos os alunos teriam jogos, que apesar de diferentes visualmente, funcionassem da mesma forma mecanicamente. Utilizamos uma apresentação de slides²⁰ com as modificações para que os alunos pudessem fazê-las.

Na Aula 13, foi proposto que os alunos adicionassem um novo nível ao jogo, sendo esta uma alteração muito mais complexa do que aquelas feitas na aula anterior. Para auxiliar os alunos a conduzir estas modificações, nós criamos materiais de apoio²¹ para ajudá-los. Após adicionar o novo nível, foi dada a liberdade aos alunos para que eles

¹⁸Aula 11: Estúdio Bugs no Ciclo da Água 

¹⁹Jogo de Clicar 

²⁰Aula 12 - Alterando o Jogo 

²¹Aula 13 - Adicionando um Nível 

continuassem alterando o jogo conforme eles quisessem. Um material²² com quatro sugestões de alterações também foi desenvolvido para inspirar os alunos.

Aula 14: Aula Final

Na última aula do GTD, nós realizamos um kahoot final abordando diversos tópicos explorados durante o GTD e os alunos testaram os jogos uns dos outros e votaram nos projetos que eles mais gostaram. O kahoot utilizado²³ continha questões sobre conceitos estudados pelos alunos durante o decorrer do GTD e sobre blocos de comandos do Scratch utilizado pelos alunos na implementação de seus projetos.


Finalizado o kahoot, realizamos a votação dos projetos do Jogo de Clicar que os alunos acharam mais interessantes. Nos minutos finais da aula, após termos realizado a votação, deixamos os alunos livres para utilizar o computador e fazer o que eles quisessem.


6.3.2 Avaliação da Experiência

Conforme apresentado no Capítulo 4, avaliamos os recursos didáticos segundo a percepção dos alunos por meio do questionário de avaliação que era respondido no final da aula e grupos focais.

No Estudo I, com exceção do Programando em Papel, todos os recursos didáticos aplicados no GTD foram avaliados pelos alunos por meio do **Questionário de Avaliação** (Anexo C). A atividade Programando em Papel ficou muito complexa e longa para que os alunos pudessem finalizá-la e, no fim, acabamos esquecendo de distribuir o questionário para os alunos responderem. O Estacionamento Algorítmico e LightBot foram utilizados em apenas uma aula, logo foram avaliados apenas uma vez pelos alunos. Por outro lado, o Code.org e o Scratch foram utilizados em múltiplas aulas, logo, os alunos avaliaram múltiplas atividades realizadas nestas ferramentas. O Code.org foi avaliado quatro vezes e o Scratch cinco vezes, logo apresentamos o resultado consolidado de todos os questionários aplicados. Para comparar as ferramentas avaliadas múltiplas vezes com aquelas foram avaliadas uma única vez, os valores nos gráficos são apresentados em porcentagens. Os questionários e perguntas não respondidas pelos alunos que faltaram ao GTD ou que esqueceram de responder foram desconsiderados.

Os **grupos focais** foram realizados com os participantes na última semana do primeiro semestre letivo de 2019. O roteiro destes grupos focais (Apêndice D) abordou temas tais como a experiência dos alunos com a Computação, o conhecimento prévio

²²Aula 13: Outras Mudanças 

²³Aula 14 - Kahoot Final 

sobre os tópicos abordados e atividades conduzidas, o aprendizado através do GTD de Pensamento Computacional e sobre o próprio GTD (e.g, expectativas, sugestões, interesse em continuar). Imagens das ferramentas e atividades que os alunos tinham realizado no decorrer do GTD foram levadas para os grupos focais. Elas foram utilizadas como materiais de apoio para perguntar explicitamente sobre o que não foi discutido pelos alunos, ou seja, ferramentas e atividades que eles decidiram não mencionar ou que eles não se lembravam. Os alunos foram divididos em quatro grupos com alunos do mesmo sexo e da mesma turma. O intuito desta divisão foi deixá-los mais à vontade durante a condução dos grupos focais, especialmente as meninas. Assim, quatro grupos focais foram conduzidos, sendo cada um deles composto por 4 participantes.

Preferência dos alunos em relação aos recursos didáticos

A Figura 6.5 apresenta os resultados referentes ao item do questionário que abordava o quanto os alunos gostaram de ter realizado as atividades. Os valores entre parênteses nas legendas indicam a quantidade de respostas obtidas no total para cada ferramenta e atividade. Como informado previamente, o Code.org e Scratch foram recursos aplicados em múltiplas aulas, logo houveram mais respostas para eles.

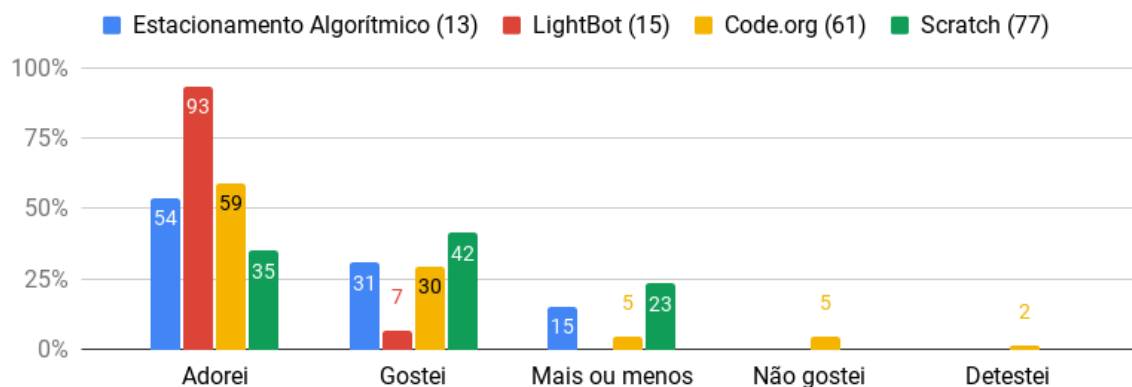


Figura 6.5: Avaliação das ferramentas e atividades

Em geral, os resultados foram positivos para todas as quatro ferramentas avaliadas. Considerando apenas o extremo positivo da escala, com exceção do Scratch, o número de alunos que disseram ter adorado as ferramentas foi maior que o número daqueles que disseram que gostaram.

Os comentários dos alunos sobre o **Estacionamento Algorítmico** nos grupos focais também foram positivos. Os participantes indicaram que gostaram de ter feito a atividade no curso. Alguns exemplos de comentários estão indicados abaixo.

G1P2: Ah, eu gostei. Eu gostei muito.

G1P1: Eu gostei também.

* * *

G1P8: Ah, eu gostei da primeira atividade que tem o carrinho.

G1P11: Ah, esse também é muito legal.

(Meninas - Sobre o Estacionamento Algorítmico)

O **Programando em Papel** não foi avaliado nos questionários; porém conversamos sobre ele com os alunos nos grupos focais. A maioria deles indicou que eles gostaram da atividade; contudo, um dos pontos negativos apontados pelos alunos foi que ela ficou entediante. G1P1 e G1P4 afirmaram que elas tiveram que resolver muitos desafios do Programando em Papel, tornando a aula repetitiva. Já para a participante G1P9, a atividade ficou entediante porque era fácil.

G1P9: Eu gostei, mas é entediante.

G1P1: É.

Pesquisador1: Entediante porque tava fácil ou porque tinha muita coisa?

G1P9: Entediante porque tava fácil, mas não era tão legal.

G1P4: Não foi tão chato. Só foi chato quando ficou repetindo.

G1P9: Teve muita aula só pra uma coisa.

* * *

G1P1: E também eu tava entediada de fazer isso porque a gente tava fazendo muito isso aqui [Programando em Papel].

(Meninas - Sobre o Programando em Papel)

O **LightBot** foi a ferramenta que os alunos mais gostaram (93% marcaram que adoraram e 7% que gostaram), sendo avaliado de forma positiva por todos os alunos nos questionários. A maioria dos comentários sobre o LightBot nos grupos focais também foram bastante positivos. Os alunos indicaram que gostaram muito dele e alguns dos participantes inclusive sugeriram que o utilizássemos mais nos GTDs.

G1P6: Ah, eu também gostei muito deste aqui.

G1P13: Eu também gostei do LightBot.

G1P10: Eu também gostei. Esse aí é muito legal.

G1P3: Eu gostei muito.

Pesquisador1: O que vocês acharam legal nele? Muita gente gostou dele.

G1P6: E também dá pra ele pular. Tem um tanto de jogo que a gente jogou que não pulava.

(Meninos - Sobre o LightBot)

* * *

G1P1: Na minha opinião esse aqui (LightBot) teria que fazer muito mais. A gente teve pouco...

(Menina - Sobre o LightBot)

Das quatro atividades avaliadas no questionário, apenas o **Code.org** foi avaliado negativamente pelos alunos (5% marcaram que não gostaram e 2% que detestaram). Nos grupos focais, alguns alunos sugeriram que não utilizássemos o Code.org em GTDs futuros; e outros, que o utilizássemos menos. O participante G1P11 afirmou que as atividades eram muito repetitivas; e G1P15, que era entediante.

Pesquisador2: E se fosse... vamos supor que a gente vá oferecer este GTD igual ele foi pra quem ainda não fez. Que que vocês teriam de sugestão pra ficar mais legal pra quem for fazer ele de outra vez?

G1P15: Tirar o Code.org.

G1P7: Fazer só uma vez o Code.org

G1P11: É.

Pesquisador1: Ficou muito repetitivo?

G1P11: Sim.

Pesquisador2: Mas o Code.org... a gente fez várias atividades. Assim, todas eram parecidas, mas diferentes, né? Tinha a do Angry Birds, tinha a da múmia, tinha a da abelha.

G1P15: Mas é um pouco entediante.

(Meninos - Sobre o Code.org)

No caso das atividades realizadas no **Scratch**, o Anime seu Nome foi aquela que recebeu mais comentários positivos nos grupos focais.

Pesquisador1: Então, todos vocês gostaram... em geral vocês gostaram do Scratch então? De criar as coisas?

Todas: Sim.

Pesquisador2: Destas atividades do Scratch, qual que vocês acharam mais legal?

G1P16: Acho que foi a primeira.

G1P8: A de animar o nome.

Pesquisador1: A de animar o nome?

G1P11: É

(Meninas - Sobre o Anime seu Nome)

Inclusive participantes tais como as alunas G1P4 e G1P9, que apontaram não ter gostado muito do Scratch, mencionaram que o Anime seu Nome foi legal. Elas disseram que acharam o Scratch entediante, porém elas não souberam explicar o que tornou o Scratch entediante para elas. Ademais, imprevistos influenciaram negativamente a opinião de G1P2 que afirmou não ter gostado da atividade realizada na aula anterior ao grupo focal (i.e.: Aula 13 - em que foi proposta a alteração do jogo, criando um novo nível para ele), pois ela acabou ficando sem computador para criar o seu próprio projeto. Um dos computadores do laboratório não estava funcionando e G1P2 teve que realizar a atividade junto com uma colega.

G1P4: Não gostei da atividade da aula passada.

Pesquisador1: Do Scratch?

G1P2: Do Scratch. Eu também não gostei.

Pesquisador1: Por quê?

G1P9: É entediante.

G1P4: Eu não gostei do Scratch porque é entediante.

Pesquisador1: Mas, por que foi entediante pra você?

G1P4: Sei lá...

Pesquisador2: Mas você não gostou de mexer no jogo, você não gostou de nenhuma atividade do Scratch.

G1P4: Não, eu gostei. Eu só não gostei de... mexer no...

G1P9: Dos números dançar [Anime seu Nome]?

G1P4: Gostei [do Anime seu Nome].

G1P9: Eu gostei um pouquinho do Kahoot!, um pouquinho dos zumbis [desafios da lição Labirinto: Ciclos], e muito daquele dos bichinhos... dos nomes que dançam [Anime seu Nome]. Eu gostava de fazer eles girar, colocar música...

Pesquisador2: Ah, legal. E o que você não gostou?

G1P9: A atividade da aula passada.

Pesquisador2: Mas você não gostou também porque você achou chato mexer no jogo ou porque... o jeito que é o... Por que que você não gostou da aula passada?

G1P9: Ah, eu não sei explicar. Eu não achei tão legal. Era entediante. Dava vontade de sair do joguinho (Scratch) e entrar no Friv [plataforma de jogos no navegador] rapidinho.

G1P2: Eu também não gostei do Scratch. Da atividade da aula passada. Eu não gostei da au... da aula passada porque eu fiquei sem computador, então não deu para fazer o joguinho.

(Meninas - Sobre o Scratch)

G1P14 afirmou que gostou do Scratch porque ele gostou de ficar programando. Além disso, G1P7 demonstrou interesse em aprender mais sobre o Scratch ao afirmar que tentaria utilizá-lo em casa para vir preparado para a aula seguinte.

G1P14: Agora aprendi a mexer no Scratch e é mó [sic] legal. Ficar programando as coisas.

G1P7: Eu vou tentar mexer, né?. Quarta, antes de eu vim [sic] pro GTD, eu vou tentar mexer e entender.

(Meninos - Sobre o Scratch)

Outro ponto que emergiu foi a liberdade que os usuários têm de criar coisas no Scratch. Este foi um ponto positivo da ferramenta mencionado por alguns participantes. Por exemplo, G1P13 e G1P3 mencionaram que gostaram do Scratch porque nele eles têm a liberdade de desenvolver os projetos da forma que eles querem, como por exemplo, escolher o que os personagens do projeto vão fazer. As meninas do segundo

grupo focal também disseram algo similar. Elas gostaram de poder criar seus próprios jogos, que foi uma nova experiência para elas.

Pesquisador1: Mas do Scratch, vocês gostaram dele?

G1P3/G1P13: Sim.

Pesquisador2: Todas as atividades? Vocês acharam legal?

G1P3: Sim.

O: Apesar de ser um pouco difícil né? E por que que vocês gostaram? Tem algum motivo específico?

G1P13: Porque a gente podia escolher o que o bonequinho ia fazer. Tipo... a gente vai criar tipo... o que o boneco vai fazer... o que o nosso boneco ia fazer e não o de outra pessoa.

G1P3: É

* * *

G1P16: Porque a gente tinha que criar a nossa própria coisa. Então, a gente podia fazer o que a gente quisesse.

G1P11: E eu nunca fiz isso.

G1P8: Nem eu.

G1P11: Aí eu achei legal e interessante.

Pesquisador2: Você também gostou, G1P8?

G1P8: Sim.

Pesquisador2: E você G1P5?

G1P5: Também.

(Meninos - Sobre o Scratch)

Em relação ao **Kahoot!**, apenas comentários falando bem do recurso didático emergiram durante os grupos focais. Ele foi a ferramenta favorita de vários alunos como G1P1, G1P7, G1P12 e G1P15 que mencionaram os desafios e a competição como fatores positivos da ferramenta.

Pesquisador2: E por que que vocês gostaram mais do Kahoot!?

G1P12: Porque tinha desafio.

G1P15: É. Desafio e competição.

G1P7: É muito legal.

G1P14: É. É muito legal.

G1P7: E era raciocínio rápido.

Meninos - Sobre o Kahoot

* * *

G1P1: O Kahoot... eu gostei dele... eu não sei explicar muito bem o por quê, mas eu achei ele mais legal.. assim... de fazer, mais divertido, de tá meio que disputando com as pessoas aqui do GTD.

Menina - Sobre o Kahoot

Dificuldade dos recursos didáticos

Os resultados referentes à dificuldade das atividades para os alunos estão sumarizados na Figura 6.6. Nenhum aluno considerou que o Estacionamento Algorítmico foi uma atividade difícil. No mínimo, eles consideraram que ele não foi nem difícil e nem fácil. Apenas uma pequena porcentagem dos alunos achou que o LightBot, Code.org e Scratch foram difíceis (7%, 8%, 5% respectivamente).

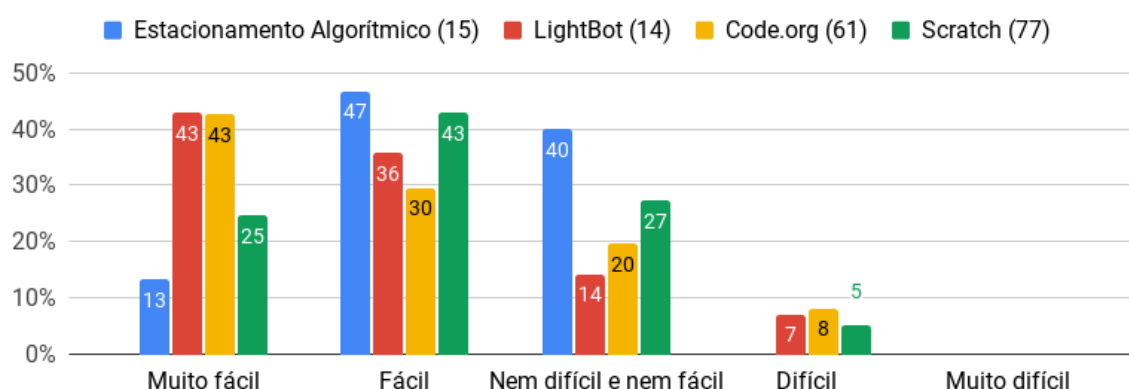


Figura 6.6: Dificuldade das ferramentas e atividades

O nível básico **LightBot** foi difícil para certos alunos como a participante G1P4, que comentou que o LightBot foi muito difícil para ela a partir do terceiro desafio. Assim, ela contou com a ajuda das amigas para que conseguisse passar do terceiro desafio. Por outro lado, as participantes G1P1 e G1P2 o acharam fácil e gostaram de resolver os desafios do LightBot.

G1P4: É, eu achei difícil. Eu não consegui mexer.

G1P1: Não foi difícil, foi fácil.

G1P4: A G1P1 gostou porque ela conseguiu fazer. Eu não passei da terceira fase.

G1P2: Ah, eu achei ele fácil. Eu amei ele.

G1P4: Você que passou da terceira fase pra mim.

G1P2: Verdade. A terceira fase... a G1P1 passou pra mim e pra G1P4.

Pesquisador1: Vocês gostaram?

G1P1/G1P2: Eu gostei.

G1P9: Ela não passou pra mim, né G1P1?

G1P4: Eu gostei, mas eu achei difícil.

G1P9: Eu gostei.

(Meninas - Sobre a dificuldade do LightBot)

Outro fator que fez com que o LightBot fosse um pouco difícil para alguns alunos foram os comandos de virar e a perspectiva do robô no cenário. Os comandos para

virar o robô para a esquerda e para a direita levam em consideração a perspectiva do robô. Assim, quando o robô está voltado para a tela do computador, a esquerda do robô é a direita do aluno e a direita do robô é a esquerda do aluno.

G1P7 Eu também não entendi.. eu acho que no LightBot, os trem (comandos) de virar. Era muito confuso.

G1P15: Sim.

G1P7 Eu sempre colocava um e sempre dava errado. Aí eu sempre colocava o outro.

G1P14: Pra virar o bonequinho você tem que tá do jeito que ele tá (imaginar a perspectiva do robô). Aí o que eu tinha que fazer era eu tinha que virar (o próprio corpo) assim (indicando como ele estava fazendo) pra saber pra que lado que ele ia.

(Meninos - Sobre a dificuldade do LightBot)

Em relação ao **Code.org**, a necessidade deles terem que utilizar a quantidade mínima de blocos foi apontado como algo difícil nos grupos focais por G1P12. A lição *Abelha: Condicionais* também foi considerada difícil por G1P12 e G1P7.

G1P12: Igual o da Abelha do Code.org, que você tem que pegar o Néctar.

G1P7: O da abelha eu gostei. Da abelha eu gostei. Era muito legal. Confuso, mas era legal.

G1P12: Às vezes não tem néctar no negócio. Aí é confuso, tem que fazer tudo de novo. E tem que usar os blocos o suficiente. Aí tem que fazer tudo de novo.

(Meninos - Sobre a dificuldade do Code.org)

No caso do **Scratch**, foi mencionado nos grupos focais que a programação nele é difícil e também que os algoritmos ficam muito grandes algumas vezes, fazendo com que eles ficassem com preguiça de arrastar os blocos a todo momento.

G1P14: Mas programar no Scratch é difícil.

G1P7: E dá preguiça ficar arrastando.

G1P14: Igual naquele jogo que você (Pesquisador1) fez na aula passada. Na hora que eu fui ver o código, tinha um tantão assim.

Pesquisador2: É. Ele é grande né?

G1P12: Eu não sabia nem fazer o negócio.

G1P7: É. É muito grande.

G1P7: Do G1P15 então! Nossa! (dizendo que o projeto de G1P15 ficou muito grande também).

(Meninos - Sobre a dificuldade do Scratch)

Pontos observados pelos pesquisadores no decorrer do estudo

Agora, apresentaremos alguns pontos de interesse observados por nós durante a condução do GTD de PC I 2019/1. Estes pontos, assim como os resultados dos questionários

e grupos focais, foram levados em consideração para realizarmos alterações no curso a fim de melhorá-lo para o segundo semestre de 2019.

Na primeira aula, foi observado que alguns alunos não participaram ativamente na resolução de todos os desafios do **Estacionamento Algorítmico**. Conforme apresentado na descrição da aula, a atividade foi conduzida em duplas para encorajar a colaboração na resolução de problemas. Notamos que as duplas adotaram abordagens diferentes para resolução dos desafios. Houve duplas que trabalharam em conjunto para resolver os desafios e duplas que preferiram trabalhar mais individualmente, sendo que cada membro tinha sua própria vez para resolvê-los. Além disso, notamos alunos que ficaram desmotivados e acabaram deixando que suas duplas resolvessem os desafios sozinhas. Também foi percebido que alguns alunos tiveram dificuldade em anotar os passos das soluções. Eles não conseguiram lembrar bem como tinham resolvido os desafios porque pedimos que eles resolvessem todos os desafios antes de anotar os passos. Considerando os pontos negativos observados, decidimos que no GTD do próximo semestre: (i) esta atividade seria realizada individualmente pelos alunos para que todos exercitem igualmente o raciocínio lógico e pratiquem o registro das soluções; e que (ii) os alunos deveriam registrar a solução de um desafio antes de tentar resolver outros desafios.

Em várias aulas do GTD, determinados alunos conseguiram terminar o que propomos em muito pouco tempo se comparado ao tempo gasto pelo restante da turma. Por exemplo, nas aulas onde aplicamos o Code.org (i.e.: Aulas 4, 5, 6 e 7), alguns alunos conseguiram terminar os desafios das lições muito rapidamente. Assim, para evitar que eles ficassem sem nada para fazer, nós disponibilizamos desafios extras ou outras lições no Code.org que abordavam o mesmo tema. Isso também ocorreu na Aula 12, onde realizamos as modificações no jogo junto com os alunos. Embora a aula tenha sido mais estruturada, alguns alunos tiveram dificuldades em fazer alguns passos sozinhos, requerendo o auxílio dos professores. Consequentemente, demoramos um pouco para avançar para o próximo passo em algumas partes da atividade, fazendo com que os alunos que conseguiram terminar rapidamente ficassem esperando e ansiosos.

Na *Aula 10: Criação de Jogos E Estórias*, a abordagem de disponibilizar os roteiros aos alunos digitalmente por meio de PDFs não funcionou muito bem. A separação dos passos em diferentes arquivos pareceu confundir os alunos. Alguns deles não entenderam que os diversos arquivos representavam diferentes passos para construir um mesmo projeto. Consequentemente, eles criaram diferentes projetos separados para cada passo da atividade. Alguns deles acabaram excluindo o que tinham feito quando passaram para o arquivo seguinte. Os alunos que escolheram criar uma história tentaram fazê-la sem seguir estritamente o passo a passo disponibilizado; contudo, eles

ainda assim conseguiram criar uma história simples apesar da história só acontecer se eles executassem os *scripts* do Scratch manualmente.

Em relação às atividades conduzidas no **Scratch** após o “Anime seu Nome”, nós percebemos que os alunos tiveram dificuldades em fazer sozinhos diversas das tarefas propostas. Em muitos momentos eles ficaram sem saber o que fazer ou a como superar os problemas que eles estavam enfrentando, inclusive com o auxílio dos roteiros das atividades disponibilizados a eles. Os roteiros das atividades disponibilizados no próprio site da ferramenta acabaram sendo um ponto de dificuldade para os alunos na Aula 10 porque alguns blocos utilizados no roteiro estavam com nomenclatura desatualizada, fazendo com que os alunos ficassem confusos. Notamos também, que a maioria deles não tentava resolver sozinhos os problemas que eles estavam tendo. Assim, eles frequentemente pediam a ajuda dos professores conduzindo o curso.

No que diz respeito ao **Kahoot!**, os alunos ficaram bem empolgados durante a condução das revisões. A gamificação trazida pela plataforma pareceu ter tido uma influência positiva neles. No entanto, observamos que vários alunos ficavam impacientes quando eles tinham que esperar o restante da turma a responder às perguntas. Alguns deles acabavam apressando os colegas que ainda estavam pensando e outros ficavam conversando após terem respondido as questões, falando alto qual resposta eles achavam que estava errada ou correta. Assim, a maioria das questões foram respondidas muito rapidamente pela turma embora eles tivessem 2 minutos para pensar e responder a cada uma delas. Estes pontos negativos ocorreram com maior frequência nos kahoots iniciais conduzidos.

Outro aspecto negativo do Kahoot! foi que ele não era muito adequado para as perguntas que queríamos incluir. Por exemplo, uma pergunta com um algoritmo do Programando em Papel necessitaria de alternativas com imagens das possíveis figuras que seriam pintadas pelo algoritmo. Contudo, as alternativas das perguntas podiam apenas estar no formato de texto. Assim, adaptações tiveram que ser feitas para contornar esta limitação. A plataforma nos deixa adicionar uma imagem para cada pergunta dos kahoots. Nós utilizamos essa funcionalidade para exibirmos uma imagem contendo o problema a ser resolvido pelos alunos e as opções de solução para este problema (Figura 6.7).

Essa abordagem funcionou bem para quase todos os problemas e questões que queríamos incluir nos kahoots. No entanto, para algumas delas, o tamanho da imagem não foi o suficiente para incluir todas as informações necessárias (i.e.: problemas e alternativas) de tal forma que eles pudessem ser vistas com clareza por todos os alunos no laboratório de informática. Nestas situações, alguns dos alunos, especialmente aqueles sentados no fundo da sala, tiveram que sair de suas mesas e ir até a frente da

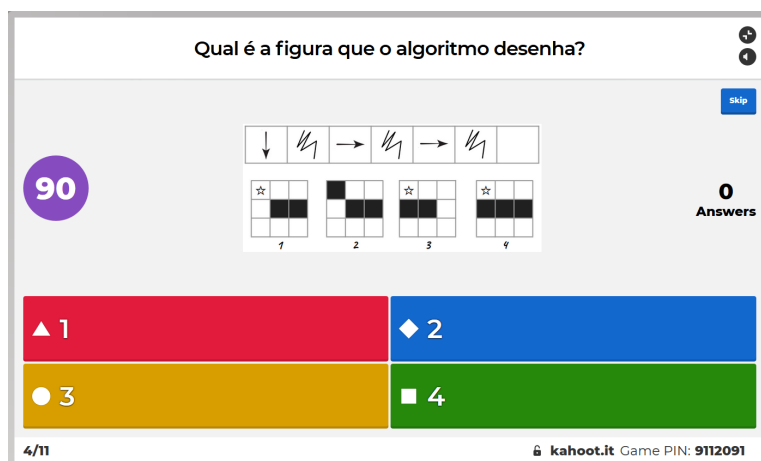


Figura 6.7: Exemplo de pergunta utilizada no GTD de PC I 2019/1

sala para que pudessem resolver os problemas.

Entrevistas com alunos que não continuaram participando

Nos grupos focais foi visto que os alunos demonstraram interesse em continuar participando do GTD. Logo, no segundo semestre de 2019, nós oferecemos o GTD de PC II para os alunos que participaram do GTD de PC I no primeiro semestre. Dos 16 alunos, oito resolveram continuar e oito resolveram não continuar. Destes, sete meninos e uma menina quiseram continuar participando e sete meninas e um menino não quiseram continuar. Assim, no início do segundo semestre de 2019, foram conduzidas entrevistas semiestruturadas e individuais com os alunos que optaram por não participar do GTD de PC II.

A principal motivação para conduzirmos esta entrevista foi entender o que estes alunos levam em consideração quando decidem quais GTDs eles querem fazer e o que os levou a não continuar participando. Os temas abordados no roteiro desta entrevista (Apêndice F) foram referentes às preferências dos alunos em relação às disciplinas regulares e GTDs do Centro Pedagógico, ao aprendizado, ao GTD de Pensamento Computacional e, finalmente, aos tópicos do Pensamento Computacional e Computação abordados. Os resultados destas entrevistas foram obtidos e analisados no projeto de conclusão de curso de Braga [2019].

Dos oito alunos que não continuaram participando do GTD no segundo semestre de 2019, apenas a participante G1P4 afirmou não ter continuado participando porque ela não gostou do tema do GTD e porque não conseguiu aprender sobre o tema. Os outros sete alunos afirmaram que gostaram de ter participado GTD de PC I e que conseguiram aprender sobre o tema. No entanto, quando eles foram escolher os GTDs

que eles queriam participar no segundo semestre de 2019, eles decidiram participar de outros GTDs para explorar outros assuntos.

Em relação ao interesse pela área da Computação, seis dos participantes (5 meninas e 1 menino) entrevistados nos disseram que tinham interesse pela área e que tinham vontade de aprender mais sobre a Computação no futuro. Os outros dois participantes (2 meninas), incluindo G1P4, disseram não ter interesse pela área e que não queriam aprender mais sobre o tema no futuro.

Também questionamos os alunos se eles tinham indicado o GTD de PC I para os seus amigos que não tinham participado no primeiro semestre. Com exceção da participante G1P4, que não gostou do GTD, todos os outros sete alunos, inclusive a participante que afirmou não ter interesse na área, recomendaram que os seus amigos e amigas participassem do GTD de PC I no segundo semestre de 2019.

6.3.3 Discussão e Lições Aprendidas

Nesta subseção apresentamos as discussões e o que aprendemos durante e após a condução do GTD de Pensamento Computacional 2019/1.

As atividades desplugadas foram tão boas quanto às atividades plugadas

As opiniões dos alunos sobre as atividades realizadas de forma desplugada e as atividades realizadas nos computadores não foram notavelmente diferentes. Para todas as ferramentas e atividades de ensino conduzidas houve alunos que gostaram e que também não gostaram delas, sendo que o número de alunos que afirmaram não terem gostado foi menor que os que afirmaram terem gostado para todas as ferramentas discutidas nos grupos focais. No início da primeira aula do GTD nós questionamos a turma a respeito da motivação deles em participar do GTD e todos os alunos responderam que eles escolheram o GTD porque eles gostavam de computador. Além disso, alguns deles também expressaram que queriam ter utilizado o computador durante a condução das duas primeiras aulas. Assim, acreditávamos que a opinião dos alunos em relação às atividades desplugadas seria menos positiva; contudo, os resultados dos questionários e grupos focais indicaram que eles também gostaram das atividades desplugadas conduzidas nas duas primeiras aulas. Embora todas as ferramentas e atividades tenham causado impressões positivas na maioria dos alunos, não podemos dizer que resultados similares teriam sido obtidos caso o GTD fosse conduzido exclusivamente por meio de atividades desplugadas que abordam conteúdos e práticas equivalentes às aquelas trabalhadas através das ferramentas plugadas.

Atividades do Programando em Papel e Code.org ficaram repetitivas

Como visto anteriormente, alguns alunos apontaram nos grupos focais que a Aula 2 ficou entediante devido ao grande número de exercícios similares do Programando em Papel que eles tiveram que resolver. Foram incluídos exercícios para serem resolvidos sem o comando de repetição e exercícios para serem resolvidos com o comando de repetição, aumentando o número de exercícios a serem resolvidos na segunda aula do GTD de PC I 2019/1 e fazendo com que eles não pudessem ser completamente finalizadas pelos alunos.

As lições plugadas do Code.org também ficaram repetitivas segundo alguns alunos. Um deles diretamente afirmou que ele achou que o Code.org ficou repetitivo e outro disse a mesma coisa, porém de forma indireta, ao sugerir que utilizássemos o Code.org em apenas uma aula em GTDs futuros. Ademais, após termos feito uma reflexão de toda a experiência do GTD de PC I, concluímos que três das aulas alocadas para o Code.org foram desnecessárias, podendo ter sido utilizadas para que os alunos desenvolvessem outros projetos no Scratch no fim do GTD. Especificamente, a Aula 4 (Labirinto: Sequência) e a Aula 7 (Abelha: Condicionais), sendo que esta última também foi avaliada por nós como ineficiente para o ensino e a prática das estruturas condicionais.

A quarta aula onde foi conduzida a lição *Labirinto: Sequência* foi tida como desnecessária, uma vez que o conteúdo e a forma como ele é praticado é equivalente ao que os alunos fizeram no LightBot na aula anterior. Todos os comandos que os alunos devem utilizar no *Labirinto: Sequência* (i.e.: ir para frente, virar à esquerda, virar à direita) estão presentes no LightBot. Além disso, o LightBot, inclusive, possui outros comandos (i.e.: pular, acender) que tornam a atividade mais complexa e desafiadora para os alunos. A linguagem de programação em blocos foi a única novidade introduzida na Aula 4; no entanto, ela poderia ter sido introduzida juntamente com os *Laços de Repetição* através da lição *Labirinto: Ciclos*. Portanto, a menos que seja necessário que os alunos pratiquem mais o sequenciamento de ações simples, a lição *Labirinto: Sequência* não é necessária caso o LightBot seja utilizado.

No que diz respeito a Aula 7 (Abelha: Condicionais), foi percebido durante a aula que os alunos ficaram bastante confusos sobre a utilização do bloco *SE* mesmo com o auxílio dos professores. Ademais, conforme apresentado na descrição da aula, a utilização do bloco *SE* na lição é simples e sempre feita da mesma forma para todos os desafios. Portanto, consideramos que a introdução e a prática das condicionais poderiam ter sido feitas diretamente no Scratch, o qual permite que as estruturas condicionais sejam utilizadas de diferentes maneiras, tornando o aprendizado de condicionais

mais significativo para os alunos.

Portanto, para o GTD de PC I 2019/2, nós decidimos (i) que não utilizaríamos os laços de repetições no Programando em Papel e que reduziríamos o número de exercícios e (ii) que não utilizaríamos as lições *Labirinto: Sequência* e *Labirinto: Ciclos*.

Problemas com as linguagens de programação visual

Como discutido no Capítulo 2, as LPV em blocos utilizadas em ferramentas como o Code.org e o Scratch buscam tornar a programação mais simples ao fazer com que a sintaxe não seja uma barreira para os usuários. Desta forma, os alunos não precisam memorizar inúmeros comandos e se preocupar com erros de compilação; porém, alguns pontos negativos das linguagens de programação em blocos no Scratch foram percebidas durante o GTD.

Primeiramente, quando projetos com regras mais complexas devem ser implementados, os *scripts* na linguagem visual em blocos ficam muito extensos, dificultando o entendimento para os usuários com pouca experiência. Nas Aulas 12 e 13 onde os alunos realizaram as modificações no jogo de clicar, os *scripts* que eles tiveram que utilizar ficaram longos se comparados aos *scripts* que eles tinham utilizado nas aulas anteriores. Um dos alunos apontou nos grupos focais que ele ficou intimidado após ter visto o tamanho dos *scripts* que eles tinham que utilizar nas duas últimas aulas.

Outro fator negativo das Linguagens de Programação Visual observada por nós e apontada pelos alunos é o fato que eles têm que ficar arrastando diversos blocos para implementar os projetos. Isto foi mencionado por eles como um dos pontos negativos do Scratch pois era entediante e eles ficavam com preguiça de ter que ficar arrastando vários blocos para desenvolver os projetos na ferramenta.

O Anime Seu Nome foi uma atividade adequada para introduzir o Scratch aos alunos

Nos grupos focais foi visto que a maioria dos alunos disse ter gostado da atividade Anime seu Nome, inclusive os alunos que afirmaram que o Scratch foi chato e entediante. Ambas a simplicidade da atividade e a ludicidade empregada na execução dela podem ter sido fatores que fizeram com que eles gostassem do *Anime Seu Nome* e tivessem impressões positivas sobre ele.

Os projetos do *Anime seu Nome* não são difíceis de serem criados, possuindo uma complexidade lógica adequada para os alunos do 6º ano que estão tendo contato com a programação e com o Scratch pela primeira vez. Por outro lado, o restante dos projetos que nós propusemos para o restante do GTD, tinham complexidade muito

mais elevadas se comparados ao Anime seu Nome. Os projetos do Anime Seu Nome no mínimo requerem que os alunos manipulem um pequeno número de blocos para criar as animações para as letras (Figura 6.8).

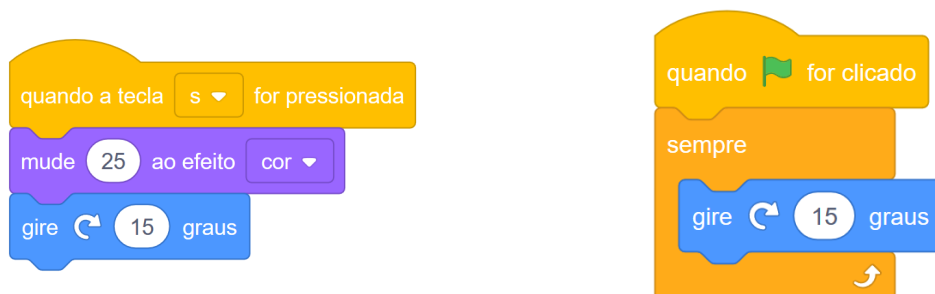


Figura 6.8: Anime seu Nome: exemplos de scripts criados pelos alunos

Os jogos propostos após o Anime seu Nome requeriam que diversos tipos de blocos fossem utilizados em conjunto para que a movimentação dos personagens e objetos e regras dos jogos fossem implementadas (Figura 6.9). Além disso, nós também introduzimos as estruturas condicionais, variáveis e operadores (i.e.: bloco verde na figura) nestes projetos, tornando a complexidade lógica destes projetos muito mais alta.

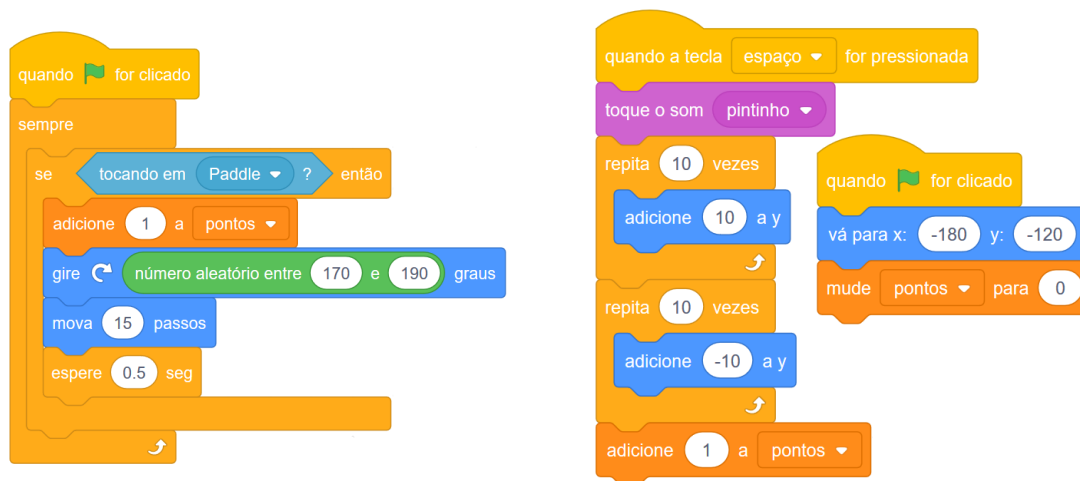


Figura 6.9: Exemplos de scripts criados pelos alunos: Jogo do Pong e Jogo de Pular

Embora a ordem em que as ferramentas tenham sido introduzidas aos alunos faça sentido do ponto de vista das Metáforas de Perspectivas Culturais (MPC) a fim de fazer com que a cultura estrangeira (i.e. Computação e Pensamento Computacional) seja exposta aos alunos de maneira gradual, a forma em que elas foram utilizadas teve grande influência na percepção da dificuldade e apropriação da cultura. Especialmente para ferramentas como o Scratch que permitem que os usuários tenham a liberdade de

desenvolver o que eles quiserem sem grandes limitações, se o grau de complexidade dos projetos e tarefas que os alunos devem realizar for muito alto, o processo de aprendizado e desenvolvimento das habilidades do PC pode ficar prejudicado. Isto foi percebido durante o GTD de PC I 2019/1, pois em diversos momentos os alunos ficavam sem saber o que fazer ou a como superar obstáculos que eles estavam enfrentando mesmo com o auxílio dos roteiros disponibilizados a eles.

Assim como utilizamos um projeto pequeno e de complexidade lógica baixa para que os alunos aprendessem a utilizar os principais blocos do Scratch, a utilização de diferentes projetos de menor porte para que os alunos praticassem a utilização de apenas um tipo de bloco poderia ter sido uma abordagem mais eficiente. Estes projetos teriam pequenas tarefas a serem implementadas e cada um deles focaria apenas na aplicação de um tipo de estrutura (i.e.: projetos apenas para prática de condicionais, projetos apenas para prática de variáveis). E por fim, caso haja aulas disponíveis, os alunos poderiam tentar finalmente desenvolver um projeto maior para que eles aplicassem e praticassem tudo que eles aprenderam por meio dos projetos menores.

É recomendado preparar lições ou tarefas extras

Como apresentamos anteriormente, houve grande discrepância no tempo gasto pela turma para finalizar as atividades em múltiplas aulas do GTD. Assim, destaca-se a importância em preparar tarefas extras ou desafios para alunos que tenham desempenho elevado e consigam terminar as tarefas propostas para a aula rapidamente. Desta maneira, estes alunos não ficariam ociosos durante o restante da aula, fazendo outras coisas no computador ou distraindo seus colegas de classe.

Outra opção a ser considerada para este tipo de situação é pedir que os alunos que conseguirem terminar as tarefas antes do fim da aula auxiliem os seus colegas que estão tendo dificuldades. Esta é uma abordagem interessante que pode contribuir para o desenvolvimento das habilidades do PC dos alunos que estão assumindo o papel de ajudantes; contudo, isto pode ser prejudicial àqueles sendo ajudados, pois os ajudantes poderiam simplesmente fazer as tarefas no lugar dos alunos tendo dificuldades. Este ponto negativo da abordagem acabou acontecendo na Aula 3, onde algumas alunas resolveram os desafios do LightBot no lugar de duas amigas, que estavam tendo dificuldades.

Kahoot! como Ferramenta de Revisão

A utilização do Kahoot! para revisar conteúdos durante o GTD de PC I 2019/1 foi um sucesso. Apesar de termos notado alguns pontos negativos referentes a como os alunos

se portaram durante os kahoots e aos tipos de perguntas que podem ser utilizadas, a experiência em geral não foi afetada negativamente para os alunos. Isto pôde ser visto durante a condução de todos os kahoots no decorrer GTD e por meio dos comentários positivos sobre o Kahoot! durante os grupos focais. Embora tenhamos utilizado o Kahoot! como um recurso secundário para que os alunos revisassem o que foi visto nas aulas anteriores de uma forma divertida e engajadora, ele foi um dos destaques nos grupos focais. Nenhum dos 16 alunos participantes mencionou pontos negativos do Kahoot! nos grupos focais, pelo contrário, falaram apenas bem da ferramenta, sendo considerado por vários deles a melhor atividade do GTD de PC I 2019/1. Sendo assim, consideramos que, no fim, o Kahoot! foi uma plataforma adequada e eficiente para condução de revisões dos conteúdos da Computação e dos recursos didáticos utilizados por nós no GTD de PC I 2019/1.

6.4 Estudo II

O segundo estudo, o qual foi realizado no segundo semestre de 2019 com outros 11 participantes, também teve como o foco a introdução da Computação e o desenvolvimento das habilidades do Pensamento Computacional por meio do ensino e prática da lógica de programação. Ele aconteceu por meio do GTD de PC I 2019/2, que foi o resultado da revisão do GTD ministrado no Estudo I a fim de mitigar os pontos negativos do curso apontados pelos alunos e observados por nós. Esta edição do GTD de PC I teve menos aulas disponíveis que o anterior, devido ao calendário da escola e imprevistos. Assim, o GTD acabou tendo 11 aulas, e não 14 como no semestre anterior.

No restante da seção, nós detalhamos as aulas e atividades conduzidas no curso, os resultados da avaliação da experiência, também obtidos a partir das percepções dos alunos e da nossa percepção e, no fim, discutimos os resultados e lições aprendidas.

6.4.1 Descrição das Aulas

A Tabela 6.2 apresenta a visão geral das aulas, indicando as MPC associadas a elas, os recursos adotados, o conteúdo abordado e uma breve descrição do que foi feito em cada uma delas. Na tabela, as linhas em verde representam aulas do GTD de PC I 2019/1 que sofreram modificações e as linhas em azul representam aulas completamente novas que foram incluídas no GTD de PC I 2019/2. Diferentemente do GTD de PC I 2019/1, a MPC Observador à Distância esteve presente no GTD de PC I 2019/2 por meio da atividade Decomposição da Turma da Mônica.

Tabela 6.2: Aulas conduzidas no GTD de PC I 2019/2

<i>Aulas</i>	<i>Recursos</i>	<i>Conteúdo</i>	<i>Descrição</i>
Aula 1	Estacionamento Algorítmico [VD]	Raciocínio Lógico	Apresentação do GTD de PC I, professores e alunos. Apresentação do GTD de PC I, professores e alunos. Prática do raciocínio lógico ao solucionar desafios do Estacionamento Algorítmico
Aula 2	Decomposição e Programando em Papel [OD,VTG]	Algoritmos e Decomposição	Introdução aos algoritmos. Decomposição de tarefas do cotidiano. Criação de algoritmos para pintar figuras utilizando comandos de movimento e colorir
Aula 3	LightBot [VTG]	Algoritmos	Revisão dos conteúdos das aulas anteriores e resolução dos desafios do <i>LightBot</i>
Aula 4	Code.org [VTG]	Laços de Repetição	Introdução aos loops. Resolução dos desafios da lição <i>Labirinto: Ciclos</i> para prática da identificação de ações e grupos de ações que se repetem e a utilização de <i>loops</i>
Aula 5	Code.org [VTG]	Laços de Repetição	Finalização da lição <i>Labirinto: Ciclos</i> e, para os alunos que a terminaram, resolução dos desafios das lições <i>Artista: Ciclos</i> e <i>Abelha: Ciclos</i>
Aula 6	Scratch [EcT]	Plano Cartesiano e Scratch	Revisão de todo o conteúdo visto até o momento. Introdução simplificado ao Plano Cartesiano e apresentação do Scratch e os seus principais blocos de movimento, aparência e eventos
Aula 7	Scratch [EcT]	Anime um Nome	Revisão do Plano Cartesiano e comandos de movimentos do Scratch. Condução da atividade <i>Anime seu Nome</i> , onde os alunos têm que animar as letras de seus nomes
Aula 8	Scratch [EcT]	Condicionais	Introdução às condicionais. Criação do Jogo Pong para prática das condicionais através da implementação das regras do jogo
Aula 9	Scratch [EcT]	Variáveis	Introdução às variáveis. Implementação de um sistema de pontuação para o jogo Pong desenvolvido na aula anterior
Aula 10	Scratch [EcT]	Desafios	Finalização da implementação do sistema de pontuação iniciado na aula anterior e implementação de desafios
Aula 11	Scratch [EcT]	Aula Livre	Criação de uma apresentação de slides para encontro dos pais. O restante da aula foi livre

Aula 1: Exercitando o Raciocínio Lógico

Para iniciar a aula, apresentamos o GTD de PC I 2019/2 e os professores que o estariam ministrando. Também pedimos que os alunos se apresentassem e nos falassem o que os levaram a escolher o GTD de PC I.

Em seguida, apresentamos o Estacionamento Algorítmico, suas regras e como ele seria realizado em sala de aula. Para isso, utilizamos o projetor de vídeo para apresentar um exemplo simples da atividade. Neste GTD, o **Estacionamento Algorítmico** foi realizado **individualmente** pelos alunos para que todos eles exercitassem igualmente o raciocínio lógico. Os cinco primeiros desafios da atividade foram disponibilizados aos alunos, que estavam sentados em dupla devido à organização do laboratório de informática do CP. Um aluno resolveu os desafios de número ímpar e o outro, os desafios de número par.

Pedimos aos alunos para anotarem a solução do primeiro desafio assim que eles

conseguissem resolvê-lo antes que eles tentassem resolver o próximo. Por fim, os alunos passaram suas soluções para que as suas duplas as testassem, seguindo os passos e indicando os erros na solução caso eles existissem.

Aula 2: Introdução aos Algoritmos e Decomposição de Problemas

Explicamos o conceito de algoritmos, fazendo a associação do conceito com o passo a passo do Estacionamento Algorítmico da aula anterior similarmente ao primeiro GTD.

Em seguida, exibimos um vídeo de curta duração demonstrando a importância da definição precisa de instruções através de um problema do dia-a-dia: preparar um pão com geleia²⁴. Então, conduzimos a atividade **Decomposição da Turma da Mônica** para que os alunos praticassem a decomposição de problemas. Cada aluno ficou responsável por decompor uma única tarefa. Depois que eles terminaram, discutimos as soluções com a turma, buscando determinar se elas estavam corretas, incorretas e como elas poderiam ser melhoradas.


Por fim, o **Programando em Papel**²⁵ foi realizado. Os alunos fizeram apenas uma versão simplificada do “Criando Algoritmos”, que continha 3 dos 6 exercícios utilizados no Estudo I para serem resolvidos sem a utilização do bloco de repetição.


Aula 3: Algoritmos e Sequenciamento de Ações Plugado - LightBot

Esta aula ocorreu de maneira similar a aula do GTD de PC I 2019/1. Nos primeiros minutos da aula foi conduzida uma revisão com os alunos utilizando o Kahoot!. Assim como no GTD anterior, foram revisados aspectos teóricos e práticos abordados na aula anterior. Os aspectos teóricos foram revisados através de questões sobre algoritmos. Já os aspectos práticos, através de questões sobre o Programando em Papel para que os alunos determinassem quais algoritmos desenhavam a figura mostrada na pergunta ou qual figura era desenhada pelo algoritmo mostrado na pergunta. Pequenas modificações foram feitas nas perguntas abordando o Programando em Papel, dado que neste GTD conduzimos o Programando em Papel sem a introdução do comando de repetição. Após a condução do kahoot²⁶, o link para que os alunos acessassem a versão livre do **LightBot** foi passado aos alunos para que eles resolvessem pelo menos os oito desafios do primeiro nível.

Nós pretendíamos introduzir o Kidlo Coding²⁷ (ferramenta similar ao LightBot) para os alunos que conseguiram terminar os oito desafios do primeiro nível do LightBot

²⁴Disponível em: <https://www.youtube.com/watch?v=pdhqwbUWf4U>

²⁵Aula 2 - Programando em Papel Simplificado 

²⁶Aula 3 - Kahoot de Revisão I 

²⁷Kidlo Coding 

porque os desafios do segundo nível são muito mais complexos para os alunos. Contudo, vários alunos acabaram passando para o segundo nível por conta própria e um dos alunos, para o qual apresentamos o Kidlo Coding nos disse que ele o achou muito infantil e que queria tentar resolver os desafios do segundo nível do LightBot. Assim, o Kidlo Coding acabou não sendo utilizado na aula e, similarmente ao primeiro semestre, deixamos que os alunos tentassem resolver os desafios do segundo nível do LightBot.

Aula 4: Aprofundando em Algoritmos: Loops

Nesta aula foi abordado e praticado a repetição de ações através da lição **Labirinto: Loops** do Curso 2 da plataforma Code.org. Antes de apresentarmos aos alunos o conceito de laços de repetição, um kahoot²⁸ foi conduzido nos minutos iniciais da aula para que os alunos revisassem os desafios básicos do LightBot.


Em seguida, apresentamos brevemente aos alunos o conceito de laços de repetição, para quê eles são utilizados e a plataforma Code.org juntamente com a linguagem de programação em blocos que os alunos utilizariam para resolver os desafios da lição.

Por fim, foi dado início a parte prática da aula, onde os alunos ficaram resolvendo os desafios da lição *Labirinto: Ciclos* até o fim da aula. Foi enfatizado que eles deveriam tentar utilizar o número mínimo de blocos para resolver os desafios. Também foi pedido que eles verificassem se eles realmente utilizaram o número mínimo de blocos ou não após eles resolverem os desafios. A própria plataforma diz aos alunos se eles resolveram os desafios de maneira eficiente ou não.

Aula 5: Aprofundando em Algoritmos: Loops (continuação)

Na aula 5, os alunos deram continuidade à resolução dos desafios da lição **Labirinto: Loops**, uma vez que eles não conseguiram terminá-los apenas em uma aula. Apenas uma aluna conseguiu terminar todos os desafios da lição na Aula 4. Essa aluna, inclusive, conseguiu completar todos eles utilizando a solução ótima (i.e.: solução utilizando o menor número de blocos).

Como foi visto que os alunos estavam tentando dificuldades em utilizar os blocos de repetição corretamente, foi proposto um exercício rápido e desplugado, onde foi dado um pequeno script do Code.org utilizando o bloco *REPITA* para que os alunos indicassem quais comandos e em qual ordem eles apareceriam se o script fosse executado. Depois deste pequeno exercício, enfatizamos uma vez mais sobre a importância de resolver os desafios de maneira eficiente e demos algumas dicas de como os alunos poderiam identificar padrões de ações para serem repetidos.

²⁸Aula 4 - Kahoot de Revisão II 

Durante o restante da aula, os alunos voltaram ao Code.org e continuaram resolvendo os desafios e aprimorando as soluções dos desafios que eles não tinham resolvido de maneira eficiente na aula anterior. Para a aluna que já tinha terminado todos os desafios na aula anterior, foi dado acesso a outras duas lições do Curso 2 do Code.org que também abordavam a repetição de ações: **Artista: Ciclos**²⁹ e **Abelha: Ciclos**³⁰. Deixamos que ela escolhesse os desafios que ela queria resolver. Estas lições também foram disponibilizadas aos outros alunos que terminaram os desafios da lição *Labirinto: Ciclos* nesta aula.

Aula 6: Plano Cartesiano e Introdução ao Scratch

No início da aula realizamos um kahoot³¹ para revisar tudo que os alunos aprenderam até o momento.

Após a revisão, apresentamos o Plano Cartesiano aos alunos, que foi algo que não ensinamos à turma do primeiro semestre. Consideramos essa explicação importante, uma vez que conhecer o plano e os seus eixos poderia facilitar o entendimento de como os blocos de movimento no Scratch funcionam. A movimentação no Scratch é guiada ou pelos eixos X e Y ou pela direção que um ator está apontando. Os alunos do 6º ano ainda não tinham estes conhecimentos sobre o Plano Cartesiano e número negativos, então estes tópicos foram apresentados de forma simplificada e por meio de exemplos para que os alunos visualizassem e praticassem a identificação dos pontos de no plano.

Por fim, apresentamos o Scratch aos alunos e associamos a explicação do Plano Cartesiano com os blocos de movimento. Junto com os alunos, testamos comandos básicos de movimentos, aparência, repetição e eventos. Durante os minutos finais da aula, nós deixamos que os alunos explorassem o Scratch e testassem outros comandos por conta própria.

Aula 7: Scratch - Anime seu Nome

A atividade Anime seu Nome, juntamente com a votação do projeto que os alunos mais gostaram, foi mantida a mesma do GTD anterior. No entanto, antes de dar início a atividade, foi feita uma revisão abordando o que os alunos aprenderam sobre o Plano Cartesiano e blocos de movimento do Scratch na aula anterior. O kahoot³² utilizado continha questões envolvendo a identificação dos pontos no Plano Cartesiano

²⁹Aula 5 - Artista: Ciclos [!\[\]\(80e2cf815349ac5586702775416baccb_img.jpg\)](#)

³⁰Aula 5 - Abelha: Ciclos [!\[\]\(7467c1dd0ef2428537a0061c375ff0d0_img.jpg\)](#)

³¹Aula 6 - Kahoot de Revisão III [!\[\]\(a9a3626ecfc46a7f5e12b55748edfd9d_img.jpg\)](#)

³²Aula 7 - Kahoot de Revisão IV [!\[\]\(88373652cb4db366000be55ff3e7e196_img.jpg\)](#)

e o funcionamento dos blocos de movimento que os alunos testaram na Aula 6. Durante o restante da aula, os alunos desenvolveram os seus projetos do Anime seu Nome.

Aulas 8 - Condicionais - Pong

Para as últimas aulas no Scratch, selecionamos o Pong como projeto para os alunos desenvolverem. O Pong é um jogo com regras simples, que permitem que os alunos possam praticar a utilização da repetição de ações e condicionais. Materiais de apoio foram criados para ajudar os alunos na implementação do projeto e das tarefas propostas nas três aulas.

No início da aula, realizamos uma explicação teórica sobre condicionais da mesma forma que foi feito no GTD anterior, utilizando a dinâmica das palmas, e apresentamos os blocos do Scratch referentes às estruturas condicionais. Depois da explicação os alunos fizeram o desenvolvimento básico do jogo, utilizando as condicionais para programar algumas das regras do jogo. Cada aluno desenvolveu a sua própria versão do Pong, podendo customizá-lo visualmente da forma que eles quisessem.

Apesar do site do Scratch oferecer seu próprio material de apoio para desenvolver o Pong, nós preferimos criar o nosso próprio material³³ de forma que os alunos conseguissem segui-lo mais facilmente porque o nome de alguns blocos de comando estavam desatualizados no material de apoio original.


6.4.1.1 Aulas 9 e 10: Variáveis - Pong


Nestas duas aulas, foram feitas a adição de pontuação ao jogo e a customização do jogo com o objetivo de torná-lo mais significativo e interessante para quem estiver jogando.

Para que a pontuação fosse adicionada ao jogo, seria necessário a utilização de variáveis, que são simples de serem criadas e manipuladas por meio dos blocos do Scratch. Assim, no início da aula apresentamos o conceito de variáveis aos alunos e como elas são especificamente utilizadas no Scratch.

Em seguida, os alunos fizeram a implementação da pontuação. Assim como para a Aula 8, também criamos o nosso próprio roteiro³⁴ para que os alunos seguissem e adicionassem a pontuação aos seus jogos a fim de evitar os problemas que os alunos tiveram ao seguir os roteiros originais do Scratch no GTD do primeiro semestre.

Para os alunos que finalizaram a adição da pontuação, propusemos algumas mudanças a serem realizadas no projeto, sendo elas tanto visuais quanto funcionais. As seguintes mudanças foram sugeridas: alterar cenário caso a pontuação chegue a um

³³Aula 8 - Pong: Condicionais 

³⁴Aulas 9 e 10 - Pong: Variáveis 

determinado valor, adicionar uma nova raquete no topo que não pode tocar na bola e adicionar uma nova bola que faz com que a pontuação diminua.

Além dessas mudanças, foi também dado um desafio para os alunos tentarem implementar. O objetivo do desafio era tornar o jogo mais difícil quando o cenário é alterado, simulando um segundo nível. Assim, sugerimos que a velocidade da bola aumentasse quando o cenário mudasse. Para resolver este desafio, é necessário a utilização de uma variável para controlar a velocidade da bola, utilizando os blocos de movimento para fazer a bola mover X passos, onde X é a variável que controla a velocidade.

Aula 11: Aula Final

A última aula do GTD de PC I 2019/2 foi dividida em dois momentos: (i) preparação de uma apresentação do GTD e (ii) aula livre.

No primeiro momento da aula, foi compartilhado com os alunos uma apresentação de slides criada na plataforma Google Slides para que eles pudessem falar sobre suas impressões do GTD e apresentassem os seus projetos do Scratch. Esta apresentação foi criada para ser exibida aos pais dos alunos, que viriam até o Centro Pedagógico para reunião de entrega dos resultados finais do ano letivo e seriam convidados para conhecer os GTDs que se dispuseram a apresentar as atividades realizadas.

No segundo momento da aula, os alunos que tinham terminado de montar suas partes da apresentação puderam entrar no site do Scratch para jogar e testar as diferentes versões do Pong criadas pelos seus colegas. Para aqueles que terminaram de testar os jogos dos colegas, foi deixado que eles explorassem livremente o Scratch para continuarem a desenvolver seus jogos, criarem outros projetos ou testarem projetos compartilhados pela comunidade do Scratch.

6.4.2 Avaliação da Experiência

Nesta subseção apresentamos os resultados dos questionários, grupos focais e também pontos observados por nós durante a condução das aulas. Assim como no Estudo I, adotamos o Questionário de Avaliação de Atividade (Apêndice C) e grupos focais para avaliarmos os recursos didáticos segundo os alunos.

Neste estudo, ambos o Code.org e Scratch foram aplicados em múltiplas aulas e os resultados apresentados agregam as respostas dos questionários. O Code.org foi aplicado em duas aulas, totalizando 22 respostas coletadas (11 em cada aula), já o Scratch foi aplicado em quatro aulas, totalizando 42 respostas coletas. Ambas as aulas 6 e 8 tiveram 10 respostas porque um dos alunos faltou nessas aulas.

Também conduzimos grupos focais com os 11 participantes do segundo estudo, porém uma aluna (G2P11) não pôde participar no dia que os conduzimos. Assim, realizamos uma entrevista individual com ela seguindo o mesmo roteiro dos grupos focais para conduzir a entrevista, que ocorreu de forma semiestruturada. O roteiro dos grupos focais do segundo semestre de 2019 (Apêndice E) abordou os mesmos temas do roteiro anterior com pequenas alterações e com a adição de um tema abordando especificamente o roteiro das atividades que os alunos usavam para construir projetos Scratch. Assim como no semestre anterior, utilizamos imagens das atividades do GTD como materiais de apoio para relembrar os alunos das atividades realizadas por eles. Nós tentamos colocar alunos da mesma turma nos mesmos grupos focais para deixá-los mais confortáveis

Preferência dos alunos em relação aos recursos didáticos

Os resultados para o primeiro item do questionário de avaliação estão apresentados na Figura 6.10. A grande maioria dos alunos avaliou todos os recursos didáticos positivamente (i.e.: adorei, gostei) nos questionários. Além disso, considerando apenas as avaliações positivas, a porcentagem de respostas que foram para o item “adorei” foi maior ou igual que as do item “gostei” para todos os recursos avaliados.

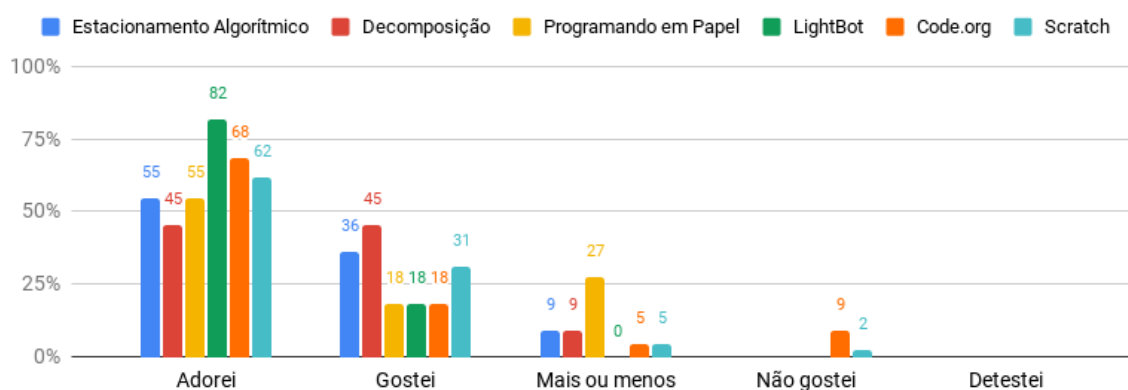


Figura 6.10: Avaliação das ferramentas e atividades

Da mesma forma, os comentários dos alunos sobre os recursos didáticos foram bastante positivos nos grupos focais. Muitos alunos afirmaram que não acharam nada chato no curso. Para G2P4 em especial, o GTD de PC I 2019/2 foi a aula mais legal do segundo semestre.

Pesquisador: Vocês não acharam nada chato?

G2P4/G2P6/G2P7: Não.

G2P4: Sua aula é a mais legal que tem. Da semana...

* * *

Pesquisador: E você G2P1? Pode falar se não gostou ou... [perguntando sobre o que ela não gostou]

G2P1: Eu gostei de tudo.

* * *

Nos grupos focais os alunos apontaram que gostaram do **Estacionamento Algorítmico** porque ele foi uma atividade desafiadora que os faziam raciocinar como no caso de G2P2, G2P3, G2P10 e G2P11.

G2P2: Ah, eu achei legal porque a gente tem que pensar bastante pra se mov... pra movimentar os carros...

G2P3: É.

* * *

G2P10: Eu achei legal porque era um momento pra gente raciocinar.

* * *

G2P11: Esse é legal. Do carrinho é legal.

Pesquisador: Ela não foi no computador e mesmo assim você acha que você gostou?

G2P11: Sim.

Pesquisador: E por que que você achou ela legal?

G2P11: Ah, porque a gente... é um desafio né?

Em relação às respostas neutras, apenas o **Programando em Papel** se destacou com 27% das respostas. O fato do Programando em Papel também ser desafiador, fazendo com que os alunos tivessem que pensar para resolver os desafios foi mencionado nos grupos focais como algo positivo.

G2P6: Lembro. É muito legal também.

G2P5: Eu gostei de fazer.

G2P6: Também [gostei de fazer].

G2P4: [Gostei] Demais.

Pesquisador: E por que vocês gostaram dessa? O que você achou? Por quê? Um motivo...

G2P5: Porque tava me... desafiando um pouco.

G2P4: E porque às vezes a gente achava que era uma coisa, mas era outra. Às vezes a gente achava que tinha que pintar um lugar, mas tinha que pintar o outro... confundia um pouco.

G2P4: Mas, aí é legal [porque confunde].

Pesquisador: Isso é um aspecto bom ou ruim?

G2P4: Bom.

Pesquisador: Bom por confundir e fazer você pensar? No caso...

G2P4: É.

G2P6: É. E faz você entender.

(Sobre o Estacionamento Algorítmico)

No entanto, outros alunos apontaram que não gostaram de ter que ficar escrevendo os comandos na atividade, sendo este um ponto negativo do Programando em Papel para eles.

G2P10: Eu achei chato.

Pesquisador: Achou chato?

G2P8: Eu achei legal.

G2P9: Eu achei normal.

G2P10: Ah, a gente ficar escrevendo as setas... assim... é complicado, eu achei chato.

G2P8: É verdade, esse negócio de escrever... eu achei chato.

(Sobre o Programando em Papel)

É possível perceber algumas similaridades entre os resultados do Estudo I e Estudo II. No Estudo II o LightBot também foi avaliado de forma extremamente positiva (82% adoraram e 18% gostaram), não havendo avaliações neutras e negativas. Similarmente às falas dos alunos sobre o Estacionamento Algorítmico, G2P4 disse que o LightBot foi legal porque ele o desafiava. Os aspectos visuais do LightBot chamaram a atenção de um dos alunos, que afirmou serem muito bons para um jogo no navegador.

G2P4: Eu gostei desse aí porque desafia muito a mente.

Pesquisador: Do robô?

G2P4: É.

* * *

G2P1: Eu gostei. Eu gostei muito desse aqui.

G2P3: Ah, eu gostei.

Pesquisador: E por quê? Por que vocês gostar... o que chamou atenção de vocês nele? O que vocês gostaram?

G2P2: Em questão de na... mesmo tempo que é visão 2D e 3D. Ou, eu achei isso muito legal pra um jogo de... de google, de navegador. Sem ter que instalar nenhum aplicativo.

G2P3: Eu gostei porque... [pensando]. Ah, eu não sei porque eu gostei.

G2P1: Ah, porque é legal.

(Sobre o LightBot)

O **Code.org** e o **Scratch** foram as únicas ferramentas com respostas negativas nos questionários, porém ninguém afirmou que as detestou. Apenas 9% (2 de 22) das respostas coletadas sobre o Code.org e apenas 2% (1 de 42) das respostas coletadas sobre o Scratch foram negativas. Já nos grupos focais, apenas dois alunos fizeram

comentários negativos sobre eles. Eles disseram que não gostaram do Scratch porque a ferramenta é chata e porque ter que utilizar muitos blocos para realizar ações simples foi algo ruim para eles. Eles também disseram que não gostaram do Code.org. G2P3 especificou que ela achou o Code.org chato porque ele era parecido com o Scratch.

G2P2: Não, eu não gostei do Scratch não.

G2P3: Também não.

G2P2: O Scratch é muito chatinho.

G2P3: Sim.

G2P2: No Scratch a gente fica lá [fazendo gesto da pessoa arrastando blocos]...

G2P3: [risadas]...

G2P2: Que nem bobo olhando pra tela do computador, arrastando bloquinho.

G2P3: Pra você fazer um bichinho mexer, você tem que col[ocar]...

G2P2: Bichinho mexe, mas ainda mexe travado. Parece que você tá 0,1fps.

G2P3: Assim, você coloca um monte de bloquinho pra fazer o bichinho andar, assim... 2 passos.

G2P2: Um quadrado.

G2P3: É.

G2P2: Um milímetro.

* * *

G2P2: Eu não gostei muito do Code.org não.

G2P3: Eu também não porque ele é tipo o Scratch.

G2P2: Ele é tipo o Scratch não. Ele é basicamente o Scratch.

G2P3: É.

G2P2: Basicamente o Scratch.

(Sobre o Scratch e Code.org)

Eles também apontaram que não gostaram de ter que ficar seguindo os passos dos roteiros do Scratch, uma vez que eles queriam ter a liberdade de fazer o que eles quisessem na ferramenta.

Pesquisador: Vocês acharam ruim seguir o roteiro?

G2P2: Achei. Péssimo.

G2P1: Achei também.

Pesquisador: Mas porque o roteiro estava ruim? Ou porque vocês queriam fazer alguma coisa... livre.

G2P2: Uma coisa mais livre...

G2P1: É.

G2P2: Porque se a gente errasse uma coisa, a gente aprendia outra coisa [sobre o aprendizado com os próprios erros].

G2P3: É.

G2P1: É.

G2P2: Em uma coisa mais livre, a gente abria a mente pra isso também.

(Sobre os roteiros do Scratch)

No entanto, a nova experiência que foi a criação de jogos e projetos foi algo positivo sobre o Scratch para G2P1. G2P10 também fez um comentário similar, pois ela gostou de ter criado e customizado o seu próprio jogo.

G2P1: Eu gostei do Scratch.

Pesquisador: Por quê?

G2P1: Ah, o Scratch porque tinha alguns blocos lá que eu não sabia. Eu achei esses blocos mais novos pra mim... pra eu saber, sabe? Que... eu nunca ouvi falar sobre esse... sobre esse... aplicativo...

Pesquisador: Você achou interessante porque era algo novo pra você?

G2P1: Sim.

* * *

G2P10: Eu gostei do jogo que foi a gente fez [sobre sua atividade preferida].

Pesquisador: Esse é o último do Scratch?

G2P10: Sim.

Pesquisador: E por que você gostou do Scratch, G2P10?

G2P10: É porque a gente criou o nosso próprio jogo.

Por outro lado, a criação do jogo Pong acabou sendo uma experiência um pouco frustrante para G2P4 e G2P5, pois eles perderam as coisas que eles criaram. G2P5 chegou a apagar coisas do projeto, que o Scratch não permite recuperar, fazendo com que ele tivesse que refazer o que ele já tinha feito. Já G2P4 teve um problema similar quando ele desligou o computador sem ter salvo o projeto.

Pesquisador: E a atividade que vocês gostaram menos?

G2P6: Não tenho.

G2P7: Não sei. Não tenho.

G2P4: Eu tenho uma.

Pesquisador: Qual? Pode falar.

G2P5: Eu acho que foi mais ou menos o da criação do jogo. Porque teve uma vez que eu sem querer apaguei tudo, aí deu um pouco de raiva. G2P5: Nem pra salvar automático assim, sabe?

G2P4: É esse aí mesmo. Foi o da criação de jogo porque... às vezes era difícil e meio complicado e também por causa disso, que teve um dia que eu desliguei meu computador. Eu achei que eu tinha salvado... e perdi tudo.

G2P5: Ainda mais, eu acho que eu perdi o que eu tinha feito na... na aula passada.

G2P4: A energia caiu e desligou o computador.

(Grupo 2)

Pontos positivos do Kahoot! que emergiram nos grupos focais foram a possibilidade dos alunos aprenderem ao jogar, o feedback dado aos alunos sobre as perguntas que eles erraram e também a competitividade.

G2P2: *Eu gostei mais do Kahoot! porque a gente compete, se diverte...*

* * *

G2P8: *O Kahoot! é bom porque você estuda... depois no finalzinho você vai dando as perguntas que você já deve ter estudado pra aprender... aí se errou, você sabe que errou. Você pode estudar mais pra saber sobre as coisas... pra aprender.*

Com exceção da participante G2P11, todos os alunos falaram exclusivamente bem do Kahoot!. G2P11 gostou de tudo que foi feito no GTD, porém o Kahoot! foi a atividade que ela menos gostou, pois ela teve dificuldades em responder às perguntas dos kahoots e acabava errando várias delas.

Pesquisador: *Ah, entendi. E o que você gostou menos?*

G2P11: *Acho que é do Kahoot! porque...*

Pesquisador: *Você gostou menos do Kahoot!?*

G2P11: *É. Eu achei legal, mas toda hora eu ficava errando porque eu não sabia muito não.*

Pesquisador: *Aí você gostou menos?*

G2P11: *Sim.*

Dificuldade dos Recursos Didáticos

A Figura 6.11 apresenta os resultados referentes à dificuldade das atividades para os alunos. Assim como no primeiro estudo, nenhuma atividade foi avaliada como “Muito difícil”. Com exceção do Estacionamento Algorítmico, a maioria das respostas foram positivas (i.e.: muito fácil, fácil) para todas atividades avaliadas, sendo que três delas (i.e.: Decomposição, LightBot e Scratch) foram consideradas como muito fácil por pelo menos 50% dos alunos.

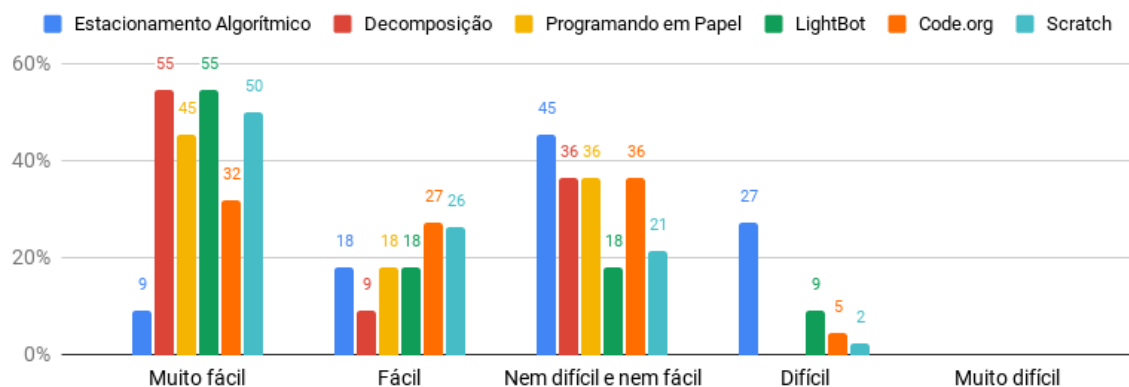


Figura 6.11: Avaliação das ferramentas e atividades

O **Estacionamento Algorítmico**, que no Estudo I foi considerado fácil pela maioria dos alunos, foi considerado fácil por apenas 27% dos alunos participantes do

Estudo II (9% muito fácil, 18% fácil). Além disso, ele não tinha sido considerado difícil para os alunos do primeiro estudo, mas foi considerado como difícil por 27% dos alunos que participaram do segundo estudo. Os 45% restante das respostas foram neutras. Contudo, nenhum aluno apontou nos grupos focais que ela foi difícil, e conforme visto previamente, algumas atividades sendo desafiadoras foi algo que eles consideraram positivo.

Uma quantidade pequena de alunos considerou o LightBot, Code.org e Scratch difíceis (9%, 5% e 2% respectivamente). Em um dos grupos focais os alunos mencionaram que a dificuldade do **LightBot** fez com que eles gostassem menos dele se comparado às outras atividades.

G2P8: Ah, gostei menos... não foi menos... eu gostei tipo... sabe o de comandar o boneco pra ir pro lado? Só achei difícil no finalzinho, mas de não gostar... eu gostei de tudo.

Pesquisador: E você G2P9? E qual que você gostou menos?

G2P9: Não sei. Foi daquele bonequinho que tinha que... aquele que você tinha que... tinha que acender a luz.

Pesquisador: E por quê?

G2P9: Porque ele é difícil.

Pesquisador: E você, G2P10?

G2P10: Eu também achei difícil aquela parte dos comandos. E só.

(Sobre a dificuldade do LightBot)

O **Code.org** foi a ferramenta com respostas mais equilibradas se considerarmos as duas opções positivas e a opção neutra (32% muito fácil, 27% fácil, 36% nem difícil e nem fácil). G2P11 afirmou que alguns dos desafios do Code.org foram difíceis, porém ela ainda considerou a atividade legal apesar das dificuldades enfrentadas. Similarmente, G2P9 teve dificuldades em resolver os desafios, porém ele ficou com receio de falar isso no grupo focal.

G2P11: Ah tá. Esse é mais ou menos, mas é legal também.

Pesquisador: Porque é mais ou menos?

G2P11: Ah porque eu achei um pouco difícil. Alguns (desafios), mas era legal.

* * *

G2P9: Eu achei ela legal, só que... Nada. Nada.

Pesquisador: Não, pode falar.

G2P9: É que eu errei.

Pesquisador: Mas você achou ruim isso ou... você achou legal mesmo você tendo errado?

G2P9: Legal.

Sobre a dificuldade do Code.org

No que diz respeito ao **Scratch** e aos roteiros das atividades utilizados nas aulas, G2P10 afirmou que a segunda aula no Scratch ficou um pouco entediante quando ela teve que ficar procurando os blocos corretos indicados no roteiro da atividade que passamos aos alunos. Segundo ela, ficou difícil de encontrar os blocos necessários utilizando os roteiros do Anime seu Nome.

G2P10: Eu achei legal, só achei entediante na hora de ficar procurando os negocinho [blocos] lá. Um tanto de vez eu tive que pedir ajuda

Pesquisador: Você fala pra procurar os comandos?

G2P10: Sim.

Pesquisador: Você achou difícil pra achar?

G2P10: Sim.

Sobre o Scratch

Em relação aos roteiros das atividades do Scratch, não houve consenso entre os alunos. Em geral, eles acharam que ajudaram, em especial os que eram coloridos, facilitando a identificação dos comandos. No entanto, alguns deles os acharam difíceis.

Os alunos do terceiro grupo apontaram que os roteiros das atividades foram importantes para que eles pudessem realizar as atividades no Scratch. G2P9 e G2P10 comentaram que os roteiros coloridos foram melhores, porém para G2P10 os roteiros em preto e branco também ficaram bons. G2P8 apontou que ambos os roteiros com cor e sem cor foram a mesma coisa para ele.

Pesquisador: Então, falando do roteiro... então vocês acharam que ajudou?

G2P10: Eu achei. Muito.

G2P8: Sim. Realmente.

Pesquisador: Mesmo quando... porque tinha... o primeiro que a gente passou tava... porque tinha... tava colorido. O segundo depois não tava. Você achou que ficou bem pior quando não tava colorido?

G2P10: O colorido ficou... melhor...

G2P9: melhor.

G2P10: ...mas, ajudou do mesmo jeito... preto e branco.

G2P8: Eu achei a mesma coisa. Ajudou dos dois jeitos.

(Sobre os roteiros do Scratch)

Para os alunos do primeiro e do segundo grupo focal os roteiros do Scratch foram difíceis de serem seguidos, especialmente os que foram impressos em preto e branco, fazendo com que alguns blocos fossem difíceis de serem localizados.

G2P2: Eu ficava olhando pra lá e pra cá, pra lá e pra cá. Achava bosta nenhuma.

G2P3: É... e tipo às vezes a gente nem achava os bloquinhos.

Pesquisador: Vocês acharam difícil de entender?

G2P2: Achei meio estranho.

Pesquisador: O primeiro que a gente passou tava colorido. E o segundo não tava.

G2P2: Como não tava colorido a gente não sabia em que quadro que era: se era comando, sequências...

G2P1: Controles...

* * *

Pesquisador: E vocês acharam que os roteiros eram simples de seguir ou vocês acharam...

G2P6: Mais ou menos.

G2P5: Depende.

G2P6: Alguns difíceis porque tava sem cor.

Pesquisador: Os sem cor, vocês acharam difícil?

G2P6: Aí a gente achava que tava em um... [lugar], só que tava em outro... é.

G2P7: Sem cor é difícil.

(Sobre os roteiros do Scratch)

Pontos observados pelos pesquisadores no decorrer do estudo

Notamos que pedir para os alunos anotarem os passos de determinado desafio do **Estacionamento Algorítmico** assim que eles conseguissem resolvê-lo foi uma abordagem melhor que aquela adotada no Estudo I. No GTD de PC I 2019/2 os alunos demonstraram ter menos problemas para lembrar e registrar os passos. Contudo, observamos que eles ainda tiveram dificuldade em lembrar quais eram exatamente os passos que eles tinham seguido quando eles foram fazer a anotação. Alguns alunos conseguiram lembrar os passos e resolver os desafios mentalmente sem precisar mover os veículos no tabuleiro em papel, mas não foram todos que tiveram esta facilidade. Isso foi ainda mais difícil para os desafios que requeriam um grande número de passos para serem resolvidos. Desta forma, vários alunos ainda tiveram que resolver os desafios novamente para lembrarem quais eram os passos que eles deveriam anotar na folha de registro de soluções.

Na segunda aula, a inclusão da atividade **Decomposição da Turma da Mônica** e do vídeo que demonstra a importância da definição precisa e correta de tarefas teve resultados positivos. A atividade de decomposição pareceu ser bem útil pelo o que foi observado da participação e reação dos alunos. Eles demonstraram ter entendido a importância de ordenar corretamente e detalhar bem os passos ao decompor os problemas. Da mesma forma, as reações dos alunos ao assistirem o vídeo foram positivas. Eles ficaram focados no vídeo e prestando atenção e vários deles deram risadas ao ver as consequências geradas pela má definição dos passos para preparar o pão com geleia. Em relação ao **Programando em Papel**, os alunos finalizaram os desafios propostos

muito rapidamente e a Aula 2 acabou ficando curta. Mesmo com a inclusão do vídeo e a atividade de decomposição, que não estavam presentes no GTD anterior, a segunda aula durou cerca de 50 minutos.

A reação dos alunos quando realizamos os **kahoots** no Estudo II foram similares às reações dos alunos participantes do Estudo I. Os alunos deste GTD também demonstraram ter gostado dos kahoots e de jogá-los, ficando sempre eufóricos e agitados sempre que as respostas eram reveladas. Alguns deles chegaram a pedir para que os kahoots fossem repetidos, embora as perguntas e respostas fossem as mesmas.

Na lição “Labirinto: Laços” do **Code.org**, os alunos tiveram grande dificuldade em identificar padrões e transformá-los em repetições para utilizarem menos blocos nos desafios. Assim, demos dicas individualmente para alunos que estavam tendo muitas dificuldades e, no fim, eles conseguiram entender melhor como identificar os padrões. Na aula seguinte, as mesmas dicas foram apresentadas para toda a turma no início da aula e, da mesma forma, foi observado que elas ajudaram a turma a resolver alguns dos desafios utilizando a menor quantidade de blocos.

Por fim, foi notado que a maioria dos alunos do GTD de PC I 2019/2 não tinham o costume de utilizar os computadores de mesa. No princípio, vários deles estavam digitando as *URLs* na caixa de texto do Google, que era a página inicial dos navegadores instalados nos computadores. Então, foi necessário que ensinássemos a estes alunos onde colocar as URLs de sites nos navegadores de internet. Em relação ao gerenciamento de contas nos sites do Code.org e Scratch, observamos que os alunos não tinham consciência de sair de suas contas antes deles fecharem os navegadores e desligarem os computadores. Além disso, vários deles salvaram as credenciais das suas contas nos navegadores, o que não é recomendável, uma vez que os computadores do Centro Pedagógico são públicos. Desta forma, qualquer pessoa poderia acessar as contas destes alunos sem permissão. Similarmente, também frisamos que eles deveriam sair de suas contas após eles finalizarem as atividades, porém muitos deles ainda não saíram de suas contas nas aulas seguintes. Comentários relacionados a isso também emergiram nos grupos focais.

G2P3: Eu não mexia... Eu não mexo muito no computador não. Então, eu não sei nada.

* * *

G2P8: Na minha casa eu nunca mexo no computador.

G2P10: Eu acho legal porque em casa a gente não tem tanto contato com o computador porque a gente fica muito mais no celular do que no computador. Aí agora... a gente sabe mais sobre internet e o que tem disponível no celular. E eu acho legal aprender sobre o computador às vezes. Até pra ensinar os nossos pais.

6.4.3 Discussão e Lições Aprendidas

Nesta subseção discutimos a experiência que tivemos com o GTD de PC 2019/2 e o que aprendemos ao conduzi-la.

Requerer que os alunos relembrem de cabeça as soluções do Estacionamento Algorítmico não é uma boa abordagem

A nossa proposta de utilizar o Estacionamento Algorítmico na primeira aula do GTD de PC I foi para fazer com que os alunos praticassem o raciocínio lógico e tivessem contato com algoritmos de forma indireta. A abordagem de fazer com que alunos registrem os passos para solucionar os desafios é interessante, pois as soluções registradas podem ser relacionadas com o conceito de algoritmos que é apresentado nas aulas posteriores. Contudo, o registro das soluções em papel é difícil de ser feito. Embora tenhamos pedido para os alunos do Estudo II registrarem as soluções logo em seguida, eles ainda tiveram problemas em lembrá-las.

Para evitar que alunos tenham problemas para lembrar as soluções, pode ser melhor instruí-los a irem anotando os passos utilizando lápis conforme eles tentam resolver os desafios e caso os passos anotados estejam incorretos, eles podem simplesmente apagar o que eles tinham anotado. Outra abordagem que pode ser adotada, é a utilização de folhas de rascunho para que os alunos não tenham que se preocupar em ficar apagando o que eles anotaram a todo momento.

A Aula 2: Programando em Papel não foi bem planejada

Em comparação a Aula 2 do GTD de PC I do primeiro semestre de 2019, que ficou muito complexa e extensa devido ao número de atividades e exercícios planejados, a Aula 2 do GTD de PC I 2019/2 ficou curta, pois os alunos conseguiram terminar as atividades propostas rapidamente. Conforme apresentado na descrição da Aula 2 do segundo semestre de 2019, o comando de repetição e a atividade Criando Figuras utilizadas no primeiro GTD não foram incluídas para evitar que a aula ficasse repetitiva e, consequentemente, entediante para os alunos.

A aula durou aproximadamente 50 minutos, logo mais desafios do Programando em Papel poderiam ser incluídos para que os alunos não ficassem ociosos durante os 30 minutos finais da aula. Uma outra possibilidade seria a inclusão da atividade Criando Figuras, em que os alunos devem desenhar uma figura, construir o algoritmo que a desenha e dar o algoritmo para que sua dupla o siga. Esta parte do Programando em Papel, que tinha sido feita na Aula 2 do GTD de PC I do primeiro semestre, foi

removida para evitar que a Aula 2 do GTD de PC I 2019/2 ficasse muito pesada para os alunos. Em nosso planejamento, consideramos que a adição do vídeo e inclusão da atividade de decomposição seria o suficiente para compensar a exclusão das atividades envolvendo o bloco de repetição e criação de figuras. Contudo, este não foi o caso e não tínhamos feito preparações para este tipo de situação tais como trazer exercícios ou atividades extras para os alunos que conseguissem resolver os desafios muito rápido. Assim, deixamos o restante da aula livre como forma de recompensar os alunos por conseguirem fazer tudo que tínhamos planejado.

Necessidade de endereçar problemas de usabilidade do Scratch

O salvamento automático dos projetos sendo criados no Scratch é inconsistente, ou seja, algumas vezes o salvamento automático acontece e em outras ele não acontece. Esta inconsistência não é um problema para usuários que têm a consciência de salvar os seus projetos constantemente. No entanto, ela se torna um grande problema para usuários que estão contando com o salvamento automático ou para usuários com pouca experiência com computadores, que foi o caso dos alunos participando do GTD de PC I 2019/2. Alguns alunos deste GTD acabaram perdendo os seus projetos, pois eles fecharam o navegador da internet sem terem salvo o projeto manualmente. Houve também uma queda de energia que fez com que eles perdessem projetos. Ambas estas situações levaram a frustração e irritação dos alunos quando eles viram que eles tinham perdido o que eles tinham criado.

Além disso, houve problemas com alguns alunos excluindo personagens em seus projetos do Scratch por engano. Embora poucos participantes tenham mencionado isto nos grupos focais, outros alunos do GTD de PC I 2019/2 acabaram fazendo o mesmo. Eles excluíram personagens de seus projetos acidentalmente e, conseqüentemente, tiveram que adicioná-los novamente e refazer todos os *scripts* que eles tinham criado previamente. Ao excluir personagens ou cenários de um projeto no Scratch, o usuário perde não apenas o objeto sendo excluído, mas também todos os scripts criados para este objeto. Esta ação no Scratch é irreversível, ou seja, após os objetos, juntamente com seus *scripts*, serem excluídos, não é possível que eles sejam recuperados. Ademais, a plataforma não pede ao usuário para confirmar se ele realmente deseja excluir um objeto ou não. Desta forma, caso o usuário clique no ícone de exclusão, o objeto é excluído imediatamente.

Mecanismos de controle que permitem que os usuários revertam ações realizadas e mecanismos de prevenção de erros que pedem que usuários confirmem ações perigosas são essenciais em sistemas computacionais e devem incluídas no Scratch. Desta

forma, o número de ocasiões em que usuários realizarão ações perigosas (i.e.: exclusões) por engano é reduzida e eles também terão a possibilidade de se recuperar de ações realizadas por acidente ou porque eles se arrependeram de realizar por determinado motivo. Enquanto isso não faz parte do Scratch, é importante que professores utilizando o Scratch em suas aulas peçam que os alunos tenham cuidado para não excluir objetos e frisem que os alunos devem sempre salvar os seus projetos antes de fecharem o navegador.

Criação de projetos próprios no Scratch é uma experiência mais engajadora

Similarmente aos alunos do GTD de PC I 2019/1, participantes do GTD de PC I 2019/2 também afirmaram que gostaram de ter tido a possibilidade de ter criado os seus próprios projetos no Scratch embora o tipo destes projetos e o seu funcionamento básico tivessem sido definido por nós. Nós fizemos a definição de quais projetos os alunos deveriam criar para fazê-los praticar determinados tópicos e utilizar determinados blocos de comandos. Esta abordagem pode se tornar um ponto negativo quando o projeto definido é algo em que os alunos não têm interesse, fazendo com que eles não consigam se identificar com aquilo que eles estão criando. Isto foi mencionado no grupo focal por alguns alunos que disseram não ter gostado de ter que ficar seguindo roteiros para criar os projetos. Para estes alunos, as atividades no Scratch teriam sido melhores se eles pudessem ter explorado temas que eles tinham interesse e ter criado os projetos que eles quisessem. Outro ponto levantado por um dos alunos foi o fato de que ele teria uma experiência mais significativa caso ele tivesse a oportunidade de explorar a ferramenta conforme ele quisesse. Ele afirmou no grupo focal que ao fazer o que ele quer, ele teria aprendido mais através da própria exploração e da tentativa e erro. Apesar do Scratch estar primariamente associado à MPC Estrangeiro com Tradutor, a utilização dos roteiros de atividades para os alunos seguirem fazem com que as atividades conduzidas no Scratch no GTD de PC I 2019/2 se aproximem um pouco da MPC Viajante em Tour Guiado embora os alunos tenham tido a oportunidade de realizar pequenas alterações no projeto.

Permitir que os alunos criassem os seus próprios projetos no Scratch talvez pudesse ter sido uma experiência mais engajadora para os alunos, porém a limitação de tempo para condução das aulas dos GTDs no Centro Pedagógico (1 hora e 20 minutos) e a própria inexperiência dos alunos, que estão utilizando pela primeira uma ferramenta de alta complexidade tal como o Scratch, torna isso complicado de ser alcançado em apenas um semestre. A nossa intenção com este GTD de PC I era permitir que os alunos aplicassem o que eles tinham aprendido até o momento através da cria-

ção de projetos pré-definidos no Scratch para que, em seguida, eles pudessem explorar a ferramenta e modificar os projetos livremente. Como mencionado anteriormente, esta abordagem que nós adotamos para o Scratch não está completamente associada à metáfora Estrangeiro com Tradutor; no entanto, ela é necessária para que os alunos adquiram noções básicas sobre o funcionamento do Scratch e dos principais blocos de comandos que estão disponíveis para os usuários do Scratch utilizarem. Caso houvesse mais tempo e aulas disponíveis, a próxima etapa do curso seria os alunos criarem os seus próprios projetos no Scratch explorando tópicos que eles tenham interesse.

O Scratch pode ser visto como uma ferramenta boba

Outro ponto de vista incomum e intrigante que surgiu nos grupos focais foi que o Scratch é uma ferramenta muito boba e sem graça. O aluno que teve esta reação em relação ao Scratch era um aluno que tinha expectativas de construir sistemas e mexer no hardware dos computadores do Centro Pedagógico da UFMG. Ele nos informou através do questionário de perfil e grupo focal que ele teve contato com computadores desde pequeno, pois o pai dele trabalha com TI. Para ele, que estava acostumado com computadores e jogos digitais, o Scratch era uma ferramenta muito simples e os projetos criados nele não se comparavam aos jogos tradicionais de computador. Isso veio pelo fato da interface do Scratch ser muito amigável e os projetos de animações e jogos serem muito limitados e lentos visualmente. De fato, se utilizarmos jogos de computador tradicionais como base de comparação, os projetos que os alunos estavam desenvolvendo no Scratch possuem qualidade inferior. Projetos complexos e de alta qualidade podem sim ser desenvolvidos no Scratch, porém para tal seria necessário a experiência de pessoas da área da Computação, que não é o caso dos estudantes do 6º ano do Ensino Fundamental que estavam participando do nosso GTD de PC I.

Sabemos que o Scratch foi criado tendo as crianças e pessoas sem experiência com a Computação como o público-alvo e, logo, a interface, linguagem de programação utilizada (i.e.: linguagem de programação em blocos) e sugestão de projetos disponíveis na plataforma são simples como forma de introduzir a Computação de maneira amigável às pessoas. Este aluno pode ter sido um caso à parte, porém a reação dele ainda foi um pouco surpreendente, pois não esperávamos que crianças da faixa etária de 11 e 12 anos teriam uma impressão como esta sobre o Scratch. Dessa forma, seria interessante avaliar se outros alunos nesta faixa etária e que tenham uma experiência maior com Computação, compartilham também esta visão deste aluno. Em caso positivo, talvez deva-se considerar outras ferramentas que poderiam ser usadas no lugar do Scratch, e que permitissem a introdução de PC, de acordo com a metáfora Estrangeiro com

Tradutor.

O uso da linguagem de programação em blocos não é sempre bem recebida

Os alunos participantes do GTD de PC I 2019/2 que disseram não ter gostado do Scratch (2 dos 12 alunos), indicaram que ter ficar arrastando os blocos a todo momento foi um dos pontos negativos da ferramenta. Ambos criticaram que eles têm que utilizar um grande número de blocos para fazer com que os personagens “*deem apenas dois passos*” nos cenários e, que ainda por cima, os personagens se moviam em poucos quadros por segundo (i.e: FPS - frames per second). Esta segunda crítica foi feita pelo aluno que considerou o Scratch uma ferramenta sem graça. Para eles, ter que ficar arrastando os blocos foi entediante e fez eles ficarem com preguiça. Eles também afirmaram que não gostaram do Code.org porque o Code.org é similar ao Scratch, visto que ambos utilizam a linguagem de programação visual em blocos. No entanto, o Scratch foi apresentado aos alunos após o Code.org, logo eles utilizarem a similaridade com o Scratch como um dos motivos para eles não terem gostado do Code.org foi algo inusitado. Se pensarmos logicamente, eles deveriam afirmar que acharam o Scratch ruim, pois ele é similar a lição *Labirinto: Ciclos* realizada no Code.org. No entanto, não sabemos se isto aconteceu porque o Scratch tinha sido a ferramenta mais recente que eles tinham utilizado no GTD ou porque eles já tinham discutido sobre o Scratch previamente no grupo focal.

Baixo desempenho no Kahoot! pode ser um aspecto negativo para os alunos

Assim como os alunos do GTD de PC I 2019/1, as impressões dos alunos em relação ao Kahoot! foram positivas. Os alunos gostaram de ter participado dos kahoot! ao ponto que vários deles pediram no decorrer do GTD para que repetíssemos os mesmos kahoots que eles já tinham feito quando os informamos que não tínhamos preparado novos kahoots com outras perguntas. Contudo, uma participante GTD de PC I 2019/2 apontou que dentre todas as ferramentas e atividades realizadas, o Kahoot! foi aquela que ela gostou menos. Considerando a experiência do semestre anterior e a reação da turma do GTD de PC II durante a condução dos kahoots, uma participante ter considerado o Kahoot! como a ferramenta que ela gostou menos foi algo inesperado por nós. Esta aluna apontou que ela de fato gostou do Kahoot!, porém ela gostou menos dele porque ela ficava errando bastante durante a condução da atividade. A dificuldade das ferramentas e atividades é um fator que influencia o que os alunos pensam em relação a elas, dado que dificuldade está diretamente relacionada à experiência dos alunos ao utilizar as ferramentas e conduzir atividades. Embora existam alunos que

gostam de ser desafiados e de superar obstáculos, há também aqueles que podem se sentir desmotivados e desencorajados quando eles não conseguem atingir os resultados esperados. A participante que disse ter gostado menos do Kahoot! se encaixa mais neste segundo grupo de alunos, uma vez que ela ficar errando as perguntas foi algo que fez com que o Kahoot! não fosse tão legal quanto as outras ferramentas e atividades.

Letramento Digital ainda não é realidade das crianças e deve ser abordado em cursos

Embora o foco do GTD de PC I 2019/2 seja o Pensamento Computacional e as suas habilidades, os alunos acabaram tendo também a oportunidade de desenvolver um pouco o Letramento Digital através das atividades conduzidas nos computadores do laboratório de informática do Centro Pedagógico. Conforme indicado nos resultados das nossas observações, vários alunos não sabiam utilizar navegadores da internet corretamente e não tinham consciência da importância de manter as credenciais de suas contas em segredo e sair de suas contas em computadores públicos.

É desejável que o Letramento Digital também seja abordado, mesmo que superficialmente, em cursos que têm o Pensamento Computacional como foco. Desta forma, diferentemente de como nós fizemos no GTD de PC I 2019/2 ao, por exemplo, ensinar informalmente os alunos que estavam tendo problemas para acessar as *URLs*, estes conhecimentos e habilidades necessários para determinadas aulas poderiam ser brevemente introduzidos a toda a turma no início da aula. Por exemplo, (i) em uma aula onde os alunos deverão acessar *URLs*, os professores podem montar slides para apresentar brevemente os navegadores de internet e como eles devem ser utilizados pelos alunos; ou (ii) em uma aula onde os alunos deverão criar ou acessar contas, é desejável que sejam discutidas as credenciais de acesso a contas, a importância de senhas fortes e outros tópicos que os alunos podem não saber (i.e.: salvar senhas em navegadores de computadores públicos, não sair de suas de contas após o uso, etc.).

6.5 Estudo III

O terceiro estudo teve oito participantes que optaram em continuar aprendendo sobre a Computação e o Pensamento Computacional. Ele aconteceu no segundo semestre de 2019 através do GTD de PC II, que teve como foco a utilização do Scratch para desenvolvimento de diversos tipos de projetos.

A seção foi organizada similarmente às seções onde apresentamos os dois primeiros estudos. Primeiramente, descrevemos as aulas conduzidas no GTD de PC II.

Em seguida, apresentamos os resultados da avaliação da experiência. E por último, apresentamos a discussão dos resultados e o que aprendemos ao conduzir esse GTD.

6.5.1 Descrição das Aulas

O GTD de PC II 2019/2 teve um total de 14 encontros. A Tabela 6.3 apresenta a visão geral das aulas ministradas.


Tabela 6.3: Aulas conduzidas no GTD de PC II 2019/2

<i>Aulas</i>	<i>Recursos</i>	<i>Conteúdo</i>	<i>Descrição</i>
Aula 1	Scratch [EcT]	Revisão e Plano Cartesiano	Introdução simplificada ao Plano Cartesiano. Revisão de conceitos e blocos do Scratch utilizados no GTD de PC I 2019/1
Aula 2	Scratch [EcT]	Sensores	Introdução aos blocos de sensores. Continuação às tarefas da aula anterior e utilização de sensores para movimentação
Aula 3	Scratch [EcT]	Clones	Introdução aos clones no Scratch. Criação de um jogo em que um personagem tem que desviar de objetos caindo do céu
Aula 4	Scratch [EcT]	Pergunte	Introdução aos blocos <i>pergunte e resposta</i> . Criação de um jogo de perguntas e respostas sobre temas escolhidos pelos próprios alunos
Aula 5	Scratch [EcT]	Variáveis	Introdução às variáveis. Criação de um jogo educativo para o ensino e prática de tabuada
Aula 6	Scratch [EcT]	Modificação dos Projetos	Revisão do conceito de variáveis e como elas são usadas. Implementação de modificações para os projetos criados nas aulas 2, 3 e 4
Aula 7	Scratch [EcT]	Definição de Projetos	Definição dos projetos individuais que os alunos desenvolveriam nas próximas aulas
Aula 8, 9 e 10	Scratch [EcT]	Criação dos Projetos	Desenvolvimento dos projetos individuais definidos pelos alunos
Aula 11	Scratch [EcT]	Árvores de Decisão	Introdução às árvores de decisão. Geração de uma árvore de decisão para guiar o fluxo de uma história a ser implementada no Scratch. Implementação da história
Aula 12	Scratch [EcT]	Árvores de Decisão	Desenvolvimento de uma pequena história em conjunto com os alunos. Os alunos decidiram o que aconteceria na história
Aula 13	Scratch [EcT]	Árvores de Decisão	Continuação à implementação da história iniciada na Aula 11
Aula 14	Scratch [EcT]	Criação de Apresentação	Criação de uma apresentação de slides para encontro dos pais. O restante da aula foi livre

Aula 1: Revisão e Plano Cartesiano

Primeiramente, foi distribuído para os alunos uma folha³⁵ com um exercício para eles relacionarem termos que tinham aprendido no GTD de PC I 2019/1 com as suas definições.

Em seguida, apresentamos o Plano Cartesiano à turma, que é um conhecimento importante para que os alunos consigam entender o funcionamento dos blocos de mo-

³⁵Aula 1 - Revisão de Conceitos 

vimento no Scratch e posicionamento de personagens e objetos no cenário. O Plano Cartesiano foi apresentado no GTD de PC I 2019/2, mas ele não tinha sido apresentado para a primeira turma no GTD de PC 2019/1. Utilizamos a mesma abordagem adotada no GTD de PC I 2019/2 para apresentarmos o Plano Cartesiano no GTD de PC II. A explicação foi feita de forma simplificada, pois o Plano Cartesiano é apenas ensinado aos alunos na disciplina de matemática nos anos posteriores. A explicação foi feita por meio de slides e exemplos com animações, dando foco na identificação de pontos.

Por fim, foi feita a revisão prática sobre o Scratch com a proposta de pequenas tarefas a serem resolvidas utilizando os blocos apresentados aos alunos no semestre passado³⁶. Algumas destas tarefas foram fazer um personagem andar em um quadrado, fazer um personagem andar em um retângulo e fazer personagens andarem quando teclas do teclado fossem pressionadas.


Aula 2: Revisão e Sensores

No início da segunda aula, foi conduzido um kahoot³⁷ para revisar o Plano Cartesiano e os comandos do Scratch que utilizam os eixos X e Y como referência. As questões do Plano Cartesiano continham pontos que deveriam ser identificados pelos alunos. As questões do Scratch continham comandos e os alunos deveriam identificar o resultado da execução destes comandos.


Em seguida, foi apresentado aos alunos um novo tipo de bloco do Scratch que não tinha sido apresentado oficialmente aos alunos no semestre anterior: Blocos de Sensores. Os blocos sensores e seus funcionamentos foram explicados através de slides e, após a apresentação, foi dada continuidade à atividade prática de revisão iniciada na Aula 1. Além das tarefas dadas na aula passada, foi pedido que eles utilizassem os sensores para controlar a movimentação de um personagem no cenário³⁸.

Aula 3: Clones

No início da Aula 3, foi feita uma pequena revisão com os alunos por meio dos slides para explicar as soluções para três tarefas abordadas nas Aulas 1 e 2 que os alunos tiveram dificuldade ou não conseguiram resolver completamente: fazer um personagem andar em círculos, fazer um personagem andar em triângulo e controlar um personagem através de sensores.

³⁶Aula 1 - Revisão do Scratch 

³⁷Aula 2 - Kahoot de Revisão I

³⁸Aula 2 - Revisão e Sensores 

Em seguida, foi apresentado aos alunos os blocos de comando responsáveis pela criação e controle de clones, que são cópias de objetos criadas durante a execução do projeto. Estes blocos foram praticados através da criação de um pequeno jogo em que o jogador controla um personagem para fugir de objetos caindo do céu³⁹. Os blocos de gerenciamento de clones tiveram que ser usados para fazer com que cópias de um objeto fossem criadas na parte superior do cenários e com que estas cópias caíssem do céu depois de serem criadas.

Aula 4: Pergunte


Na quarta aula, foi proposto a criação de um jogo de perguntas e respostas. Deixamos os alunos escolherem os próprios temas das perguntas caso eles desejassem. Perguntas da disciplina de Ciências abordando tópicos já aprendidos pelos alunos foram disponibilizadas para eles utilizarem em seus projetos caso preferissem.


Para a criação de um jogo de perguntas e respostas no Scratch, é necessário a utilização de dois blocos de sensores que não tinham sido explicitamente apresentados aos alunos até o momento: *pergunte* e *resposta*. Assim, no início da aula o funcionamento destes blocos foi ensinado aos alunos e, durante o restante dela, os alunos desenvolveram os seus jogos⁴⁰.

Aula 5: Variáveis

Na Aula 5, o principal tema abordado foram as variáveis. Apesar deles terem utilizado variáveis em alguns projetos conduzidos no semestre passado, o conceito e propósito das variáveis não tinha sido formalmente ensinado a eles.

O uso das variáveis foi praticado através de um outro jogo de perguntas e respostas, porém desta vez com um tema pré-definido. Foram propostas duas versões de um jogo educativo de matemática para o ensino e prática da tabuada⁴¹. Na primeira versão, o personagem tinha que perguntar ao jogador qual era o resultado da multiplicação de dois números aleatórios. Já na segunda versão, é perguntado ao jogador qual tabuada ele quer praticar. Por exemplo, se o jogador disser que quer praticar a tabuada do número 6, o personagem deve sempre perguntar ao jogador quanto é 6 vezes um número aleatório.

³⁹Aula 3 - Closes 

⁴⁰Aula 4 - Perguntas e Respostas I 

⁴¹Aula 5 - Perguntas e Respostas II

Aula 6: Modificando Projetos

No início da Aula 6 foi conduzido um pequeno kahoot de revisão com os alunos. Este kahoot⁴² teve como o tema as variáveis, que foram apresentadas formalmente aos alunos na aula anterior. Além do conceito e propósito das variáveis, havia questões em que os alunos tinham que indicar a saída dada a execução de alguns comandos ou qual utilização das variáveis era adequada para resolver determinado problema.

Para a parte prática da aula foram utilizados os projetos desenvolvidos pelos alunos nas aulas anteriores. Para cada tipo de projeto foram propostas modificações para os alunos fazerem nos projetos. Eles puderam escolher qual projeto e quais modificações fazer⁴³.

Aula 7: Definição de Projetos

A Aula 7 foi de caráter livre e exploratório para que os alunos definissem um projeto para eles criarem no Scratch. Durante a primeira metade da aula, foi deixado que eles testassem projetos compartilhados pela comunidade do Scratch na busca de ideias e inspirações para os seus próprios projetos.

No restante da aula, foi pedido que os alunos utilizassem o editor de texto para fazer o registro das especificações de seus projetos. Eles tiveram que registrar em um editor de texto no computador uma descrição dos projetos que eles queriam fazer e também como o projeto funcionaria.


Aulas 8, 9 e 10: Desenvolvimento de Projetos


Nas Aulas 8, 9 e 10 os alunos fizeram o desenvolvimento dos projetos que eles tinham definido na Aula 7. Alguns alunos não tinham terminado de definir e registrar o que eles queriam fazer na Aula 7, então eles terminaram de fazer isso no início da Aula 8.

Não foi definido quantas aulas seriam alocadas para que os alunos desenvolvessem seus projetos, pois não sabíamos exatamente quão rápido os projetos seriam desenvolvidos. No fim, três aulas foram necessárias para que todos conseguissem terminar as principais funcionalidades que eles queriam incluir nos projetos.

Aula 11: Definindo e Criando uma Estória

Nesta aula, introduzimos o conceito de árvores de decisão aos alunos de forma simplificada para que eles o aplicassem na geração de estórias que eles implementariam no

⁴²Aula 6 - Kahoot de Revisão 2 

⁴³Aula 6 - Modificações propostas 

Scratch. Após a explicação, os alunos tiveram que utilizar a árvore de decisão para definir como a estória que eles queriam desenvolver se desdobraria de acordo com escolhas feitas pelos personagens. A árvore de decisão que eles criaram tinha quatro fins diferentes. Eles utilizaram o restante da aula, para dar início ao desenvolvimento das estórias que eles criaram.

Aula 12: Criando uma Estória em Conjunto

A Aula 12 foi alocada para que os alunos terminassem o desenvolvimento de suas estórias. No entanto, a internet não estava funcionando bem no dia, impossibilitando que os alunos acessassem seus projetos no Scratch.

Para evitar a perda completa da aula, foi criada uma estória em conjunto com os alunos localmente. Projetamos a tela de um computador e os alunos sentaram todos no centro da sala. O tema da estória, as escolhas dos personagens, diálogos e os fins foram criados a partir de ideias dadas pelos alunos conforme desenvolvíamos a estória no Scratch. A estória criada foi sobre um aluno e suas escolhas referentes à escola (e.g: ir à aula ou não, estudar para a prova ou não, etc.).

Aula 13: Criando uma Estória

Na Aula 13, foi feito o que tínhamos planejado para a Aula 12 a princípio. Foi dada continuidade ao desenvolvimento das estórias dos alunos no Scratch. Como esta seria a última aula para tarefas do GTD, foi pedido que os alunos desenvolvessem no mínimo um dos lados da árvore de decisão no Scratch, resultando em dois fins diferentes. Todos os alunos conseguiram finalizar um dos lados da árvore e alguns deles conseguiram desenvolver os dois lados, resultando em estórias com quatro fins diferentes.

Aula 14: Aula Final

A última aula do GTD de PC II 2019/2, aconteceu da mesma forma que a última aula do GTD de PC I 2019/2. Os alunos criaram a apresentação de slides utilizando o mesmo *template* da apresentação do GTD de PC I 2019/2 que foi compartilhada com os alunos. Eles tiveram que incluir na apresentação as impressões que tiveram do GTD e falar sobre seus projetos individuais. Para os alunos que terminaram de montar a apresentação, foi deixado que eles fizessem o que eles quisessem no computador.

6.5.2 Aulas que não puderam ser Conduzidas

Como explicamos no início do capítulo, o planejamento inicial das aulas não pôde ser seguido completamente no GTD de PC II 2019/2. A nossa intenção era passar por todas as cinco Metáforas de Perspectivas Culturais com os alunos que participaram do GTD de PC I 2019/1 no primeiro Estudo. No entanto, a última MPC, o Estrangeiro sem Tradutor, acabou não estando presente neste GTD. Nossa intenção era apresentar a linguagem de programação Python aos alunos para que eles pudessem criar pequenos programas de entrada e saída de texto nas últimas aulas do GTD.

Neste GTD houveram dois imprevistos. A Aula 7 teve que ser postergada em uma semana, pois o Centro Pedagógico aderiu a uma greve de dois dias, impossibilitando a condução desta aula no dia previsto. Além disso, houve o problema com a internet que ocorreu na Aula 12, fazendo com que os alunos não pudessem terminar suas estórias no dia definido. Desta forma, não pudemos ter nem mesmo uma única aula completa para apresentarmos a linguagem Python mesmo que de maneira muito breve e sucinta.

6.5.3 Avaliação da Experiência

No Estudo III também foi utilizado o mesmo Questionário de Avaliação (C) para avaliar as aulas do Scratch que nós planejamos. Assim, foram aplicados sete questionários de avaliação no decorrer do semestre. As aulas dedicadas aos projetos dos alunos e aulas extras conduzidas para finalização de tarefas não foram avaliadas. Entrevistas individuais e semiestruturadas (Apêndice G) também foram conduzidas com os oito alunos participantes. Sete destas entrevistas foram conduzidas pessoalmente no próprio Centro Pedagógico da UFMG e uma delas, com a participante G1P1, foi conduzida por telefone. A participante G1P1 não foi à escola no dia que conduziríamos a entrevista e, logo, não pudemos entrevistá-la pessoalmente. Em seguida, nós apresentamos os resultados dos questionários, entrevistas e também dos pontos relevantes observados durante a condução do GTD de PC II no Estudo III.

Preferência dos alunos em relação aos recursos didáticos

A Figura 6.12 apresenta os resultados do questionário referente a quanto os alunos gostaram das aulas. Algumas aulas tiveram sete respostas, pois um dos alunos não foi a aula e, logo, não pôde avaliá-las.

Nenhum aluno avaliou negativamente as atividades nos questionários de avaliação. Os comentários nas entrevistas, em geral, também foram positivos. Vários alunos comentaram que eles gostaram de todas as atividades feitas no GTD de PC II.

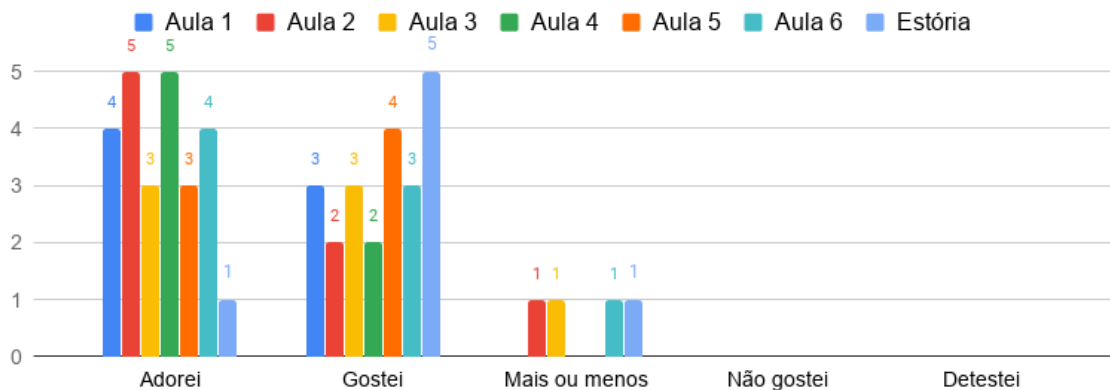


Figura 6.12: Avaliação das ferramentas e atividades

Pesquisador: E qual que você gostou menos? Ou que você não gostou?

G1P1: Ah, nenhuma. Eu gostei de todas.

* * *

Pesquisador: E dessas, teve alguma que você gostou menos ou que você não gostou?

G1P6: Não.

Pesquisador: Não foi nada chato pra você? Pode falar se tiver...

G1P6: Não. Tiveram algumas melhores que as outras, mas as outras não são ruins.

* * *

Pesquisador: E o que você gostou menos? Ou não gostou?

G1P12: Não tem.

* * *

G1P14: Ah, eu achei bem legal tudo.

Pesquisador: Legal? Nenhum você gostou menos?

G1P14: Não.

(Sobre o GTD de PC II)

Dentre as sete aulas avaliadas nos questionários, as aulas 2, 3, 6 e 11 foram avaliadas como mais ou menos, porém por apenas um dos alunos que responderam aos questionários. As Aulas 1 e 2, onde foi feita a revisão de blocos de comandos do Scratch, foram apontadas por G1P6 e G1P15 como as aulas que eles gostaram menos, pois os projetos não eram interessantes e porque foram um pouco difíceis respectivamente.

Pesquisador: Então, qual que você achou que foi a menos melhor...

G1P6: É... a primeira. O bicho só ficava girando.

Pesquisador: Você achou chato?

G1P6: É... só ficava girando, girando, girando. Aí perdeu a graça.

* * *

Pesquisador: E qual delas você gostou menos? Ou que você não gostou?

G1P15: De fazer andar no quadrado, e do círculo porque... e de primeira, eu não tinha entendido como que era pra fazer. Eu fiquei um pouquinho irritado.

Pesquisador: Mas, você conseguiu fazer né?

G1P15: Sim.

(Sobre as Aulas 1 e 2)

A Aula 11, onde foi feita a criação da árvore de decisão para estórias e a sua implementação no Scratch, não foi tão bem avaliada se comparada ao restante das aulas (1 - Adorei, 5 Gostei, 1 - Mais ou Menos). Nas entrevistas, o projeto da estória foi o único projeto que recebeu comentários completamente negativos, porém por G1P7 apenas, que mencionou ter gostado de tudo que foi feito no GTD com exceção do projeto da estória, que foi muito chato e difícil para ele.

G1P7: Aquela atividade da estória foi muito chata.

Pesquisador: Você achou a da estória muito ruim?

G1P7: Muito.

Pesquisador: Por quê?

G1P7: Porque é muito chato. É difícil e é chato.

Pesquisador: Foi a mais chata de todas pra você?

G1P7: Foi. Todas as outras foram legais.

(Sobre o projeto da estória)

Por outro lado, G1P1 e G1P14 indicaram que a árvore de decisão foi o tema da Computação mais legal deste GTD. G1P1 relacionou o tema com um jogo que, similarmente as estórias criadas pelos alunos, possui diferentes finais dependendo das escolhas feitas pelos jogadores. No caso de G1P14, ele gostou de utilizar as árvores de decisão para definir o rumo que a estória criada por ele iria tomar.

G1P1: Eu... o que eu mais gostei... eu acho... foi a da árvore de decisão que dá pra você montar uma história com vários finais diferentes. E eu conheço um jogo que é parecido, que usa uma árvore de decisão, muito grande. O The Walking Dead. Tem várias decisões e cada episódio você vai tomando escolhas e vai terminando cada um com um final diferente. Então dá pra você fazer várias vezes que você sabe todos os finais. Por isso que eu acho isso legal.

* * *

G1P14: Ah, eu gostei mais... foi o da estória.

Pesquisador: O da estória? Esse último... e por quê?

G1P14: É mais legal, que você pode criar os caminhos.

(Sobre o projeto da estória)

O restante das aulas (Aulas 1, 4 e 5) receberam apenas respostas positivas nos questionários. Dentre elas, a Aula 4, onde os alunos criaram um jogo de perguntas pela primeira vez, foi a aula mais bem avaliada com 5 respostas para o item *Adorei* e 2 para o item *Gostei*. Nenhum aluno nas entrevistas fez comentários negativos sobre os jogos de perguntas e respostas desenvolvidos nas aulas 4 e 5. Para G1P1, os jogos de perguntas e respostas foram os projetos mais interessantes, pois através deles é possível que o conhecimento das pessoas sobre determinado tema seja testado e que elas aprendam e pratiquem enquanto estão jogando. G1P6 também indicou que gostou mais dos jogos de perguntas porque eles podem aprender jogando e do jogo de tiro desenvolvido em seu projeto individual porque é divertido.

G1P1: As de fazer... é... perguntas. Perguntas do Gray's Anatomy que eu fiz, a de Matemática e a de Ciências. Foi as que eu mais gostei.

Pesquisador: Porque que você gostou delas?

G1P1: Eu gostei porque dá pra conhecer, se as pessoas sabem do tema. E também aquela de Ciências e de Matemática, é... dá pra aprender com elas. Dá pra estudar.

* * *

G1P6: Eu gostei mais das perguntas e do de tiro.

Pesquisador: E por quê? O das perguntas?

G1P6: Porque você aprende brincando.

Pesquisador: E o de tiro? Porque que você gostou mais?

G1P6: Ah, o de tiro é divertido. Você pode desviar dos negócio (das bolas) e você pode destruir também.

(Sobre os jogos de perguntas e respostas)

O fato de termos utilizado apenas o Scratch no GTD de PC II também influenciou a opinião dos alunos sobre o curso. Houve alunos tais como G1P6 e G1P15 que acharam o GTD de PC II melhor, pois foi dado foco ao Scratch e eles puderam criar mais projetos.

Pesquisador: E sobre... em relação a este GTD em comparação com o outro. O que você achou?

G1P6: Esse daqui foi melhor... a gente conseguiu fazer mais coisa.

Pesquisador: E em relação à diversão?

G1P6: Esse daqui é bem mais legal porque a gente pode criar mais coisa.

* * *

G1P15: Eu prefiro esse [ao GTD de PC I 2019/1].

Pesquisador: Por quê?

G1P15: Porque nesse daqui a gente viu mais coisas. E teve mais o Scratch.

Pesquisador: Por que você gostou mais do Scratch?

G1P15: Porque o Scratch dá pra fazer mais coisas e criar o seu jogo.

(Sobre as abordagens do GTD de PC I e GTD de PC II)

Por outro lado, G1P7 considerou a abordagem do GTD de PC I melhor que a do GTD de PC II, pois o primeiro GTD foi mais diversificado no que diz respeito às ferramentas utilizadas, fazendo com que o GTD de PC I ficasse mais legal.

Pesquisador: Ah é. Esse a gente fez pouco, né? E no outro também a gente fez passou diferentes ferramentas... diferentes sistemas, teve o LightBot, teve o Scratch, teve o Code.org... Então, você achou que isso foi pior ou melhor? Porque nesse a gente só ficou no Scratch é? Uma coisa só.

G1P7: Porque... é... esse, a gente focou no Scratch. Esse foi mais sério pra aprender mais sobre o Scratch. No outro, tava mostrando jogos pra gente entender sobre os comandos... que todos são baseados em comandos pra gente fazer o Scratch.

Pesquisador: Mas se fosse olhar essas duas abordagens: só o Scratch e o outro com várias coisas diferentes... qual que você achou mais legal?

G1P7: O outro.

(Sobre as abordagens do GTD de PC I e GTD de PC II)

Por fim, houve participantes que mencionaram nas entrevistas o fato de termos utilizado pouco o Kahoot! no GTD de PC II em comparação ao GTD de PC I conduzido no semestre anterior. Por exemplo, G1P14 e G1P15 sugeriram que realizássemos mais kahoots caso conduzíssemos o GTD de PC II novamente no futuro e G1P7 afirmou que ele achou o GTD de PC II pior, pois o Kahoot! foi utilizado menos.

Pesquisador: E em relação às sugestões. O que que você achou que ficou ruim desta vez?

G1P15: Só teve um Kahoot!...

Pesquisador: Foi só um que a gente fez?

G1P15: Foi 1 ou 2...

* * *

Pesquisador: E sugestões? O que você achou que a gente poderia ter feito diferente... pra fazer ficar mais legal...

G1P14: Não sei. Acho que dava pra fazer mais atividades de Kahoot!.

Pesquisador: E em questão de diversão, você achou esse mais divertido? Ou o outro?

G1P7: O outro porque tinha mais Kahoot!.

(Sobre a pouca utilização do Kahoot!)

Dificuldade dos recursos didáticos

Os resultados apresentados na Figura 6.13 ilustram o quão difícil foi para os alunos cada uma das sete aulas avaliadas.

A primeira aula foi a única que obteve apenas respostas positivas (1 - Muito Fácil, 6 - Fácil). Esta aula foi a menos complexa do GTD visto que foi uma revisão de

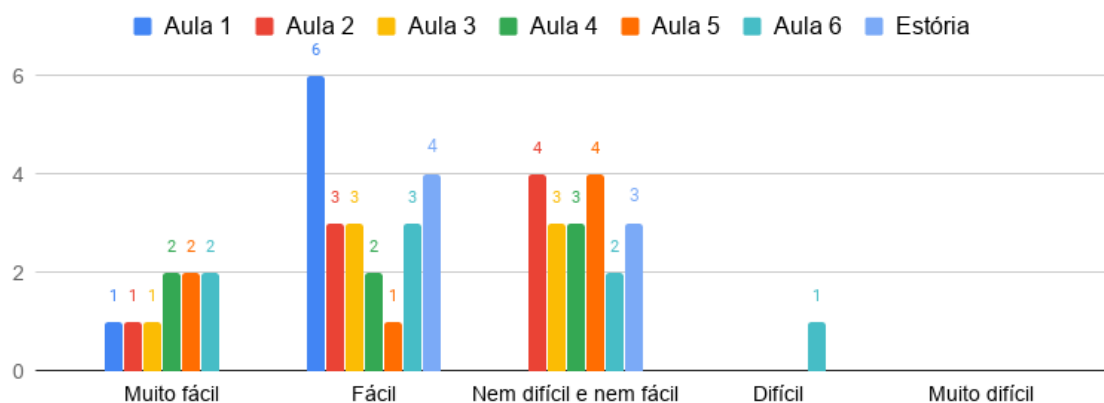


Figura 6.13: Estudo III: Dificuldade das Aulas

assuntos que os alunos já tinham aprendido no GTD de PC I 2019/1. Além disso, as tarefas dadas nesta aula também tinham sido mais simples.

Poucos alunos consideraram as aulas como muito fáceis. As aulas 1, 2 e 3 foram consideradas como muito fáceis por apenas um dos alunos que responderam ao questionário e as aulas 4, 5 e 6 por apenas dois alunos. A maioria dos alunos consideraram as aulas ou como simplesmente fáceis ou como nem difíceis e nem fáceis.

Não houve respostas para a opção *Muito Difícil* do questionário para nenhuma das aulas avaliadas e apenas a sexta aula, onde os alunos fizeram modificações nos projetos anteriores, chegou a ser considerada difícil, porém por apenas um aluno.

Comentários relativos à dificuldade das aulas não emergiram nas entrevistas. As únicas exceções foram os exemplos apresentados anteriormente sobre a atividade de revisão e de criação da estória. Contudo, questionamos os alunos sobre a dificuldade do GTD de PC II em relação ao GTD de PC I. O participante G1P3 achou que o GTD de PC II ficou mais fácil e, conseqüentemente, melhor que o GTD de PC I, uma vez que ele já tinha experiência com a lógica de programação e o Scratch. G1P1 também considerou o GTD de PC II mais fácil que o anterior.

G1P3: Esse GTD foi melhor.

Pesquisador: Tem algum motivo específico?

Eu tenho, só que eu não sei explicar.

Pesquisador: Tenta pensar aí. Pode pensar...

G1P3: Tá... (pensando)... ah, por causa que... tipo... no semestre passado eu era... eu não sabia de nada. Aí, era mais difícil pra mim. Mas agora, como eu sei mais, aí pra mim é mais fácil. Aí pra mim torna mais fácil o GTD. Acho mais legal.

Pesquisador: Você achou que esse ficou mais fácil porque você já tinha aprendido no semestre passado...

G1P3: Sim

* * *

Pesquisador: Ah sim. E a dificuldade? Ficou mais difícil desta vez?

G1P1: Não, ficou mais fácil desta vez.

(Sobre a dificuldade dos GTDs)

G1P12, ao contrário, considerou as atividades do GTD de PC II mais difíceis que aquelas feitas no GTD de PC I. Ele inclusive comentou com amigos que o GTD de PC II tinha ficado muito mais complexo.

Pesquisador: E dificuldade? Foi mais difícil?

G1P12: Sim. Ficou mais difícil.

* * *

Pesquisador: E se você tivesse que... você chegou a falar deste GTD com os seus amigos? Desta continuação? Você falou com alguém?

G1P12: Sim.

Pesquisador: E o que você falou com eles sobre esse GTD?

G1P12: Que ficou bem mais difícil.

(G1P12 sobre o GTD de PC II em relação ao GTD de PC I)

Houve também comentários apontando que apesar do GTD de PC II ter ficado mais complexo, a diferença de dificuldade entre ele e o GTD anterior não era muito grande. Isto foi indicado pelos participantes G1P14 e G1P15.

G1P14: Esse aqui ficou mais difícil porque a gente tá aprendendo coisas novas... aí... mas quando tava no semestre anterior tinha coisa que eu achava difícil também porque eu não sabia.

Pesquisador: Mas em geral esse ficou mais difícil, você acha?

G1P14: Sim

* * *

Pesquisador: Você achou que esse ficou mais difícil?

G1P15: Não, não muito.

Pesquisador: Não?

G1P15: Mas só algumas coisas.

Pesquisador: Quais? O que você achou?

G1P15: A árvore decisão, no início eu também não entendi como fazer... e só.

(G1P15 sobre os GTDs de PC I e PC II)

Pontos observados pelos pesquisadores no decorrer do estudo

A sétima aula que foi dedicada exclusivamente para que os alunos definissem o que eles fariam em seus projetos individuais foi bastante improdutivo. A nossa intenção era de permitir que os alunos explorassem projetos compartilhados no site do Scratch e buscassem inspirações para os seus próprios projetos. No entanto, vários deles utilizaram

esta oportunidade apenas para fins de entretenimento e, no fim, tiveram dificuldades em decidir o que fazer em seus projetos individuais. Apenas dois alunos decidiram por conta própria o que fazer no Scratch: G1P1, que criou um jogo de perguntas sobre um seriado americano e G1P15, que decidiu continuar desenvolvendo o que projeto que ele iniciou por conta própria no primeiro semestre de 2019. Depois de aproximadamente 30 minutos tive que sentar com os alunos e ajudá-los a definir pessoalmente o que os seus projetos seriam.

No que diz respeito às aulas alocadas para o desenvolvimento dos projetos individuais, foi notado que os alunos ficaram bastante desinteressados em desenvolvê-los. Na Aula 8 (primeira aula de desenvolvimento), todos os alunos demonstraram estar interessados em criar os seus projetos e a pedir ajuda para implementar as funcionalidades que eles queriam. No entanto, notamos que o interesse deles em continuar os seus projetos caiu bastante nas Aulas 9 e 10, onde foi observado que os alunos adicionaram poucas modificações em seus projetos em comparação ao que tinha sido planejado a priori. Um dos alunos inclusive nos disse diretamente que não queria mais mexer em seu projeto. Além disso, os alunos ainda estavam tendo muitas dificuldades em realizar operações no Scratch envolvendo os blocos de repetição e condicionais, incluindo manipulações similares àquelas que eles já tinham feito em projetos anteriores, indicando que eles não as compreenderam realmente ou simplesmente esqueceram como utilizar aqueles tipos de blocos.

6.5.4 Discussões e Lições Aprendidas

Nesta subseção analisamos e discutimos os resultados da experiência do GTD de PC II 2019/2 e as lições aprendidas ao conduzi-lo.

O GTD de PC II foi satisfatório para os alunos

Os jogos de perguntas e respostas (i.e.: jogo de perguntas sobre temas de interesse dos alunos, jogo de perguntas de matemática) receberam apenas comentários positivos nas entrevistas. Cinco alunos os mencionaram como uns dos projetos que eles mais gostaram de ter criado no GTD de PC II. Destes alunos, dois mencionaram que eles acharam os jogos de perguntas e respostas legais, pois estes jogos, além de serem divertidos, podem contribuir para o aprendizado e serem utilizados para testar o conhecimento daqueles que os estão jogando. Vemos que o aprendizado ao jogar, tema que surgiu nos grupos focais dos GTDs de PC I quando conversamos com os alunos sobre a plataforma Kahoot!, também surgiu nas entrevistas quando conversamos com alguns alunos sobre as aulas em que eles criaram os jogos de perguntas. Eles conside-

raram não apenas o entretenimento, mas também o valor educacional que estes tipos de projetos oferecem às pessoas como aspectos positivos destes tipos de projetos. Já os outros três alunos, que disseram ter gostado dos jogos de perguntas e respostas, não chegaram a mencionar os motivos pelos quais eles gostaram destes projetos. Um deles simplesmente afirmou ter gostado do bloco de perguntas, que era um bloco que ele não tinha aprendido no semestre anterior e os outros dois não explicitaram a razão pela qual eles gostaram destes jogos. Além disso, os projetos de perguntas e respostas serviram de inspiração para um dos alunos (G1P1), que decidiu utilizar o que ela aprendeu ao criar estes jogos em seu projeto individual, que também foi um jogo de perguntas e respostas sobre um tema de interesse para ela. Considerando os resultados positivos obtidos pela introdução dos jogos de perguntas e respostas no GTD, acreditamos que estes jogos de perguntas e respostas sejam alternativas viáveis para que os alunos pratiquem diferentes formas de utilização das estruturas condicionais, variáveis e operadores matemáticos (e.g.: somar, subtrair, multiplicar, dividir) e de comparação (e.g.: igual, maior que, menor que). Devido aos resultados extremamente obtidos dos jogos de perguntas e respostas no GTD de PC II, achamos interessante incluí-los na sequência didática final do trabalho.

Dentre as atividades conduzidas no GTD de PC II, os projetos de revisão do Scratch, criado pelos alunos nas duas primeiras aulas, e da estória, criado pelos alunos nas aulas finais, foram os únicos que receberam algum tipo de comentário negativo nas entrevistas. O **projeto de revisão** recebeu críticas negativas por dois alunos, sendo que um deles (G1P15) indicou que gostou menos das aulas de revisão pela dificuldade; e o outro (G1P6), pelo projeto ser entediante. A dificuldade inicial que G1P15 teve para realizar as tarefas propostas para a aula de revisão o levaram a ficar frustrado e irritado, sendo que estas são reações naturais que podem se manifestar quando indivíduos estão enfrentando determinados obstáculos e barreiras para alcançar determinado fim. No caso de G1P6, o fato do projeto ser “sem graça” foi algo que o levou a gostar menos das duas primeiras aulas de revisão. Além de G1P6, também foi observado que G1P12 e G1P15 pareceram estar desinteressados na atividade, pois eles tentaram fazer coisas diferentes do que foi proposto na folha da atividade. Isto pode estar ligado a dificuldade em concluir as tarefas, como no caso de G1P15, ou também pela ausência de um objetivo concreto tal como a criação de um jogo, de uma animação ou de uma estória interativa. Diferentemente dos outros projetos criados pelos alunos no GTD de PC I 2019/1 e GTD de PC II 2019/2, as tarefas que os alunos tinham que resolver nas aulas de revisão eram independentes entre si, ou seja, a conclusão delas não resultavam em um projeto funcional. Já o **projeto da estória**, recebeu comentários negativos apenas do aluno G1P8, que considerou a atividade difícil e muito chata. Inclusive,

a única sugestão de remoção de algum dos projetos do GTD de PC II veio deste mesmo aluno, que sugeriu que retirássemos o projeto da estória do GTD de PC II caso fôssemos conduzi-lo mais vezes no futuro. Embora o projeto da estória tenha recebido críticas negativas por G1P8, ele recebeu críticas positivas do restante dos alunos. A árvore de decisão e a sua utilização para definição e estruturação das estórias que os alunos queriam criar foram interessantes para os alunos que gostaram de ter criado uma estória interativa no Scratch. Desta forma, consideramos que a introdução das árvores de decisões aos alunos de maneira simplificada e a inclusão do projeto da estória para os alunos para aplicar na prática as árvores de decisão foram um sucesso.

Além das críticas negativas direcionadas às atividades de revisão e criação de estórias discutidas previamente, também surgiram nas entrevistas comentários relativos ao pequeno número de kahoots conduzidos no GTD de PC II. Alguns alunos chegaram até a expressar que preferiram o GTD de PC I ao GTD de PC II porque no semestre anterior nós tínhamos conduzido um maior número de kahoots. Para fins de comparação, cinco kahoots foram realizados no GTD de PC I 2019/1; já no GTD de PC II 2019/2, apenas dois. Desta forma, estes alunos sugeriram que utilizássemos mais o Kahoot em edições futuras do GTD. Da perspectiva dos alunos, que gostaram bastante dos kahoots conduzidos no GTD de PC I, estas críticas são válidas e fazem sentidos, porém no GTD de PC II não tínhamos muitos conteúdos e diferentes tipos de ferramentas a serem revisados com os alunos, uma vez que utilizamos apenas o Scratch nesta segunda etapa. Ademais, queríamos que os alunos tivessem tempo o suficiente para finalizar as tarefas propostas para as aulas e a condução dos kahoots faria com que os alunos tivessem menos tempo dedicado à construção dos projetos no Scratch.

Conforme discutido previamente, o GTD de PC II teve pontos negativos para alguns alunos, mas a maioria deles afirmou que gostou de todas as atividades realizadas no GTD. Apesar das críticas negativas, todos os oito alunos afirmaram nas entrevistas que gostaram de ter continuado participando no segundo semestre. Isso foi refletido no baixo número de sugestões dadas nas entrevistas para melhora do GTD. Todos alunos também comentaram que recomendariam que seus colegas participando do GTD de PC I 2019/2 participassem também do GTD de PC II caso eles tivessem a oportunidade. Todos estes pontos apresentados são indícios de que o GTD de PC II, em geral, foi uma experiência satisfatória para todos os oito alunos participantes.

Projetos Individuais não funcionaram bem

Conforme apresentamos previamente, as aulas alocadas para que os alunos pudessem decidir o que fazer em seus projetos individuais do Scratch e para que eles pudessem

desenvolvê-los não ocorressem da forma que queríamos e acabaram se tornando um aspecto negativo deste GTD de PC II do nosso ponto de vista, professores ministrando o GTD.

Primeiramente, no que diz respeito a Aula 7 (i.e.: aula de definição de projetos), diversos fatores a afetaram negativamente. O principal deles foi a desorganização na forma em que ela ocorreu. Esta aula aconteceu de forma muito desestruturada, pois foi deixado que os alunos testassem projetos e jogos livremente na plataforma. Além disso, eles não foram avisados previamente de que teriam que criar um projeto próprio no Scratch. Poderíamos tê-los informado com algumas semanas de antecedência sobre os projetos individuais para que eles tivessem tempo o suficiente para pensar em algumas opções de projetos para eles criarem no Scratch e para que eles pudessem discuti-las conosco na aula de definição de projetos.

Nas aulas de desenvolvimento, os alunos não tiveram à sua disposição roteiros com tarefas a serem seguidas. A ausência destes roteiros das atividades, que os alunos vieram utilizando para criar os projetos no Scratch desde o GTD de PC I 2019/1, pode tê-los levado a ficarem desinteressados, pois eles não tinham nenhum direcionamento do que fazer. Esta poderia ter sido uma boa oportunidade para a reintrodução e prática da decomposição de problemas. Visto que os alunos desenvolveriam projetos individuais, muitas das coisas que eles deveriam implementar no Scratch seriam diferentes e provavelmente exclusivas de seus projetos. Portanto, na mesma aula de definição de projetos, seria interessante pedir que eles próprios definissem o roteiro das tarefas que eles teriam que seguir. Desta forma, eles fariam a decomposição do projeto a ser desenvolvido em sub-tarefas, que depois de completadas, resultariam no projeto final.

Dado que os alunos ainda estavam com problemas para programar tarefas relativamente simples no Scratch, isto nos leva a acreditar que eles não estavam de fato preparados tanto em questão de conhecimento da ferramenta quanto em questão de maturidade para conceber e desenvolver seus próprios projetos no Scratch. Portanto, essas aulas poderiam ter sido utilizadas na criação de projetos mais específicos e simples tais como os jogos de perguntas e respostas e o projeto da estória. Assim, eles poderiam continuar praticando diferentes formas de utilizar e combinar os blocos de comandos no Scratch.

Apesar destes pontos negativos no que diz respeito à didática, preparação e condução das aulas reservadas para os projetos individuais dos alunos, nenhum deles expressou nas entrevistas que eles não gostaram de criar os seus próprios projetos. Todos os alunos que mencionaram seus próprios projetos, o fizeram de forma positiva, apontando que os acharam divertidos e que gostaram de ter tido a oportunidade de criá-los.

GTD de PC I vs GTD de PC II

Uma grande diferença e talvez a diferença mais notável para os alunos participantes entre o GTD de PC I 2019/1 e o GTD de PC 2019/2 é o fato de que utilizamos diferentes tipos de ferramentas e atividades no GTD de PC I e apenas uma ferramenta no GTD de PC II. Pelas entrevistas conduzidas, notamos que este foi um fator de grande influência na opinião dos alunos sobre os GTDs de Pensamento Computacional. Houve alunos que preferiram o GTD de PC II porque eles gostaram de programar e criar projetos no Scratch e houve alunos que preferiram o GTD de PC I porque eles gostaram de ter tido a oportunidade de utilizar diferentes tipos de ferramentas e fazer diferentes tipos de atividade. A dificuldade dos GTDs para os alunos também está relacionada às diferentes abordagens adotadas em cada um deles. Dado que utilizamos apenas o Scratch durante todo o GTD de PC II, as atividades conduzidas nele foram relativamente mais complexas que as atividades do GTD de PC I, onde foram empregadas ferramentas associadas às primeiras Metáforas de Perspectivas Culturais. A dificuldade, porém, não foi um fator que impactou negativamente a opinião dos alunos sobre o GTD de PC II segundo o que foi dito nos grupos focais.

Conforme discutido no Capítulo 4, nos fundamentamos nas MPCs para selecionar diferentes recursos didáticos para que aspectos da Computação fossem apresentados aos alunos de forma gradual. Assim, o grau da complexidade das ferramentas e atividades introduzidas no início do GTD e o conhecimento necessários para utilizá-las e conduzi-las são relativamente menores que os das ferramentas introduzidas no final. Ademais, a diversidade das ferramentas permite que os alunos pratiquem as habilidades de PC em diferentes contextos e de diferentes maneiras. Esta abordagem, porém, leva a um maior gasto de tempo, especialmente quando é feita a transição para ferramentas de maior complexidade como o Scratch, pois os alunos devem aprender constantemente a utilizar novos tipos de ferramentas. Portanto, dependendo da carga horária dos cursos, há a possibilidade de que os alunos não terão tempo o suficiente para desenvolverem por conta própria projetos mais complexos e interessantes no Scratch como no caso dos GTDs de PC I. Em contrapartida, a abordagem de utilização exclusiva do Scratch, como foi feito no GTD de PC II, elimina a necessidade dos alunos terem que aprender diferentes ferramentas e que diferentes tipos de projetos sejam criados por eles porque eles estarão trabalhando no Scratch desde o princípio. Os tópicos da Computação introduzidos por meio de diferentes ferramentas e dinâmicas no GTD de PC I também podem ser introduzidos através de exemplos e atividades simples no Scratch e os alunos os aprenderiam já no contexto do Scratch.

Outro ponto a ser discutido no que tange às abordagens dos GTDs de PC I

e PC II é o quão engajante elas são para os alunos. O uso constante de uma única ferramenta durante todo o curso pode tornar a experiência monótona e entediante para os alunos participantes, especialmente se a impressão que eles tiverem da ferramenta for negativa, o que tornaria a experiência como um todo ruim para eles. Este não foi o caso do GTD de PC II visto que a maioria dos alunos indicou nas entrevistas ter achado a abordagem do GTD de PC II melhor que a abordagem do GTD de PC I. É importante destacar que os alunos que continuaram participando no segundo semestre tinham consciência de que utilizaríamos apenas o Scratch no GTD de PC II e, conforme visto no GTD de PC I, a opinião da maioria destes alunos já era positiva em relação ao Scratch. Consequentemente, estes resultados não são inesperados. Por outro lado, a abordagem de utilização de múltiplas ferramentas do GTD de PC I proporciona uma experiência nova aos alunos sempre que ferramentas e atividades diferentes são introduzidas. Assim, se porventura os alunos não gostarem de determinada ferramenta, a experiência como um todo pode não ser negativa para eles.

6.6 Consolidação da Sequência Didática

Neste capítulo apresentamos uma pesquisa-ação realizada no Centro Pedagógico da UFMG com o objetivo de criarmos uma sequência didática para desenvolver as habilidades do Pensamento Computacional através do ensino da Computação e programação. Múltiplos recursos didáticos foram selecionados para serem aplicados em cursos de PC no Centro Pedagógico da UFMG. Utilizamos as Metáforas de Perspectivas Culturais como fundamento para definirmos quais recursos didáticos utilizaríamos conforme apresentadas no Capítulo 5. Eles foram aplicados em três estudos (i.e.: Estudo I, Estudo II e Estudo III) em salas de aula com alunos do Ensino Fundamental II. A avaliação foi feita com base na percepção dos alunos que as utilizaram e observações dos professores. Os Estudos I e II aconteceram por meio do GTD de PC I, que foi um curso planejado para o ensino da Computação e PC empregando diferentes recursos para o ensino da programação. O Estudo II, especificamente, foi um aprimoramento da primeira versão conduzida, onde realizamos alterações para mitigarmos os pontos negativos do Estudo I. Por outro lado, o Estudo III foi uma continuação do primeiro estudo e teve como foco a utilização do Scratch para abordar tópicos e atividades mais complexas.

A partir dos resultados obtidos nos três estudos, nós desenvolvemos o *Curso de Introdução à Computação e ao Pensamento Computacional*, que é uma sequência didática para ser aplicada com alunos do Ensino Fundamental II para introduzi-los à Computação e desenvolver as suas habilidades do PC por meio da prática do pensa-

mento algorítmico. A maioria das aulas incluídas no curso estavam presentes nos dois primeiros estudos. Dentre os projetos conduzidos no Estudo III, selecionamos os jogos de perguntas e respostas, os quais foram recebidos positivamente pelos alunos, para fazerem parte do curso. Nos nossos estudos, o LightBot foi um dos recursos que obteve os melhores resultados. No entanto, a versão livre LightBot, disponibilizada como a iniciativa Hora do Código, apenas pode ser executada por meio do *plugin* Adobe Flash Player. Contudo, este *plugin* foi descontinuado em 2021 por diversos motivos, fazendo com que as aplicações que precisavam dele parassem de funcionar⁴⁴. Não temos informações se uma versão alternativa gratuita será disponibilizada pelos responsáveis pelo LightBot. Assim, embora o tenhamos deixado na sequência didática, os educadores que a quiserem aplicar, devem verificar se existe uma versão do LightBot que esteja disponível para usarem. De toda forma, deixamos indicado na sequência didática outros recursos que foram utilizados e que podem endereçar os aspectos tratados pelo LightBot.

A Tabela 6.4 apresenta a visão geral das aulas alocadas para o curso, apontando o conteúdo abordado, ferramentas e uma breve descrição de cada aula. A princípio, criamos um total de 15 aulas semanais, incluindo os projetos criados pelos alunos nos GTDs de PC I e no GTD de PC II. As 12 primeiras aulas são as aulas principais da sequência, representando o GTD de PC I. As três aulas finais (indicadas com o fundo cinza), destacadas em cinza na tabela, foram aulas envolvendo projetos do GTD de PC II, os quais são mais complexos. Elas também podem ser conduzidas por professores utilizando a sequência dependendo da maturidade da turma em relação ao Scratch e da disponibilidade de aulas. As aulas estão previstas para durar 1h e 20min em média, totalizando 20 horas no total. A sequência didática para professores com a descrição completa das aulas, atividades, ferramentas e materiais pode ser encontrada no Apêndice A. Note que o número de aulas da sequência não é a soma do número de aulas dos GTDs de PC I e do GTD de PC II. Houve aulas do GTDs de PC I que foram reestruturadas, removidas e conduzidas devido a imprevistos (i.e.: Bugs no Ciclo da Água). No caso do GTD de PC II, houve aulas de revisão para relembrar os alunos sobre o Scratch após o recesso escolar no meio do ano e a inclusão de aulas na parte principal da sequência (i.e.: jogo de perguntas e respostas). Além disso, as aulas finais dos GTDs tiveram caráter exploratório e livre, não sendo dedicadas a apresentação de tópicos e criação de projetos.

⁴⁴Para mais informações: <https://www.adobe.com/products/flashplayer/end-of-life.html>

Tabela 6.4: Visão geral das aulas do curso de PC

<i>Aulas</i>	<i>Conteúdo</i>	<i>Objetivo</i>	<i>Recursos e Atividades</i>
Aula 1	Raciocínio Lógico	[VD] Resolver desafios de lógica e registrar suas soluções	Resolução dos desafios do nível iniciante do Estacionamento Algorítmico . Além da resolução, os alunos devem fazer o registro dos passos para resolver os desafios e testá-los.
Aula 2	Algoritmos e Decomposição	[OD] Aprender o conceito de algoritmos e decompor tarefas do cotidiano. [VTD] Criar algoritmos para atividades simples	Apresentação do conceito de algoritmos aos alunos fazendo alusão às soluções registradas por eles na aula anterior. Condução da atividade Decomposição da Turma da Mônica e apresentação do vídeo ilustrando a importância da decomposição. Condução do Programando em Papel . Condução do LightBot
Aula 3	Laços de Repetição	[OD] Aprender o conceito de Laços de Repetição [VTD] Praticar a utilização dos Laços de Repetição	Revisão do conteúdo e atividades conduzidas na Aula 2 utilizando a ferramenta Kahoot!. Apresentação dos laços de repetição e Code.org . Resolução dos desafios da lição <i>Labirinto: Ciclos</i> do Curso 2 do Code.org. A lição requer que os alunos identifiquem ações ou grupo de ações que devem ser repetidas utilizando um comando de repetição
Aula 4	Laços de Repetição	[VTD] Praticar a utilização dos <i>loops</i>	Explicação teórica sobre a identificação de ações se repetindo e apresentação de algumas dicas que os alunos podem utilizar para resolver os desafios da lição. Finalização dos desafios da lição. Disponibilização das lições <i>Artista: Ciclos</i> e <i>Abelha: Ciclos</i> aos alunos que finalizarem os desafios do <i>Labirinto: Ciclos</i>
Aula 5	Ambiente de LPV	[EcT] Aprender sobre o ambiente de programação visual [OD] Aprender sobre o Plano Cartesiano se necessário	Revisão dos conteúdos abordados até a quinta aula utilizando o Kahoot!. Introdução ao Plano Cartesiano. Apresentação da ferramenta Scratch aos alunos: a ferramenta, sua finalidade e os principais blocos de movimento, aparência e evento
Aula 6	Projeto Introdutório	[EcT] Desenvolver projeto introdutório para praticar o que foi aprendido na aula anterior	Revisão do Plano Cartesiano e Scratch. Condução da atividade <i>Anime seu Nome</i> no Scratch , onde os alunos devem animar as letras dos seus nomes ou apelidos. Condução de uma votação para decidir os projetos mais interessantes no final da aula caso haja tempo disponível
Aula 7	Projeto Simples	[EcT] Desenvolver projeto simples	Apresentação e votação dos melhores projetos do <i>Anime seu Nome</i> caso não tenha sido possível realizá-las no fim da aula anterior. Desenvolvimento de um jogo simples de clicar no Scratch . No jogo, um personagem fica aparecendo e desaparecendo regularmente no cenário e o jogador deve clicar no personagem
Aula 8	Estruturas Condicionais	[OD] Aprender sobre condicionais [EcT] Desenvolver projeto utilizando condicionais	Apresentação do conceito de condicionais. Condução da <i>Dinâmica das Palmas</i> onde os alunos devem verificar se o número apresentado é maior que 5 e bater um determinado número de palmas conforme o resultado verificação. Implementação no Scratch do Jogo da Coleta, jogo simples onde o jogador deve controlar um personagem para coletar objetos no cenário

Aula 9	Estruturas Condicionais	[EcT] Desenvolver projeto utilizando outros tipos de condicionais. Ele pode ser uma extensão dos projetos anteriores se possível	Continuação à prática das condicionais no Scratch através do desenvolvimento do jogo Pong, onde o jogador deve movimentar uma raquete para impedir que a bola caia no chão
Aula 10	Variáveis	[OD] Aprender sobre variáveis [EcT] Desenvolver projeto utilizando variáveis. Ele pode ser uma extensão dos projetos anteriores caso variáveis possam ser incluídas	Apresentação do conceito de variáveis e o seu propósito. A utilização das variáveis é colocada em prática através da implementação de sistemas de pontuação para o Jogo de Clicar, Jogo da Coleta e Pong no Scratch
Aula 11	Especialização: Scratch	[OD] Aprender sobre os sensores <i>pergunte e resposta</i> [EcT] Desenvolver projeto para praticar a utilização dos sensores <i>pergunte e resposta</i>	Introdução aos blocos <i>pergunte e resposta</i> . Desenvolvimento de um Jogo de Perguntas e Respostas simples no Scratch . As perguntas e respostas são específicas, sendo sempre as mesmas. Um sistema de pontuação pode ser adicionado caso os alunos consigam terminar o desenvolvimento antes da aula terminar
Aula 12	Especialização: Scratch	[EcT] Desenvolver projeto para praticar a utilização dos sensores <i>pergunte e resposta</i>	Desenvolvimento de um Jogo de Perguntas de Matemática para prática de tabuada no Scratch . Neste jogo, as perguntas são a multiplicação de dois números aleatórios, requerendo a utilização de variáveis para armazenar os números a serem multiplicados e fazer o processo de verificação das respostas
Aula 13	Especialização: Scratch	[OD] Aprender sobre <i>clones</i> [EcT] Desenvolver projeto para praticar a utilização dos clones	Introdução aos comandos para gerenciar clones no Scratch . Desenvolvimento de um jogo de desviar no Scratch . Neste jogo, o personagem deve evitar objetos caindo do céu
Aula 14	Árvores de Decisão	[OD] Aprender sobre árvores de decisão e criar uma árvore de decisão simplificada [EcT] Desenvolver projeto baseado na árvore de decisão	Apresentação do conceito de árvores de decisão. Criação de árvores de decisão para guiar o fluxo de uma história com múltiplos fins. Implementação da história no Scratch
Aula 15	Árvores de Decisão	[EcT] Desenvolver projeto baseado na árvore de decisão	Finalização da implementação da história no Scratch iniciada na aula anterior. Modificação e extensão da história implementada


Embora tenhamos definido as aulas, indicando especificamente os recursos didáticos a serem utilizados pelos professores que cogitam em conduzir o curso em sua totalidade ou parcialmente, recursos didáticos com propósitos e dificuldades similares podem ser incluídos ou substituir aqueles selecionados por nós. Para tal, os professores podem fazer o uso da classificação dos recursos à luz das Metáforas de Perspectivas Culturais conduzida por nós [Oliveira, 2020] a fim de buscar por recursos equivalentes àqueles utilizados no *Curso de Introdução à Computação e ao Pensamento Computacional*. Há também a possibilidade de recursos não classificados por nós a princípio serem utilizados, similarmente a como fizemos com a atividade Estacionamento Algorítmico, a qual não tinha sido utilizada nos estudos analisados por nós. Contudo, seria interessante que se considere a que MPC cada um destes recursos está associado, para


que os professores decidam como utilizar estes recursos para que a Computação e PC ainda sejam introduzidos gradativamente.

Em relação à utilização de recursos didáticos diferentes, existem muitas atividades, além do Estacionamento Algorítmico, que podem ser conduzidas na primeira aula para que os alunos exercitem o raciocínio lógico e sejam expostos superficialmente ao sequenciamento de ações. Como apresentado no Capítulo 5, a plataforma Racha-Cuca⁴⁵ contém diversos desafios de lógica que podem ser utilizados de tal forma que os alunos pratiquem indiretamente o sequenciamento de ações, bastando apenas que comandos para as ações nos jogos sejam definidos para que eles indiquem a sequência de ações a serem executadas para que os desafios sejam resolvidos. [A utilização de recursos lúdicos disponíveis nas escolas ou a serem adquiridos por elas é outra opção a ser considerada para as aulas iniciais.](#)

Outro exemplo de recurso que poderia ser interessante considerar é o AppInventor⁴⁶. Nos grupos focais, alguns alunos apontaram que eles estavam mais familiarizados com dispositivos móveis. Acreditamos que este era o caso para vários alunos que participaram dos GTDs devido a dificuldade que eles demonstraram em ter ao utilizar computadores de mesa. Logo, a utilização da ferramenta AppInventor no lugar do Scratch para que os alunos desenvolvam e testem aplicativos funcionais em seus próprios *smartphones* pode ser uma abordagem engajadora e que teria bons resultados.

A sequência didática pode também ser expandida através de inclusão de aulas completamente novas ou por meios a divisão das aulas originais em múltiplas aulas caso seja necessário. Por exemplo, uma aula pode ser adicionada para os alunos explorarem o sequenciamento de ações plugado antes dos laços de repetição serem introduzidos. Isso poderia ser feito, por exemplo, através da inclusão de uma aula focada no *Kidlo Coding* ou na lição *Labirinto: Sequência* disponível no Curso 2 da plataforma Code.org. No caso das aulas conduzidas no Scratch, aulas extras podem ser adicionadas para que os alunos desenvolvam outros tipos de jogos e animações. As aulas originais propostas podem ser divididas em múltiplas aulas. Isso pode ser necessário para as aulas feitas no Scratch para que os alunos consigam finalizar as implementações dos projetos caso eles não consigam terminar no tempo previsto ou implementem as modificações propostas para os projetos originais. Finalmente, caso o professor julgue que os alunos estejam preparados, há a possibilidade de inclusão de aulas para que os alunos definam e implementem projetos individuais. Conforme discutido no Estudo III, recomendamos que o professor faça com que os próprios alunos criem seus roteiros de atividade no Scratch, ao pedir que eles decomponham os projetos que eles pretendem desenvolver em tarefas


⁴⁵Racha-Cuca 

⁴⁶AppInventor 

menores.

Conforme explicamos no Estudo III, as atividades associadas à metáfora Estrangeiro sem Tradutor estavam planejadas para serem realizadas por meio da linguagem de programação Python, visto que esta é uma linguagem de programação textual relativamente fácil de ser lida e entendida. Contudo, ela não foi utilizada devido a disponibilidade limitada de aulas, tempo e imprevistos que ocorreram. Pretendíamos em apresentar a linguagem aos alunos por meio da construção de programas simples envolvendo a impressão de texto, armazenamento de informações digitadas em variáveis e a utilizando condicionais e estruturas de repetições. Um exemplo de programa que queríamos desenvolver junto com os alunos era um programa que verificaria se um número digitado é maior que 5, similarmente à dinâmica que tínhamos feito com os alunos nas aulas em que apresentamos as condicionais. Apesar de termos decidido utilizar o Python para o Estrangeiro sem Tradutor, outras possibilidades podem ser exploradas tais como (i) a utilização de outras linguagens de programação textual (e.g.: C/C++, Java, C#) e (ii) a adoção de ferramentas didáticas para fazer a transição entre linguagens de programação visual e linguagens de programação textual. Um exemplo deste tipo de ferramenta é o Portugol Studio⁴⁷, ambiente de desenvolvimento didático, o qual é especialmente interessante não apenas por permitir que algoritmos sejam desenvolvidos em português, mas também por oferecer feedback e mensagens de erros em português aos usuários. Outro exemplo é a ferramenta TinkerCad⁴⁸, onde usuários podem programar o Arduino por meio de linguagens de programação visuais e textuais, possibilitando que eles vejam a sintaxe em ambas linguagens simultaneamente. Outra possibilidade é a utilização da ferramenta desenvolvida no trabalho de Barata et al. [2017], para que conhecimentos sobre a lógica de programação da linguagem visual Blockly fossem transferidos para *JavaScript*.

Finalmente, em relação ao Kahoot!, vimos que os resultados alcançados no que diz respeito à percepção dos alunos foi bastante positiva. Ele conseguiu cumprir o objetivo que tínhamos em mente ao o selecionarmos para ser utilizado nos cursos. Os GTDs não tinham atividades extra-classe ou avaliações; contudo, por meio do Kahoot!, os alunos puderam lembrar e consolidar conceitos apresentados por nós e as atividades realizadas. Uma vez que o Kahoot! fomenta competições saudáveis e “gamificadas”, os alunos gostaram muito de participar dos kahoots. Ademais, um uso que não fizemos dele, mas que poderia ser interessante, seria analisar os resultados dos kahoots para identificar tópicos e atividades difíceis para os alunos para que eles fossem revistos e trabalhados mais a fundo. Dessa forma, embora o Kahoot! não esteja associado

⁴⁷Portugol Studio 

⁴⁸TinkerCad 

diretamente ao ensino da Computação e PC, o seu uso pode ser benéfico nesses tipos de curso, como visto nos estudos.

Capítulo 7

Conclusão

Este trabalho apresentou uma pesquisa com o objetivo final de criar uma sequência didática para professores introduzirem a Computação e o Pensamento Computacional para alunos do Ensino Fundamental II. Para isso, propomos a adoção das Metáforas de Perspectivas Culturais (MPCs) como fundamento para apresentar o PC a alunos do EF II de maneira gradual, percorrendo o *continuum* das metáforas ao utilizar diferentes recursos didáticos.

Primeiramente, nós utilizamos as MPCs para classificar recursos didáticos adotados no Brasil para o ensino da Computação e Pensamento Computacional. A classificação representa o grau de complexidade dos recursos didáticos no que diz respeito a como eles apresentam a Computação e o PC e o quão difícil seria para que alunos do Ensino Fundamental II sem contato prévio com a Computação conseguisse utilizá-los. Esta classificação pode ser utilizada por professores que cogitam ensinar a Computação para alunos do Ensino Fundamental II. Dado que ela representa a complexidade dos recursos didáticos, professores podem adotá-la para decidir quais recursos didáticos utilizar. Assim, selecionamos recursos didáticos, dentre aqueles classificados, para serem aplicados em salas de aula, partindo dos recursos associados à primeira MPC (Viajante Doméstico) e indo até aqueles associados à quinta MPC (Estrangeiro sem Tradutor). Selecionamos pelo menos um recurso didático para cada MPC. Conduzimos então uma pesquisa-ação no Centro Pedagógico da UFMG para aplicarmos os recursos didáticos em cursos de PC ministrados a alunos do 6º ano do Ensino Fundamental II. A Pesquisa-Ação foi dividida em três estudos, sendo que um curso de PC foi conduzido em cada um deles.

O primeiro estudo ocorreu no primeiro semestre de 2019 com 16 alunos. Neste estudo conduzimos o GTD de PC I 2019/1, curso introdutório à Computação e ao Pensamento Computacional. No curso, utilizamos uma abordagem mista com atividades

desplugadas e plugadas, sendo que a maioria delas foi plugada. O curso e os recursos didáticos adotados foram avaliados por meio das opiniões dos alunos participantes que foram coletadas durante o estudo por questionários e grupos focais e por meio das observações dos professores/pesquisadores que o conduziram. Em geral, os resultados alcançados foram positivos dado que a maioria dos alunos gostou das atividades e de ter participado do curso. Contudo, observamos que vários aspectos do curso poderiam ser melhorados ao fazer alterações na abordagem adotada em certas aulas e nas atividades realizadas. Portanto, revisamos o GTD de PC I e o ministramos novamente no segundo semestre de 2019 para outros 11 participantes que não tiveram a oportunidade de participar do GTD de PC I no primeiro semestre. Este foi o segundo estudo da nossa pesquisa-ação. Adotamos a mesma abordagem utilizada no primeiro estudo para avaliarmos o GTD de PC I 2019/2. Os resultados do GTD de PC I 2019/2 foram melhores que o GTD de PC I 2019/1. Os alunos participantes tiveram menos críticas negativas e nós, como pesquisadores, observamos menos pontos a serem melhorados. Outro estudo foi realizado no segundo semestre de 2019 dado que os alunos participantes do primeiro estudo demonstraram interesse em continuar aprendendo mais sobre a Computação. Então, decidimos conduzir um terceiro estudo no segundo semestre de 2019 focado no Scratch para estes alunos.

Nos fundamentamos nos resultados das aulas e atividades dos três GTDs ministrados para criarmos o *Curso de Introdução à Computação e ao Pensamento Computacional*, uma sequência didática para professores interessados no ensino da Computação e PC. O curso é composto por 15 aulas de aproximadamente 1h e 20min, que abordam aspectos teóricos e práticos da Computação, sendo que a prática é o foco principal de todas as aulas. Os aspectos Computação que são introduzidos e praticados através do curso são: algoritmos e sequenciamento de ações, laços de repetição, condicionais e variáveis. A sequência detalha todas as aulas, indicando os objetivos de cada uma delas, materiais a serem utilizados, explicações a serem dadas e as atividades a serem conduzidas com e pelos os alunos. Nós esperamos que o Curso de Introdução à Computação e ao Pensamento Computacional sirva de apoio para professores que buscam conduzir atividades para o desenvolvimento do Pensamento Computacional em alunos do Ensino Fundamental II.

7.1 Contribuições

Nosso trabalho teve contribuições tanto teóricas, quanto práticas. Primeiramente, em relação as contribuições teóricas, fizemos um **uso original das MPCs**. As metáforas

foram originalmente propostas para apoiar o designer a refletir sobre como melhor apresentar uma cultura estrangeira em um sistema interativo (de acordo com seus objetivos para o sistema). Como elas representam um *continuum* de imersão cultura, resolvemos (i) considerar o PC uma cultura estrangeira e (ii) usar o *continuum* das metáforas como base para propor uma apresentação gradual desta “cultura” para os alunos. Esta nova utilização **contribui para a área de IHC**, visto que uma ferramenta epistêmica da área foi apropriada para pensar não apenas em design de interação, mas no caso no design de um curso. O trabalho também **contribui para o Ensino de Computação**, pois geramos uma proposta, fundamentada nas MPCs, que pode guiar o ensino gradativo de um conteúdo que possa ser abordado como uma cultura como no caso de PC.

Em relação às contribuições práticas, o **Curso de Introdução à Computação e Pensamento Computacional** gerado por nós, [juntamente com os recursos disponibilizados](#), podem ser utilizados como proposto no Apêndice A por outros educadores cogitando em ensinar a Computação e o PC a alunos do Ensino Fundamental II com pouca ou nenhuma experiência prévia. A **classificação dos recursos didáticos** existentes comumente empregados no Brasil feito em parceria com Oliveira [2020] pode ajudar professores ministrando o **Curso de Introdução à Computação e Pensamento Computacional** a selecionar outros recursos de interesse de acordo com a metáfora representada pela aula em questão. Professores também podem utilizar a classificação para definir outros tipos de sequências didáticas, que podem ser mais ou menos complexas dependendo das metáforas que eles pretendem incluir nessas sequências. Por exemplo, uma introdução ao PC pode ser feita através de oficinas mais curtas incluindo apenas as aulas iniciais (e.g.: Aulas de 1 a 4). Além disso, fizemos as **descrições das aulas lecionadas** nos três estudos, onde indicamos os recursos didáticos e as atividades conduzidas em cada uma delas. Desta forma, professores podem replicar a mesma experiência com os seus alunos caso eles tenham interesse. Os **resultados e discussão das lições aprendidas com cada uma das experiências**, apresentando a perspectiva de alunos e dos professores/pesquisadores, pode apoiar professores e ajudá-los a entender como as abordagens e recursos didáticos adotados nos estudos podem afetar alunos da mesma faixa etária dos 27 alunos que participaram da nossa pesquisa.

7.2 Limitações

Os três estudos conduzidos possuíram algumas limitações de escopo. Primeiramente, nós não avaliamos o potencial de ensino das ferramentas e atividades aplicadas, ou seja, embora tenhamos conduzido revisões através dos kahoots no decorrer das intervenções, que de certa forma avaliaram se os alunos lembravam os conceitos apresentados e atividades realizadas, nós não avaliamos sistematicamente o conhecimento e habilidades adquiridas pelos alunos através do curso. Ademais, o nosso trabalho foi qualitativo, logo, embora resultados positivos tenham sido alcançados com a condução dos cursos para os 27 alunos do Ensino Fundamental II, não podemos afirmar que os mesmos resultados seriam obtidos caso ele seja conduzido em outros contextos ou com outros grupos de alunos. Finalmente, um dos nossos objetivos era avaliar uma experiência de ensino contendo todas as cinco MPCs, utilizando atividades desde aquelas que não abordam a Computação diretamente (i.e.: Viajante Doméstico) até aquelas que permitem que os alunos conduzam tarefas similares àsquelas realizadas por pessoas que trabalham com a Computação (i.e.: Estrangeiro sem Tradutor). Desta forma, os alunos poderiam passar por todas as etapas do continuum de exposição cultural. Contudo, não foi possível que a quinta e última metáfora (i.e.: Estrangeiro sem Tradutor) fosse explorada com eles por meio da utilização de linguagens de programação convencionais devido a limitação de tempo.

7.3 Trabalhos Futuros

Os trabalhos futuros desta pesquisa envolvem aplicações do *Curso de Introdução e Pensamento Computacional* descrito no Apêndice A, seguida de sua adaptação e aprimoramento baseado nos resultados obtidos com estas aplicações. Como mencionado previamente, nosso trabalho não avaliou sistematicamente as intervenções no que diz respeito a quanto os alunos aprenderam e desenvolveram as habilidades do PC com as aulas e atividades. Em vista disso, estudos focados neste aspecto devem ser conduzidos no futuro.

Apenas um pequeno número de ferramentas, dentre aquelas classificadas por nós à luz das MPC, foram aplicadas em nossas intervenções. Conforme descrito no Capítulo 6, há a possibilidade de que outras ferramentas e atividades tais como jogos de lógica na plataforma *RachaCuca* e o *AppInventor* sejam utilizados no curso como substitutos ao Estacionamento Algorítmico e ao Scratch. Portanto, outro trabalho a ser feito é realizar novos cursos utilizando ferramentas diferentes daquelas recomendadas na sequência didática para avaliá-las o quão interessantes e difíceis elas seriam para os

alunos.

Novos estudos similares, envolvendo a pesquisa-ação, devem ser conduzidos a fim de abordarmos a metáfora Estrangeiro sem Tradutor e verificarmos o quão difícil será para os alunos utilizarem linguagens de programação convencionais após serem expostos a Computação através das quatro primeiras metáforas. Os resultados desses novos cursos podem ser utilizados para complementar a sequência didática proposta por nós para que ela aborde todo o *continuum* das MPCs.

O uso das MPCs como forma de guiar o ensino de PC foi bem sucedido no nosso trabalho, logo, há também a possibilidade que elas possam ser utilizadas para introduzir os outros dois eixos da Computação definidos pela SBC [2018] (i.e. Cultura Digital e Mundo Digital) caso eles sejam abordados como uma cultura estrangeira sendo apresentada.

Embora tenhamos disponibilizado o curso e os materiais gerado para as aulas online, seria interessante desenvolver um ambiente (i.e.: site) que organizasse a disponibilização da sequência didática e permitisse que professores interessados não apenas tivessem um acesso fácil ao material, mas pudessem também contribuir com comentários sobre sua experiência, uso de outros sistemas e mesmo novos materiais.

Finalmente, talvez seja interessante conduzir estudos que permitam relacionar a nossa sequência didática às diretrizes da SBC [SBC, 2018], currículo do [CIEB, 2017], habilidades da nova BNCC [MEC, 2020] e os livros didáticos da série *Computação no Ensino Fundamental* [Santana et al., 2019; Araujo et al., 2019; Santana et al., 2020; Araujo et al., 2020].

Referências Bibliográficas

- Adelman, C. (1993). Kurt lewin and the origins of action research. *Educational Action Research*, 1(1):7–24.
- Almeida, A.; Raabe, A. & Voigt, N. (2019a). Robótica na educação não é um bicho-papão: relato de experiência na rede pública municipal. *Anais do Workshop de Informática na Escola*, 25(1):266. ISSN 2316-6541.
- Almeida, T.; Castro, T. & Gadelha, B. (2019b). Um relato de experiência sobre o uso do pensamento computacional para potencializar o ensino de ciências na rede básica de ensino. *Anais do Workshop de Informática na Escola*, 25(1):657. ISSN 2316-6541.
- Almeida, T.; Netto, J. & Custódio, T. (2017). Desenvolvimento e configuração de cenários de robótica para fomentar a aprendizagem de programação aos alunos do ensino fundamental. *Anais do Workshop de Informática na Escola*, 23(1):205. ISSN 2316-6541.
- Almeida, W. D. & Junior, A. C. (2020). A aplicação de uma sequência didática no processo de desenvolvimento do pensamento computacional com alunos do 4º ano do ensino fundamental i. In *Anais do XXVI Workshop de Informática na Escola*, pp. 11–20, Porto Alegre, RS, Brasil. SBC. ISSN 0000-0000.
- Aono, A. H.; Rody, H. V. S.; Musa, D. L.; Pereira, V. A. & Almeida, J. (2017). A Utilização do Scratch como Ferramenta no Ensino de Pensamento Computacional para Crianças. *25º Workshop sobre Educação em Computação (WEI)*, pp. 2169--2178.
- Araujo, L. G. J.; Santana, B. L. & Bittencourt, R. A. (2019). *Computação e Comunidade : Livro do Professor*. Edição do Autor, Feira de Santana, 1 edição. ISBN 9786590132147. <https://sites.google.com/view/computacaofundamental/setimoano>.

- Araujo, L. G. J.; Santana, B. L. & Bittencourt, R. A. (2020). *Computação e o Mundo : Livro do Professor*. Edição do Autor, Feira de Santana, 1 edição. ISBN 9786500014518. <https://sites.google.com/view/computacaofundamental/setimoano>.
- Araújo, L.; Santana, B. & Bittencourt, R. (2019). Computação e comunidade: Uma proposta de educação em computação para o sétimo ano do ensino fundamental ii. *Anais do Workshop de Informática na Escola*, 25(1):325. ISSN 2316-6541.
- Barata, P. E. C.; Corrêa, J. a. V. P. & Mota, M. P. (2017). A study on knowledge transfer between programming languages by programs meanings facets. In *Proceedings of the XVI Brazilian Symposium on Human Factors in Computing Systems*, IHC 2017, New York, NY, USA. Association for Computing Machinery.
- Barbosa, S. D. J. & da Silva, B. S. (2010). *Interação Humano-Computador*. Elsevier, 1 edição. ISBN 8535234187.
- Barr, V. & Stephenson, C. (2011). Bringing computational thinking to k-12: What is involved and what is the role of the computer science education community? *ACM Inroads*, 2:48--54. ISSN 2153-2184.
- Barroso, R.; Santos, A. & Machado, V. (2019). Loop - tabuleiro lógico: uma proposta de ensino com computação desplugada. *Anais do Workshop de Informática na Escola*, 25(1):628. ISSN 2316-6541.
- Bathke, J. & Raabe, A. (2016). Pensamento computacional na educação de jovens e adultos: Lições aprendidas. *Anais dos Workshops do Congresso Brasileiro de Informática na Educação*, 5(1):1087. ISSN 2316-8889.
- Batista, G. & Damasceno, A. (2019). Análise dos motivos para o desuso de recursos computacionais por professores de escolas públicas. *Anais do Workshop de Informática na Escola*, 25(1):859. ISSN 2316-6541.
- Bauer, R.; Flores, G.; Crestani, A. & Mombach, J. (2017). Projeto codific@r: Oficinas de programação em dispositivos móveis no ensino fundamental. *Anais dos Workshops do Congresso Brasileiro de Informática na Educação*, 6(1):1210. ISSN 2316-8889.
- Bell, T.; Witten, I. H.; Fellows, M.; Adams, R. & McKenzie, J. (2015). Computer science unplugged: An enrichment and extension programme for primary-aged children. <https://classic.csunplugged.org/books/>.
- Bezerra, F. (2014). Bem mais que os bits da computação desplugada. *Anais do Workshop de Informática na Educação*.

- Blatt, L.; Becker, V. & Ferreira, A. (2017). Mapeamento sistemático sobre metodologias e ferramentas de apoio para o ensino de programação. In *Anais do Workshop de Informática na Escola*, volume 23, p. 815.
- Boucinha, R. M. (2017). *Aprendizagem do pensamento computacional e desenvolvimento do raciocínio*. Tese de doutorado, Universidade Federal do Rio Grande do Sul.
- Boucinha, R. M.; Brackmann, C. P.; Barone, D. A. C. & Casali, A. (2017). Construção do pensamento computacional através do desenvolvimento de games. *RENOTE-Revista Novas Tecnologias na Educação*, 15(1).
- Braga, L.; Braga, D.; Gusmão, A.; Souza, C. & Gualberto-Leite, N. (2018). Programchildren: Levando tecnologia para crianças de uma escola pública. *Anais do Workshop de Informática na Escola*, 24(1):295. ISSN 2316-6541.
- Braga, N. C. (2019). Análise do ensino de pensamento computacional no ensino fundamental ii. Universidade Federal de Minas Gerais. Projeto de Conclusão de Curso.
- Carlos, L.; Godinho, J. & Gomide, J. (2018). Um relato de experiência da escola de verão de programação para crianças. *Anais do Workshop de Informática na Escola*, 24(1):41--50. ISSN 2316-6541.
- Casarotto, R. I.; Bernardi, G.; Cordenonsi, A. Z. & Medina, R. D. (2018). Logirunner: um jogo de tabuleiro como ferramenta para o auxílio do ensino e aprendizagem de algoritmos e lógica de programação. *RENOTE-Revista Novas Tecnologias na Educação*, 16(1).
- Castro, M.; Werneck, V. & Gouvea, N. (2016). Ensino de matemática através de algoritmos utilizando jogos para alunos do ensino fundamental ii. *Anais dos Workshops do Congresso Brasileiro de Informática na Educação*, 5(1):1039. ISSN 2316-8889.
- CIEB (2017). Centro de Inovação para a Educação Brasileira currículo de referência em tecnologia e computação. http://curriculo.cieb.net.br/assets/docs/Curriculo_de_Referencia_em_Tecnologia_e_Computacao.pdf. Acessado em: 03-02-2020.
- Conforto, D.; Cavedini, P.; Miranda, R. & Caetano, S. (2018). Pensamento computacional na educação básica: interface tecnológica na construção de competências do século xxi. *Revista Brasileira de Ensino de Ciências e Matemática*, 1(1).

- Costa, T.; Cristiano, F.; Rocha, P. & Danielle, D. (2017). O ensino de linguagem de programação na educação básica através da robótica educacional: Práticas e a interdisciplinaridade. *Anais do Workshop de Informática na Escola*, 23(1):687. ISSN 2316-6541.
- Creswell, J. W. (2014). *Research Design: Qualitative, Quantitative, and Mixed Methods Approaches*. SAGE Publications. ISBN 9781452226101.
- Cândido, D.; Pessoa, G.; Vasconcelos, B.; Silva, K.; Oliveira, R.; Taumaturgo, M. & Falcão, T. P. (2017). Estudo comparativo de abordagens referentes ao desenvolvimento do pensamento computacional. *Anais do Workshop de Informática na Escola*, 23(1):382--391. ISSN 2316-6541.
- Daniel, G. T.; von Wangenheim, C. G.; Araújo, G.; de Medeiros, S. & da Cruz Alves, N. (2017). Ensinando a computação por meio de programação com app inventor. *Anais do Computer on the Beach*, pp. 357--365.
- de Oliveira, E. R.; Santos, N. V. & de Almeida, D. A. G. (2018). Computação para todos na escola: Relato de experiência com alunas. In *X Congreso de la Mujer Latinoamericana en Computación (LAWCC)*, Sao Paulo, Brazil.
- de Oliveira, T.; Monteiro, W.; Oliveira, F. C.; Martins, D. J. & da Silva, A. (2017). Ensino de raciocínio lógico e computação para crianças: Experiências, desafios e possibilidades. In *Anais do XXV Workshop sobre Educação em Computação*, Porto Alegre, RS, Brasil. SBC. ISSN 2595-6175.
- de Souza, C. S. (2005). *The semiotic engineering of human-computer interaction*. MIT press.
- Duarte, K.; Silveira, T. & Borges, M. (2017). Abordagem para o ensino da lógica de programação em escolas do ensino fundamental ii através da ferramenta scratch 2.0. *Anais do Workshop de Informática na Escola*, 23(1):175--184. ISSN 2316-6541.
- Falcão, T. P.; Gomes, T. C. S. & Albuquerque, I. R. (2015). Computational thinking through children's games: an analysis of interaction elements. In *Proceedings of the 14th Brazilian Symposium on Human Factors in Computing Systems, IHC 2015, Salvador, Brazil, November 03-06, 2015*, pp. 29:1--29:10. ACM.
- Friedrich, R.; dos Santos, D.; Keller, R.; Puntel, M. & Biasoli, D. (2012). Proposta metodológica para a inserção ao ensino de lógica de programação com logo e lego mindstorms. *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação - SBIE)*, 23(1). ISSN 2316-6533.

- Garcia, L.; Tenório, T.; Soares, R.; Costa, T.; Alves, C.; Albuquerque, N. & Assis, T. (2019). Uso de scratch com alunos de escola pública de penedo-al para ensino da matemática. *Anais do Workshop de Informática na Escola*, 25(1):375. ISSN 2316-6541.
- Geraldes, W.; Martins, E. & Afonseca, U. (2019). Uma experiência em ensino de programação de computadores com alunos da rede pública usando a linguagem scratch. *Anais do Workshop de Informática na Escola*, 25(1):129. ISSN 2316-6541.
- Godinho, J.; Torres, K.; Batista, G.; Andrade, E. & Gomide, J. (2017). Projeto aprenda a programar jogando: Divulgando a programação de computadores para crianças e jovens. In *Anais do XXV Workshop sobre Educação em Computação*, Porto Alegre, RS, Brasil. SBC. ISSN 2595-6175.
- Gomes, T. & Alencar, A. (2015). Análise empírica de jogos educativos para dispositivos móveis voltados a disseminação do pensamento computacional na educação básica. *Anais dos Workshops do Congresso Brasileiro de Informática na Educação*, 4(1):731. ISSN 2316-8889.
- Guarda, G.; Gonçalves, C. & Cunha, L. (2019). Jogo corrida das frações ? ludicidade e pensamento computacional. *Anais do Workshop de Informática na Escola*, 25(1):19. ISSN 2316-6541.
- Guarda, G.; Gonçalves, C.; Cunha, L. & Goulart, I. (2018). O circuito quatro desafios: Atividade lúdica apoiada pelo pensamento computacional. *Anais dos Workshops do Congresso Brasileiro de Informática na Educação*, 7(1):490. ISSN 2316-8889.
- Heintz, F.; Mannila, L. & Farnqvist, T. (2016). A review of models for introducing computational thinking, computer science and computing in k-12 education. In *2016 IEEE Frontiers in Education Conference (FIE)*. IEEE.
- INEP (2017). Instituto nacional de estudos e pesquisas educacionais anísio teixeira. censo da educação básica. <http://portal.inep.gov.br/documents/186968/0/Resumo+T%C3%A9cnico+%28vers%C3%A3o+preliminar%29+-+Censo+da+Educa%C3%A7%C3%A3o+B%C3%A9sica+2019/73e6de67-2be3-413f-9e4c-90c424d27d96?version=1.0>. Acessado em: 10-03-2020.
- Kalelioğlu, F. (2015). A new way of teaching programming skills to k-12 students: Code. org. *Computers in Human Behavior*, 52:200--210.

- Koshy, V. (2005). *Action Research for Improving Practice: A Practical Guide*. SAGE, 1 edição. ISBN 1412907551.
- Leitão, C. F.; Pereira, V. C. & Maciel, C. (2017). Exploring the communication of cultural perspectives in death-related interactive systems. In *Proceedings of the XVI Brazilian Symposium on Human Factors in Computing Systems, IHC 2017*, pp. 26:1-26:10, New York, NY, USA. ACM.
- Leitão, C. F. & Prates, R. O. (2017). A aplicação de métodos qualitativos em computação. *Jornadas de Atualização em Informática*, 2017:43--90.
- Lopes, A. & Ohashi, A. (2019). Estimular o pensamento computacional através da computação desplugada aos alunos do ensino fundamental. *Anais do Workshop de Informática na Escola*, 25(1):424. ISSN 2316-6541.
- Lye, S. Y. & Koh, J. H. L. (2014). Review on teaching and learning of computational thinking through programming: What is next for k-12? *Computers in Human Behavior*, 41:51--61.
- Maffi, C. (2018). *Inserção da robótica educacional nas aulas de matemática : desafios e possibilidades*. Tese de doutorado, Pontifícia Universidade Católica do Rio Grande do Sul.
- Marques, W.; de Souza, P. S. & Mombach, J. (2017). Pensar para programar: Projeto de ensino no curso técnico em informática. In *Anais do XXV Workshop sobre Educação em Computação*, Porto Alegre, RS, Brasil. SBC. ISSN 2595-6175.
- Martin, S.; Cavalheiro, S.; Reiser, R.; Foss, L.; Mazzini, A. R.; Bois, A. D. & Piana, C. (2018). Soccercraft: Relato de atividade para ensino aprendizagem de habilidades do pensamento computacional aplicada no sexto ano do ensino fundamental. In *Anais do XXVI Workshop sobre Educação em Computação*, Porto Alegre, RS, Brasil. SBC. ISSN 2595-6175.
- Martins, L. (2016). Ensinando lógica de programação aplicada a robótica para alunos do ensino fundamental. *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação - SBIE)*, 27(1):31. ISSN 2316-6533.
- Martins, R.; Reis, R. & Marques, A. (2016). Inserção da programação no ensino fundamental uma análise do jogo labirinto clássico da code.org através de um modelo de avaliação de jogos educacionais. *Anais do Workshop de Informática na Escola*, 22(1):121. ISSN 2316-6541.

- MEC (2020). Ministério da Educação. Base Nacional Comum Curricular. http://basenacionalcomum.mec.gov.br/images/BNCC_EI_EF_110518-versaofinal_site.pdf. Acessado em: 18-01-2020.
- Neves, V. F. A. (2013). História coletiva e construções subjetivas: uma trama de narrativas em uma creche comunitária. *Educação em Revista*, 29(1):225--246.
- Nicolaci-da Costa, A. M.; Leitão, C. F. & Romão-Dias, D. (2004). Como conhecer usuários através do método de explicitação do discurso subjacente (meds). *VI Simpósio Brasileiro sobre Fatores Humanos em Sistemas Computacionais, IHC*, pp. 47--56.
- Oliveira, G.; Assunção, O. & Prates, R. (2019). Strategies to introduce computational thinking to children: An analysis based on cultural viewpoint metaphors. *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação - SBIE)*, 30(1):547. ISSN 2316-6533.
- Oliveira, G. A. (2020). Strategies for teaching computational thinking in brazilian middle school: an analysis based on cultural viewpoint metaphors. Dissertação de mestrado, Universidade Federal de Minas Gerais.
- Oliveira, G. A. A. d.; Ferreira, R. S. M. & Prates, R. O. (2018). An exploratory analysis of interactive systems for introducing programming based on cultural viewpoint metaphors. In *Proceedings of the 17th Brazilian Symposium on Human Factors in Computing Systems, IHC 2018*, pp. 7:1--7:11, New York, NY, USA. ACM.
- Pereira, F.; Araújo, L. & Bittencourt, R. (2019). Intervenções de pensamento computacional na educação básica através de computação desplugada. *Anais do Workshop de Informática na Escola*, 25(1):315. ISSN 2316-6541.
- Raabe, A.; Ellery, N.; Jesus, E.; Silva, E.; Bombasar, J. & Santana, A. (2017). A experiência de implantação de uma disciplina obrigatória de pensamento computacional em um colégio de educação básica. *Anais dos Workshops do Congresso Brasileiro de Informática na Educação*, 6(1):1182. ISSN 2316-8889.
- Raabe, A. & Gomes, E. B. (2018). Maker: uma nova abordagem para tecnologia na educação. <https://tecedu.pro.br/wp-content/uploads/2018/09/Art1-vol.26-EdicaoTematicaVIII-Setembro2018.pdf>. Acessado em: 18-03-2020.
- Reis, E.; Santos, P.; Dutra, B.; Couto, L.; Oliveira, D. & Carneiro, M. (2018). Jovens programadores: ensino de programação e robótica para alunos do ensino básico de monte carmel-mg. *Anais dos Workshops do Congresso Brasileiro de Informática na Educação*, 7(1):649. ISSN 2316-8889.

- Resnick, M.; Maloney, J.; Monroy-Hernández, A.; Rusk, N.; Eastmond, E.; Brennan, K.; Millner, A.; Rosenbaum, E.; Silver, J.; Silverman, B. & Kafai, Y. (2009). Scratch: Programming for all. *Commun. ACM*, 52(11):60–67. ISSN 0001-0782.
- Riboldi, S. & Reichert, J. (2019). A linguagem de programação scratch e o ensino de funções: uma possibilidade. *Anais do Workshop de Informática na Escola*, 25(1):109. ISSN 2316-6541.
- Rino, M. V. (2018). *Estratégias Para a Utilização De Jogos Digitais E Tecnologias Emergentes No Ensino De Linguagem De Programação*. Tese de doutorado, Universidade Estadual Paulista.
- Rocha, K. & Basso, M. (2019). Programação em scratch na sala de aula: investigações sobre a construção do conceito de ângulo. *Anais do Workshop de Informática na Escola*, 25(1):725. ISSN 2316-6541.
- Rodrigues, S.; Aranha, E. & Silva, T. R. (2018). Computação desplugada no ensino de programação: Uma revisão sistemática da literatura. In *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*, volume 29, p. 417.
- Salgado, L. C. d. C. (2011a). *Cultural viewpoint metaphors to explore and communicate cultural perspectives in cross-cultural HCI design*. Tese de doutorado, Pontífica Universidade Católica do Rio de Janeiro.
- Salgado, L. C. d. C. (2011b). *Cultural Viewpoint Metaphors to Explore and Communicate Cultural Perspectives in Cross-Cultural HCI Design*. Tese de doutorado, Pontificia Universidade Católica Do Rio de Janeiro.
- Salgado, L. C. D. C.; Souza, C. S. D. & Leitão, C. F. (2011). On the epistemic nature of cultural viewpoint metaphors. In *Proceedings of the 10th Brazilian Symposium on Human Factors in Computing Systems and the 5th Latin American Conference on Human-Computer Interaction*, pp. 23--32. Brazilian Computer Society.
- Santana, B. L.; Araujo, L. G. J. & Bittencourt, R. A. (2019). *Computação e Eu : Livro do Professor*. Edição do Autor, Feira de Santana, 1 edição. ISBN 9786590132123. <https://sites.google.com/view/computacaofundamental/sextano>.
- Santana, B. L.; Araujo, L. G. J. & Bittencourt, R. A. (2020). *Computação e Sociedade : Livro do Professor*. Edição do Autor, Feira de Santana, 1 edição. ISBN 9786590132161. <https://sites.google.com/view/computacaofundamental/oitavoano>.

- Santana, S. & Santos, W. (2018). Jogos educativos no ensino de matemática: Qual a melhor abordagem? *Anais do Workshop de Informática na Escola*, 24(1):80–89. ISSN 2316-6541.
- Santos, C.; Silva, D.; Roque, A.; Lima, J. & Ben, M. (2019). Tecendo espaços e experiências no campo da robótica educacional para fomentar o interesse de meninas pela área de computação. *Anais do Workshop de Informática na Escola*, 25(1):9. ISSN 2316-6541.
- Santos, P. S. C.; Araujo, L. G. J. & Bittencourt, R. A. (2018). A mapping study of computational thinking and programming in brazilian k-12 education. In *2018 IEEE Frontiers in Education Conference (FIE)*, pp. 1–8. ISSN 1539-4565.
- SBC (2017). Sociedade brasileira de computação. referenciais de formação em computação: Educação básica. <http://www.sbc.org.br/noticias/10-slideshow-noticias/1996-referenciais-de-formacao-em-computacao-educacao-basica>. Acessado em: 15-07-2019.
- SBC (2018). Sociedade Brasileira de Computação. Diretrizes para ensino de Computação na Educação Básica. <https://www.sbc.org.br/documentos-da-sbc/send/131-curriculos-de-referencia/1177-diretrizes-para-ensino-de-computacao-na-educacao-basica>. Acessado em: 11-01-2020.
- Seehorn, D.; Carey, S.; Fuschetto, B.; Lee, I.; Moix, D.; O’Grady-Cunniff, D.; Owens, B. B.; Stephenson, C. & Verno, A. (2011). Csta k–12 computer science standards: Revised 2011. Relatório técnico, New York, NY, USA.
- Silva, D.; Sidnei, S.; Ângelo Jesus & Silva, C. (2016). Aplicação de robótica na educação de forma gradual para o estímulo do pensamento computacional. *Anais dos Workshops do Congresso Brasileiro de Informática na Educação*, 5(1):1188. ISSN 2316-8889.
- Silva, E. & Javaroni, S. (2018). Pensamento computacional e atividades com robótica para a promoção da aprendizagem sobre o significado do resto da divisão euclidiana. *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação - SBIE)*, 29(1):815. ISSN 2316-6533.
- Silva, G. F. M.; Salgado, L. C. d. C. & Raposo, A. B. (2013). Metáforas de perspectivas culturais na (re) definição de padrões de colaboração de um jogo de multi-toque para

- usuários com autismo. In *Proceedings of the 12th Brazilian Symposium on Human Factors in Computing Systems*, IHC '13, pp. 112--121, Porto Alegre, Brazil, Brazil. Brazilian Computer Society.
- Silva, J.; Cristiano, F.; Martins, D. & Silva, W. (2018a). Storytelling e robótica educacional: a construção de carros robôs com arduino e materiais recicláveis. *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação - SBIE)*, 29(1):1806. ISSN 2316-6533.
- Silva, K. d. S.; Pereira, N. P. & Odakura, V. (2018b). Mapeamento sistemático: estratégias para o ensino-aprendizagem do pensamento computacional no brasil. In *XXIII Congresso Internacional de Informática Educativa*, pp. 319--329.
- Silva, M. C. d. (2018). *Robótica Educacional Livre: um relato de prática no Ensino Fundamental*. Tese de doutorado, Pontifícia Universidade Católica de São Paulo.
- Silva, N.; Lima, A.; Souza, N. & Sousa, D. (2017). Raciocínio lógico nas escolas: Uma introdução ao ensino de algoritmos de programação. *Anais dos Workshops do Congresso Brasileiro de Informática na Educação*, 6(1):1011. ISSN 2316-8889.
- Silva, V.; Moura, H.; Paula, S. & Ângelo Jesus (2019). Algo+ritmo: Uma proposta desplugada com a música para auxiliar no desenvolvimento do pensamento computacional. *Anais do Workshop de Informática na Escola*, 25(1).
- Solomon, C. J. & Papert, S. (1976). A case study of a young child doing turtle graphics in logo. In *Proceedings of the June 7-10, 1976, National Computer Conference and Exposition*, AFIPS '76, pp. 1049--1056, New York, NY, USA. ACM.
- Souza, F. & Nunes, M. (2019). Práticas e resultados obtidos na aplicação do pensamento computacional desplugado no ensino básico: Um mapeamento sistemático. In *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação - SBIE)*, volume 30, p. 289. ISSN 2316-6533.
- Souza, O. (2019). Joglog - jogos de raciocínio lógico para alunos do ensino fundamental: Um estudo de caso utilizando gamificação e pensamento computacional. *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação - SBIE)*, 30(1):1022. ISSN 2316-6533.
- Spradley, J. P. (1980). *Participant Observation*. Holt, Rinehart and Winston, 1 edição. ISBN 0030445019.

- Todos pela Educação (2017). O que pensam os professores brasileiros sobre a tecnologia digital em sala de aula? <https://www.todospelaeducacao.org.br/tecnologia/>. Acessado em: 10-03-2020.
- Tripp, D. (2005). Pesquisa-ação: uma introdução metodológica. *Educação e pesquisa*, 31(3):443--466.
- Valente, J. (2016). Integração do pensamento computacional no currículo da educação básica: Diferentes estratégias usadas e questões de formação de professores e avaliação do aluno. *Revista e-Curriculum*, 14(3):864--897. ISSN 1809-3876.
- Victal, E. & Cândido, A. (2019). Aprendendo sobre o uso da robótica para introdução à programação: um relato de experiência. *Anais do Workshop de Informática na Escola*, 25(1):491. ISSN 2316-6541.
- Wing, J. M. (2006). Computational thinking. *Commun. ACM*, 49:33--35. ISSN 0001-0782.
- Wing, J. M. (2011). Computational thinking—what and why? <https://www.cs.cmu.edu/link/research-notebook-computational-thinking-what-and-why>. Accessed: 2018-11-03.

Apêndice A

Roteiro das Aulas

Este apêndice contém a descrição das aulas, atividades e recomendações para guiar professores interessados em introduzir a Computação para alunos do Ensino Fundamental II e fomentar o desenvolvimento de suas habilidades do Pensamento Computacional. Os materiais listados nas tabelas podem ser acessados através do seguinte link: [Materiais e Recursos](#).

Aula 1 - Exercitando o Raciocínio Lógico

Na primeira aula do GTD os alunos praticarão o raciocínio lógico através de um jogo de lógica desplugado chamado *Estacionamento Algorítmico*, que é um jogo simples de ser explicado e entendido. No jogo, os alunos devem movimentar os veículos para permitir que o carro vermelho consiga sair do tabuleiro.

Objetivos	Introduzir indiretamente o sequenciamento de ações.
	Introduzir indiretamente o registro de soluções.
	Introduzir indiretamente o teste de soluções.
Atividade	Estacionamento Algorítmico
Recursos e Materiais	Um conjunto do Estacionamento Algorítmico (tabuleiro e veículos) para cada aluno
	Uma Folha de Desafios para cada aluno
	Uma Folha de Registro de Soluções para cada aluno
	Folhas de rascunho para cada aluno
	Materiais de escrever básicos (lápis e borracha)
	Computador para o professor
	Projector de vídeo
	Aula 1 - Apresentação

Descrição da Aula

Esta aula é estruturada em quatro momentos principais: apresentação do curso, apresentação da atividade, resolução dos desafios e registro das soluções e teste de soluções.

Apresentação do Curso

No momento inicial da aula o(s) professor(es) e alunos devem se apresentar e conversar brevemente sobre as motivações dos alunos, o que eles esperam aprender no curso, a área da Computação, o Pensamento Computacional e a importância que a Computação e as habilidades do Pensamento Computacional têm no mundo de hoje. Em seguida, o professor deve apresentar uma visão geral do curso e das atividades que os alunos realizarão.

Os alunos podem ter expectativas que utilizarão os computadores em todas as aulas do GTD. Assim, é recomendado lembrá-los que embora o curso esteja focado na Computação e no Pensamento Computacional, atividades desplugadas serão conduzidas nas aulas iniciais e que eles também podem aprender sobre a Computação através delas.

Apresentação do Jogo

Após a apresentação inicial, o professor deve introduzir a atividade que os alunos estarão fazendo na primeira aula: o *Estacionamento Algorítmico*. A utilização de um projetor de vídeo e uma apresentação de slides com animações é recomendável para apresentar o jogo aos alunos. A *Aula 1 - Apresentação* pode ser utilizada e, caso necessário, adaptada. Ela contém slides apresentando o jogo, suas regras e um exemplo interativo de resolução e registro da solução.

Resolução e Registro de Soluções

No terceiro momento da aula os alunos devem resolver pelo menos dois dos cinco desafios contidos na Folha de Desafios. Os desafios podem ser conduzidos tanto individualmente quanto em dupla. A resolução individual é recomendada para que todos os alunos pratiquem igualmente a resolução dos desafios. É importante frisar que os alunos anotem os passos na folha de rascunho a cada movimento ou conjunto pequeno de movimentos testados para que eles não se esqueçam no final os passos que eles seguiram para resolver o desafio. Além disso, o professor deve pedi-los que os a resolução seja registrada assim que eles finalizarem o desafio antes de passarem para o próximo. Talvez não seja possível registrar a solução de todos os cinco desafios devido ao tempo

disponível, porém é importante que pelo menos uma solução seja registrada por cada aluno. O registro deve ser feito conforme indicado na apresentação do jogo, informando os veículos na ordem em que eles devem ser movimentados para que o desafio seja resolvido.

Teste de Soluções

Nos 20 minutos finais da aula, o professor deve pedir para que os alunos troquem suas soluções com um colega de turma e que eles verifiquem se anotaram no topo das folhas de resolução o número do desafio. Em seguida, o professor deve dizer aos alunos que eles devem testar as soluções que eles receberam seguindo o passo a passo registrado pelo seu colega de classe. Caso haja erros, peça que os alunos indiquem na folha de resolução qual dos passos está errado.

Aula 2 - Introdução ao Sequenciamento de Ações

A segunda aula do curso os alunos terão contato direto com a Computação e conduzirão atividades práticas que estão explicitamente relacionadas a área, permitindo que as habilidades do Pensamento Computacional sejam praticadas e desenvolvidas. Nesta aula o conceito de algoritmos é apresentado aos alunos e praticado através da atividade desplugada *Programando em Papel*. A atividade consiste em utilizar comandos de movimentar e colorir para criar algoritmos para colorir figuras.

Objetivos	Compreender o conceito de algoritmos.
	Entender a importância da definição precisa de instruções.
	Praticar a criação de algoritmos simples.
Atividades	Programação em Papel, LightBot*, KidloCoding
Recursos e Materiais	Uma folha do <i>Criando Algoritmos</i> para cada aluno
	Uma folha do <i>Criando Figuras</i> para cada aluno
	Materiais de escrever básicos (lápiz e borracha)
	Computador para o professor
	Projeto de vídeo
	Aula 2 - Apresentação

*Verificar disponibilidade

Descrição da Aula

A Aula 2 está estruturada em três partes: algoritmos e sequenciamento de ações, decomposição de tarefas, criando algoritmos e LightBot.

Algoritmos e Sequenciamento de Ações

Na primeira parte da aula o professor deve apresentar formalmente o conceito de algoritmos aos alunos. Além da apresentação do conceito, é de extrema importância que os algoritmos sejam associados às soluções do Estacionamento Algorítmico registradas por eles na aula anterior. Situações e tarefas da vida cotidiana dos alunos também podem ser dadas como exemplos e decompostas em ações menores e específicas para reforçar que algoritmos estão presentes na vida dos alunos. A *Aula 2 - Apresentação* contém slides apresentando o conceito de algoritmo, um exemplo de solução do *Estacionamento Algorítmico*, um slide para instigar a discussão sobre os algoritmos na vida real e uma receita de doce para ser dada como exemplos.

Decomposição de Tarefas

Finalizada a apresentação do tópico da aula, o professor deve introduzir a decomposição de problemas aos alunos, indicando a importância de transformar problemas maiores em múltiplos problemas menores que são mais fáceis de serem resolvidos. Para prática e fixação da decomposição de problemas, conduza uma atividade em conjunto com a turma. O slide 6 contido na *Aula 2 - Apresentação* contém a tarefa que deverá ser decomposta com os alunos: chamar e andar de elevador. Utilize a lousa ou um editor de texto do computador conectado ao projetor para anotar os passos ditos pelos alunos. É importante que este processo não seja interferido ou corrigido pelo professor. Deixe que eles discutam entre si e construam o algoritmo.

Depois que a atividade de decomposição for finalizada, avance para o próximo slide e exiba o vídeo contido nele: *Como ensinar linguagem de programação para uma criança*. O vídeo exemplifica de uma maneira divertida e engraçada o quão importante é fazer uma definição precisa de instruções para que um problema seja solucionado. Após a exibição do vídeo, volte à lousa ou para o editor de texto que contém o algoritmo criado pela turma e pergunte à turma se o algoritmo criado por eles está realmente correto ou não.

Programação em Papel

Em seguida, a atividade *Programação em Papel* é conduzida com os alunos. A atividade consiste na utilização de cinco comandos para construir algoritmos que criarão determinadas figuras. Quatro comandos são utilizados para movimentação (i.e.: ir para a direita, ir para a esquerda, ir para cima, ir para baixo) e o quinto é utilizado para colorir a posição atual da estrela, que é o objeto de referência. As explicações sobre a atividade e um exemplo interativo ilustrando o funcionamento dos comandos e o processo de resolução dos desafios também foram incluídas na *Aula 2 - Apresentação* para o professor utilizar. A atividade foi dividida em duas partes: *Criado Algoritmo* e *Criando Figuras*.

Criando Algoritmos:: os alunos devem criar algoritmos para desenhar as figuras indicadas. Distribua as folhas do criando algoritmos para os alunos e peça que eles construam os algoritmos utilizando os cinco comandos que criarão as figuras dadas. A folha de atividades contém três exercícios para que os alunos criem os algoritmos.

LightBot*

Por fim, a versão do LightBot disponibilizada como parte da iniciativa Hora do Código é apresentada aos alunos para que eles resolvam os desafios do nível básico do jogo. Esse nível possui oito desafios a serem solucionados. Nesses desafios os alunos devem utilizar os comandos disponíveis para fazer com que o robô acenda todos os ladrilhos azuis no cenário. Ele pode ser acessado através do seguinte link: [LightBot - Hora do Código](#).

Obs.: *é necessário que o professor verifique previamente se a versão Hora do Código está disponível e se ela pode ser aberta nos computadores. Caso ela não esteja, recomendamos que as atividades apresentadas a seguir sejam realizadas na aula.*

Atividades para caso o LightBot não esteja disponível

Criando Figuras: os alunos devem criar suas próprias figuras e algoritmos que as criam para que colegas de turma os executem. O professor deve distribuir a folha de atividade do *Criando Figuras* a eles. O professor deve pedir que os alunos coloram pelo menos uma figura utilizando o grid em branco e que construam um algoritmo que criará esta figura que eles criaram no espaço ao lado. Conforme os alunos forem terminando a criação das figuras e algoritmos, o professor deve distribuir a folha *Grids em Branco* para os alunos e pedir que eles troquem suas figuras e algoritmos com um colega de turma para que eles sigam os algoritmos do colega na folha *Grids em Branco*.

Por fim, peça que eles verifiquem se a figura colorida por eles é a mesma que a figura original. Caso diferenças existam, a dupla deve trabalhar junto para descobrir onde está o problema. O problema pode estar na execução incorreta do algoritmo ou nos próprios algoritmos e figuras disponibilizadas.

KidloCoding: recomendamos que o professor apresente aos alunos o KidloCoding caso as atividades do *Progamando em Papel* sejam finalizadas muito rapidamente pelos alunos. A plataforma contém diferentes atividades envolvendo o sequenciamento de ações, loops, procedimentos, entre outras. O link a seguir leva diretamente aos desafios do sequenciamento de ações: [KidloCoding](#).

Deixe que os alunos acessem a plataforma e explorem as diferentes atividades e tentem resolver os desafios até o fim da aula. A plataforma está disponível apenas em inglês, porém o funcionamento das atividades é intuitivo o bastante para que os alunos as entendam após uma curta explicação.

Aula 3 - Introdução aos Laços de Repetição

Nesta aula os alunos aprendem sobre os laços de repetição e a utilizá-los para simplificar algoritmos escritos em uma linguagem de programação em bloco.

Objetivos	Compreender o conceito de laços de repetição
	Entender como os laços de repetição podem ser utilizados para simplificar um algoritmo
	Identificar ações ou conjunto de ações que se repetem
	Utilizar os laços de repetições para agrupar ações e repeti-las
	Aprender a utilizar uma linguagem de programação em blocos
Atividades	Kahoot 1 - Algoritmos e Programando em Papel
	Code.org - Curso 2 - Labirinto: Ciclos
Recursos e Materiais	Computador para o professor
	Projektor de vídeo
	Computadores para os alunos utilizarem
	Conexão à Internet
	Seção da turma criada no Code.org
	Conta dos alunos adicionadas à seção
	Aula 3 - Apresentação

Descrição da Aula

Esta é a primeira aula do curso com atividades conduzidas exclusivamente de forma plugada. Após a condução de uma pequena revisão de conteúdo, os laços de repetição e suas utilidades serão apresentados aos alunos e eles praticarão o seu uso através da lição *Labirinto: Ciclos* do Curso 2 do Code.org.

Obs.: o professor deve criar uma conta na plataforma Code.org e criar uma seção de aula para turma. Ao criar a seção, o professor deve atribuir o Curso 2 e adicionar os alunos à seção. Os alunos não precisam ter contas pessoais para serem adicionados. A plataforma atribuirá uma senha individual, que o professor terá acesso, para cada aluno adicionado.

Revisão

Antes do professor apresentar o tópico principal da terceira aula, é recomendado uma revisão do conteúdo visto e a atividade Programando em Papel conduzida na aula anterior. A fim de conduzir a revisão, sugerimos a ferramenta Kahoot! que permite que quizzes online e gamificados sejam conduzidos com a turma de maneira divertida, instigando uma competição saudável entre os alunos. O kahoot criado por nós¹ pode ser acessado através do link no pé da página e ele pode ser copiado, adaptado e modificado pelo professor caso necessário. Durante a condução do kahoot, recomendamos que explicações sejam dadas logo após os alunos responderem cada pergunta. A ferramenta exibe a quantidade de respostas que cada alternativa recebeu e a resposta correta. Portanto, é importante que o professor além de explicar o porquê da resposta correta estar certa, também explique, para as alternativas erradas que receberam respostas, o porquê delas estarem incorretas.

Obs.: caso o LightBot tenha sido utilizado na aula anterior, o professor também pode conduzir um kahoot para os alunos revisarem os desafios do LightBot. O kahoot² que criamos possui seis questões de revisão.

Obs.: os resultados dos kahoots também podem ser utilizados pelo professor para que ele avalie o aprendizado dos alunos e reorganize aulas seguintes caso seja preciso.

Linguagem de Programação em Blocos no Code.org e Laços de Repetição

Finalizada a revisão de conteúdo, a plataforma Code.org e os laços de repetição, tópico da aula, devem ser apresentados aos alunos. A *Aula 3 - Apresentação* contém slides

¹[Link para o Kahoot 1 - Algoritmos e Programando em Papel](#)

²[Link para o Kahoot - LightBot](#)

com a definição formal de laços de repetição, suas utilidades e exemplos de uso no contexto do Code.org.

Antes do professor apresentar os laços de repetição, acesse a plataforma Code.org utilizando o computador conectado ao projetor de vídeo e introduza a plataforma aos alunos. Para a apresentação da plataforma e da linguagem de programação em blocos que os alunos utilizarão na aula, acesse a lição *Labirinto: Sequência* do Curso 2. Apresente a interface da ferramenta, os desafios, a linguagem de programação em blocos e como a linguagem é utilizada para solucionar os desafios ao resolver os primeiros desafios da lição junto com a turma.

Em seguida, o professor deve voltar para a *Aula 3 - Apresentação* para introduzir os laços de repetição para a turma. Primeiro, o professor deve apresentar formalmente o conceito. Em seguida, mostrar aos alunos como os laços podem ser utilizados para simplificar algoritmos. Slides com a definição e exemplificando a importância dos laços no contexto do Code.org foram incluídos na *Aula 3 - Apresentação*.

Para fixação do conteúdo, criamos slides com exercícios para serem conduzidos com a turma, onde os alunos devem indicar a ordem em que os comandos serão executados dado um algoritmo do Code.org. Estes exercícios foram incluídos para que eles entendam o que acontece, no que diz respeito à execução dos comandos, quando os laços de repetição são utilizados.

Explicações sobre os slides foram colocadas como lembretes para auxiliar o professor caso necessário.

Lição Labirinto: Ciclos

Após a introdução do tópico da aula, o professor deve compartilhar com os alunos a URL da seção da turma e as credenciais para eles acessarem a seção. Na página inicial da seção, o professor pode encontrar os cartões com as credenciais dos alunos, que contém a URL da seção e as suas senhas. Estes cartões podem ser impressos e distribuídos para a turma antes do início da atividade. Outra opção é a inclusão da URL da seção e as credenciais dos alunos na apresentação de slides.

Por fim, durante o resto da aula os alunos devem tentar resolver os desafios da lição. É importante que o professor frise que os alunos devem tentar resolver os desafios da lição utilizando o menor número de blocos de comando possível, o que representa a utilização eficiente dos blocos. Explique aos alunos que após a resolução de um desafio, o Code.org irá informá-los se eles utilizaram o menor número de blocos ou não. Este feedback pode não ser percebido pelos alunos, logo é recomendável informá-los sobre quando e onde ele aparece na tela do computador durante a apresentação da ferramenta

no início da aula.

Caso os alunos consigam finalizar todos os desafios da lição utilizando o número mínimo de blocos, há outras lições no Curso 2 que podem ser disponibilizadas à turma para que estes alunos tentem resolvê-las: *Artista: Ciclos*, *Abelha: Ciclos*. Porém, devido à complexidade geral da lição, a grande maioria dos alunos pode não finalizar a lição em apenas uma aula, logo recomendados que o professor dê continuidade à atividade na aula seguinte caso necessário.

Aula 4 - Introdução aos Laços de Repetição (Continuação)

Nesta aula, é dada continuidade a lição *Labirinto: Ciclos* iniciada na aula anterior para que os alunos tenham mais tempo para resolver os desafios mais complexos da lição de forma eficiente e explorem alguns dos desafios das outras lições do curso envolvendo os laços de repetição.

Objetivos	Compreender o conceito de laços de repetição.
	Entender como os laços de repetição podem ser utilizados para simplificar um algoritmo.
	Identificar ações ou conjunto de ações que se repetem.
	Utilizar os laços de repetições para agrupar ações e repeti-las.
	Aprender a utilizar uma linguagem de programação em blocos.
Atividade	Code.org - Curso 2 - Labirinto: Ciclos
Recursos e Materiais	Computador para o professor
	Projector de vídeo
	Computadores para os alunos utilizarem
	Conexão à Internet
	Seção da turma criada no Code.org
	Conta dos alunos adicionadas à seção
	Aula 4 - Apresentação

Descrição da Aula

Nesta aula, além da continuação da lição *Labirinto: Ciclos*, é apresentado aos alunos uma forma de identificar ações ou conjunto de ações que se repetem para auxiliá-los na resolução dos desafios da lição.

Revisão e identificação de ações ou grupo de ações repetidas

No início da aula de continuação, o professor faz uma curta revisão dos laços de repetição e de sua utilização na lição do Code.org e em seguida apresenta a dica para identificar ações que se repetem e para evitar a utilização de mais blocos que o necessário.

A dica consiste em: (i) resolver o desafio sem a utilização dos blocos de repetição, (ii) analisar o algoritmo para identificar o conjunto de ações que se repetem, (iii) utilizar o bloco de repetição para repetir o conjunto de ações que se repete. Um exemplo foi incluído na *Aula 4 - Apresentação* para que os alunos visualizem a realização deste processo.

O professor também deve relembrar os alunos que os círculos preenchidos com verde mais escuro indicam que a lição foi completada com o menor número de blocos e que os círculos preenchidos com verde mais claro indicam que eles utilizaram mais blocos que o necessário no desafio (Figura A.1).



Figura A.1: Indicação visual se as lições foram completadas com o menor número de blocos ou não

Por fim, dois slides foram incluídos no fim da apresentação para ilustrar dois exemplos de maus usos do bloco de repetição. O primeiro diz respeito à utilização do bloco de repetição para repetir ações apenas uma única vez, levando a utilização de um bloco extra. E o segundo exemplo é a utilização do bloco de repetição para repetir uma ação duas vezes, que embora utilize o mesmo número de blocos, acaba resultando na execução de um bloco extra (i.e.: bloco de repetição).

Labirinto: Ciclos e outras lições envolvendo os laços

Durante o restante da aula, os alunos devem continuar resolvendo os desafios da lição *Labirinto: Ciclos* sempre buscando utilizar o número mínimo de blocos. Conforme indicado na aula anterior, o Curso 2 do Code.org possui outras lições que abordam a

utilização dos laços de repetição (i.e.: Artista: Ciclos e Abelha: Ciclos). Disponibilize estas lições para a turma para e deixe que os alunos que conseguiram finalizar todos os desafios da lição Labirinto: Ciclos as explorem e tentem resolver alguns dos desafios.

Aula 5 - Introdução ao Plano Cartesiano e o Scratch

Nesta aula são apresentados para os alunos o Plano Cartesiano de forma simplificada e a ferramenta Scratch através da condução de tarefas simples.

Objetivos	Entender o que é o Plano Cartesiano e os eixos x e y .
	Aprender a identificar pontos no Plano Cartesiano.
	Aprender sobre a plataforma Scratch, suas utilizações e comandos básicos de movimento, repetição e aparência.
	Aplicar o conhecimento sobre o Plano Cartesiano para movimentar personagens no Scratch através da alteração de suas coordenadas x e y .
Atividades	Kahoot 2 - Algoritmos e Loops
	Scratch
Recursos e Materiais	Computador para o professor
	Projektor de vídeo
	Computadores para os alunos utilizarem
	Conexão à Internet
	Turma criada no Scratch caso a versão web do Scratch seja utilizada
	Conta dos alunos criadas e adicionadas a turma do Scratch
	Aula 5 - Apresentação

Descrição da Aula

A quinta aula foi estruturada em três partes: *Revisão*, *Aprendendo sobre o Plano Cartesiano*, *Introdução ao Scratch*.

Revisão

No momento inicial da aula, o professor conduz um kahoot³ para revisar todo o conteúdo abordado no curso até o momento. Os conteúdos abordados no kahoot criado por nós são: algoritmos, laços de repetição, atividades do Programando em Papel e atividades da lição Labirinto: Laços.

Aprendendo sobre o Plano Cartesiano

A primeira parte da aula foi alocada para uma explicação simplificada do Plano Cartesiano, focando nos eixos x e y e identificação de pontos no plano dado os valores de cada eixo. A apresentação do Plano Cartesiano é recomendável, pois no Scratch os alunos terão que utilizar comandos que movimentam personagens e objetos através da modificação dos valores x e y . Na *Aula 5 - Apresentação*, foram incluídos imagens do Plano Cartesiano juntamente com alguns exercícios de identificação de pontos para serem conduzidos com a turma para fixação do que foi aprendido.

Primeiramente, o professor deve explicar o que é o Plano Cartesiano e os eixos x e y de forma simplificada. Indique através do exemplo na apresentação, que o eixo x é a reta horizontal e representa as direções *esquerda* e *direita*. Similarmente, fale que o eixo y é a reta vertical, representando as direções *cima* e *baixo*. Para os valores negativos do Plano Cartesiano, não é necessário que o conceito de números negativos seja explicado explicitamente aos alunos. O professor pode simplesmente, indicar que o sinal de menos (-) indica que o ponto abaixo do eixo x ou a esquerda do eixo y .

O slide inicial também contém alguns exemplos de pontos com os seus valores para que o professor demonstre o que são os pontos e como identificá-los no plano. Finalmente, pergunte à turma se eles entenderam o que foi explicado e, caso necessário, explique novamente o que os alunos não conseguiram entender.

Por fim, o professor inicia a atividade de fixação com a turma. A atividade consiste em pedir aos alunos para identificar pontos que aparecem no plano cartesiano na *Aula 5 - Apresentação*. Os pontos aparecem um de cada vez conforme o professor avança os slides. Primeiro apenas o ponto aparece e em seguida o par ordenado (x, y) do ponto. Para cada ponto, o professor pergunta a um aluno diferente quais são os

³[Link para o Kahoot 2 - Algoritmos e Loops](#)

valores x e y do ponto. E depois que o aluno responder, questione a turma se a resposta dada está correta ou não e, caso não esteja, pergunte à turma qual seria a resposta correta. Diversos pontos foram incluídos na apresentação para que todos os alunos tenham a oportunidade de praticar.

Introdução ao Scratch

O restante da sexta é alocado para a apresentação do Scratch aos alunos e o teste de blocos de comando responsáveis pela movimentação, mudança de aparência e repetição de ações.

Primeiramente, utilizando o computador conectado ao projetor de tela, o professor apresenta o Scratch como um todo, indicando suas principais finalidades, ou seja, criação de jogos, animações e histórias. Após esta apresentação inicial, compartilhe com os alunos as credenciais de suas contas. O compartilhamento pode ser feito através de um slide que contém o nome do usuário e a senha dos alunos. Em seguida, mostre aos alunos onde eles devem ir na plataforma para acessar suas contas e para sair delas no fim da aula. É importante chamar a atenção dos alunos para que eles não salvem suas senhas nos navegadores e sempre saiam de suas contas quando finalizarem as aulas.

Obs.: a versão desktop do Scratch também pode ser utilizada, não sendo necessário que os alunos acessem suas contas. Eles podem simplesmente abrir o Scratch no computador e começar a utilizá-lo. A criação de turmas e a utilização de contas no Scratch online facilita o compartilhamento dos projetos para que todos os alunos consigam visualizar e testar os projetos de seus colegas de turma.

Em seguida, o professor pede aos alunos para testarem alguns dos blocos no Scratch. Esta etapa deve ser feita junto com toda a turma a fim de garantir que todos os alunos testem e consigam entender o funcionamento dos blocos mais comumente utilizados em projetos. O professor deve fazer junto com os alunos o seguinte conjunto de tarefas:

1. Mostrar como adicionar novos personagens no projeto e escolher um personagem para ser adicionado. Deixe que os alunos adicionem o personagem que eles quiserem.
2. Após adicionar o novo personagem, mostre como os personagens podem ser excluídos ao excluir o gato do Scratch, que é o personagem que é adicionado sempre que um projeto é criado.

3. Similarmente, mostre aos alunos como eles podem adicionar um cenário de fundo em seus projetos, porém desta vez um cenário específico deverá ser adicionado: *xy-grid*. Mostre aos alunos como utilizar a barra de busca para que eles consigam encontrar o cenário *xy-grid*.
4. Explique para os alunos que o Scratch é similar ao Code.org e também faz o uso de blocos para construir algoritmos. Apresente aos alunos os tipos de blocos presentes no Scratch.
5. Arraste o bloco *Mova 10 passos* para a área do código junto com os alunos e peça que eles cliquem no bloco para eles verem o que acontece quando ele é executado. Explique que eles podem alterar o número de passos para fazer com que o personagem se mova mais. Peça que eles arrastem outros blocos de movimento e os testem da mesma forma (e.g.: vá para posição aleatória, deslize para posição aleatória, etc.).
6. Mostre aos alunos que os projetos do Scratch seguem o plano cartesiano e que eles podem indicar as coordenadas x e y em comandos como o *Vá para (x,y)* . Primeiro, peça que os alunos utilizem os valores $(0,0)$ e os pergunte a eles para onde o personagem vai. Eles perceberão que a posição $(0,0)$ corresponde ao centro do cenário. Peça que eles testem o bloco *Deslize para (x,y)* e testem diferentes valores para o x e o y .
7. Teste também alguns comandos de aparência tais como: *diga, pense, mude 25 ao efeito cor*, entre outros.
8. Em seguida, teste com os alunos o bloco *repita 10 vezes*. Arraste o bloco *repita 10 vezes* na área do código. Depois, mostre como combinar o bloco *repita 10 vezes* com o bloco de movimento *adicione 10 a x* . Os alunos estarão familiarizados com esta combinação uma vez que ela funciona da mesma forma que nas lições do Code.org. Teste o funcionamento com os alunos. Depois peça que eles mudem o valor 10 para -10 e pergunte a eles o que acontece.
9. Similarmente, peça que os alunos adicionem outro bloco *repita 10 vezes* e o combinem com o bloco de movimento *adicione 10 a y* . Assim como na etapa anterior, teste o valor 10 e -10 para eles verem o que acontece quando o sinal de menos é adicionado.
10. Durante o restante da aula, deixe com que os alunos explorem o Scratch e testem os comandos livremente. Busque ver o que eles estão tentando fazer e auxiliá-los caso necessário.

Aula 6 - Anime o Seu Nome

Nesta aula os alunos continuam praticando comandos simples do Scratch para manipular a aparência e animar as letras de seus nomes.

Objetivos	Aprender sobre os comandos de evento.
	Praticar a utilização de comandos básicos de movimento
	Praticar a utilização de comandos básicos de aparência
	Praticar a utilização de laços de repetição
	Praticar a utilização de eventos
	Criar animações simples para um conjunto de objetos
Atividades	Kahoot 3 - Plano Cartesiano e Scratch
	Scratch: Anime o seu Nome
Recursos e Materiais	Computador para o professor
	Projektor de vídeo
	Computadores para os alunos utilizarem
	Conexão à Internet
	Turma criada no Scratch caso a versão web do Scratch seja utilizada
	Conta dos alunos criadas e adicionadas a turma do Scratch
	Cartões de Votação
	Uma cópia da <i>Aula 6 - Roteiro</i> para cada aluno
	<i>Aula 6 - Apresentação</i>

Descrição da Aula

A Aula 6 está estruturada em quatro partes: *Revisão*, *Blocos de Eventos*, *Anime seu Nome* e *Apresentação e votação*.

Revisão

No momento inicial da aula, recomendamos também que os tópicos abordados na aula anterior sejam revisados com a turma. Criamos o Kahoot 3 - Plano Cartesiano e Scratch ⁴ para que o professor faça esta revisão. Ele contém exercícios de identificação de pontos similares àqueles da aula anterior e questões sobre alguns blocos de movimento do Scratch. Caso o professor não queira fazer a revisão através do Kahoot, também incluímos slides de revisão na *Aula 6 - Apresentação*.

Blocos de Eventos

Após conduzida a revisão com os alunos, o professor apresenta os blocos de eventos aos alunos e como eles são utilizados no Scratch. Deve ser explicado que estes blocos são utilizados para que *scripts* sejam executados apenas quando certos eventos ocorrem. Exemplos de eventos devem ser dados para que os alunos realmente entendam o que eles são. Alguns exemplos de principais eventos utilizados nos projetos do Scratch são: quando a bandeira verde for clicada, quando teclas do teclado são pressionadas, quando o personagem for clicado.

Fale com os alunos que alguns destes blocos de eventos serão utilizados por eles em seguida quando eles forem conduzir a atividade desta aula.

Anime seu Nome

Para a parte prática da aula conduza a atividade Anime seu Nome. Nesta atividade, os alunos devem criar animações para letras de seus nomes ou apelidos. As animações podem envolver a mudança da aparência das letras (alteração da cor, tamanho, etc.) ou a movimentação delas (girar, deslizar, etc.).

No primeiro momento da aula, introduza a atividade aos alunos. Diga sobre o que é a atividade, ou seja, **sobre a criação de animações para letras dos nomes dos alunos tais como mudança da aparência ou movimentação**. Em seguida, exiba no projetor de tela o vídeo tutorial⁵ para dar ideias e inspirar os alunos.

⁴[Link para o Kahoot 3 - Plano Cartesiano e Scratch](#)

⁵[Link para o Vídeo Tutorial](#)

Antes de iniciar a atividade, caso o professor queira dar uma motivação extra aos alunos, informe os alunos que os projetos serão compartilhados e apresentados para a turma ou no fim desta aula ou no começo da próxima. Diga também que uma votação será realizada para que os alunos escolham os projetos que eles mais gostaram. A única regra para votação é que os alunos não podem votar em seus próprios projetos. Leve brindes para presentear os alunos que desenvolveram os projetos mais legais segundo a turma.

Recomendação: *para facilitar o compartilhamento e visualização, sugere-se a criação de um Estúdio da turma para que os alunos coloquem seus projetos. Assim, o professor pode acessá-los e exibi-los para a turma através do projetor de tela. Uma competição para ver quem criou as animações mais interessantes, onde os alunos podem votar no projeto que acharem mais legal, também pode ser incluída como forma de engajar e motivar os alunos.*

Finalmente, o professor pede que os alunos entrem em suas contas do Scratch e criem um novo projeto para o Anime seu Nome. Enquanto eles estão acessando a ferramenta, distribua as folhas *Aula 6 - Roteiro* para os alunos. Diga a eles, que o roteiro contém diversas opções de animações que eles podem implementar para cada letra, porém frise que eles podem adicionar múltiplas animações para a mesma letra e que eles estão livres para combiná-las, modificá-las ou criar as suas próprias animações do zero.

Recomendação: *é recomendável que os roteiros sejam impressos em colorido para que os alunos consigam encontrar os blocos mais facilmente.*

Apresentação e votação

Caso haja tempo, no fim da aula o professor conduz a apresentação e votação dos projetos criados pelos alunos. Peça que todos os alunos salvem, compartilhem e adicionem seus projetos ao estúdio da turma, que foi previamente criado. Demonstre como eles devem fazer isso através do projetor de tela para que eles possam entender como fazer isto. Em seguida, o professor distribui para a turma os cartões de votação, lembrando os alunos que eles devem escrever seus nomes e que eles não podem votar em seus próprios projetos.

Inicie a apresentação dos projetos, abra um projeto de cada vez e execute os scripts criados pelo dono dos projetos. Pode ser necessário que o professor questione os alunos sobre o funcionamento do projeto durante a apresentação para que ele seja

executada da forma como foi visionada pelos alunos.

Após a apresentação de todos os projetos, peça para que os alunos marquem no cartão de votação os projetos que eles mais gostaram.

Obs.: *caso não seja possível realizar a apresentação, votação e contagem dos votos na Aula 6, as conduza rapidamente no início da aula seguinte.*

Aula 7 - Jogo de Clicar

Nesta aula os alunos continuam a praticar a utilização dos blocos básicos do Scratch através da criação de um jogo simples de clicar.

Objetivos	Praticar a utilização de comandos básicos de movimento
	Praticar a utilização de comandos básicos de aparência
	Praticar a utilização de laços de repetição
	Praticar a utilização de eventos
	Criar um jogo de clicar
Atividade	Scratch - Jogo de Clicar
Recursos e Materiais	Computador para o professor
	Projektor de vídeo
	Computadores para os alunos utilizarem
	Conexão à Internet
	Turma criada no Scratch caso a versão web do Scratch seja utilizada
	Conta dos alunos criadas e adicionadas a turma do Scratch
	Estúdio da turma para os alunos compartilharem seus projetos (recomendado)
	Uma cópia da <i>Aula 7 - Roteiro</i> para cada aluno
	<i>Aula 7 - Apresentação</i>

Descrição da Aula

Esta aula está estruturada em três partes: *Revisão e Apresentação de novos blocos de comando*, *Desenvolvimento das mecânicas básicas do jogo*, *Implementação de modificações*

Revisão e Apresentação de novos blocos de comando

O professor inicia a aula conduzindo uma pequena revisão através da *Aula 7 - Apresentação* para relembrar os alunos dos blocos de eventos introduzidos na aula anterior. O professor deve dar mais atenção para os seguintes eventos: *quando a bandeira verde for clicada* e *quando o ator for clicado*. Estes dois eventos deverão ser utilizados pelos alunos quando eles forem implementar o jogo durante o restante da aula.

Além da revisão dos blocos de eventos, incluímos também na *Aula 7 - Apresentação* um slide com os principais blocos que os alunos irão utilizar na aula: *mostre*, *esconda*, *vá para posição aleatória* e *espere 1 segundo*. O professor deve explicar o funcionamento destes blocos para que os alunos já tenham ideia de como utilizá-los quando eles forem seguir as tarefas da folha *Aula 7 - Roteiro*.

- **mostre e esconda:** faz o personagem aparecer e desaparecer respectivamente.
- **vá para posição aleatória:** muda a posição do personagem para um lugar aleatório do cenário.
- **espere 1 segundo:** interrompe a execução do script por 1 segundo. Quando um segundo for passado é dada continuação à execução. Este bloco deverá ser utilizado para fazer com que o personagem permaneça na tela por um segundo antes de desaparecer e com que ele permaneça escondido por um segundo antes de voltar a aparecer.

Desenvolvimento do funcionamento básicos do jogo

O professor agora explica o que os alunos desenvolverão na aula. Avance para o slide da *Aula 7 - Apresentação* que contém um vídeo demonstrando o funcionamento básico do jogo de clicar enquanto a explicação é dada. É importante indicar que **o jogo começa quando a bandeira verde é clicada**, que **o fantasma permanece por um segundo na tela depois de aparecer** e que **o fantasma fica escondido por um segundo depois que ele some**.

Após a explicação, o professor pede que os alunos entrem em suas contas e distribui a folha *Aula 7 - Roteiro* para eles enquanto eles estão fazendo login. O professor

deve falar com eles que eles devem tentar conduzir as tarefas do roteiro por conta própria e que só podem pedir ajuda depois de terem ao menos tentado implementar os passos ou solucionar problemas.

Recomendação: *corte a folha Aula 7 - Roteiro e separe as duas partes. Distribua apenas a PARTE I para os alunos inicialmente para evitar que eles tentem implementar os passos da PARTE 2 antes de concluírem os passos da PARTE II.*

Implementação de modificações

Na folha *Aula 7 - Roteiro*, foram propostas modificações extras para que os alunos que conseguirem finalizar a primeira parte do roteiro tentem implementá-las. Para os alunos que finalizarem a PARTE I, peça que eles tentem implementar algumas das modificações propostas na PARTE II.

Obs.: *o projeto com as modificações implementadas pode ser acessado através do seguinte link:*
[Aula 7 - Jogo de Clicar](#)

Caso um estúdio da turma tenha sido criado para os projetos da aula, peça que os alunos compartilhem seus projetos e os adicionem ao estúdio depois que eles terminarem a implementação do projeto.

Aula 8 - Introdução às Condicionais

Nesta aula os alunos aprendem sobre as estruturas condicionais e as colocam em prática através do desenvolvimento do Jogo da Coleta.

Objetivos	Aprender o que é uma condição no contexto da Computação.
	Entender o que acontece quando uma condição é satisfeita ou não.
	Aprender o que são as estruturas condicionais.
	Aprender sobre as estruturas condicionais no contexto do Scratch.
	Aprender sobre os sensores <i>tocando</i> e <i>tecla pressionada</i> .
	Praticar a utilização da estrutura condicional SE .
	Continuar a prática dos blocos de comandos previamente utilizados.
Atividade	Criar um jogo de coleta simples.
	Scratch - Jogo de Clicar
Recursos e Materiais	Computador para o professor
	Projektor de vídeo
	Computadores para os alunos utilizarem
	Conexão à Internet
	Turma criada no Scratch caso a versão web do Scratch seja utilizada
	Conta dos alunos criadas e adicionadas a turma do Scratch
	Estúdio da turma para os alunos compartilharem seus projetos (recomendado)
	Uma cópia da <i>Aula 8 - Roteiro</i> para cada aluno (recomenda-se que seja impresso colorido)
	<i>Aula 8 - Apresentação</i>

Descrição da Aula

A Aula 8 foi estruturada em 3 partes: *Introdução às estruturas condicionais* e *Implementação do Jogo da Coleta*.

Introdução às estruturas condicionais

Na parte introdutória da Aula 8, o professor apresenta o que são condições e as estruturas condicionais aos alunos e os questiona sobre o que é uma *condição* e se eles conseguem pensar em algum exemplo de condição.

Em seguida, o professor conduz a *Dinâmica da Palma*. A dinâmica é uma atividade simples e tem como objetivo fazer com que os alunos pratiquem a verificação de uma condição e que reajam de acordo com o resultado da verificação. A dinâmica foi dividida em duas partes: a primeira faz o uso de uma condicional única (e.g.: se o número na tela for maior que 5, bata uma palma), já a segunda faz o uso de duas condicionais (e.g.: se o número na tela for maior que 5, bata uma palma. Senão, bata duas palmas). A apresentação *Aula 8 - Apresentação* possui cinco números para cada parte da dinâmica. O número 5 está presente em ambas as partes com o objetivo de confundir os alunos, pois ele não é nem menor e nem maior que 5, sendo este um caso que não cai em nenhuma das condições. Desta forma, os alunos não devem nenhuma palma em ambas as partes da dinâmica.

Para finalizar a parte introdutória, o professor dá alguns exemplos de usos das condicionais referenciando as ferramentas que eles utilizaram no curso até o momento tais como:

- **Login no Site do Scratch:** verificação se o usuário e a senha estão corretos. Se estiver correto, a plataforma deixa o usuário acessar a conta. Senão, uma mensagem de erro é exibida informando o usuário que as credenciais informadas estão erradas.
- **Kahoot!:** Verificar SE as respostas dadas estão corretas ou não.
- **Code.org:** Verificar SE o algoritmo está correto ou SE o número de blocos utilizados é o mínimo.

E depois, apresenta brevemente os blocos que serão utilizados pelos alunos para implementação do Jogo Pong durante o restante da aula.

Implementação do Jogo da Coleta

Agora, o professor apresenta o que os alunos estarão utilizando e desenvolvendo na parte prática da aula. Os novos blocos a serem utilizados nesta aula são os seguintes:

- **bloco SE:** condicional simples que verifica se uma condição é verdadeira.
- **sensor TOCANDO EM:** indica se o objeto está tocando em algo.
- **sensor TECLA PRESSIONADA:** indica se determinada tecla está sendo pressionada.

Similarmente a aula anterior, antes de pedir para os alunos acessarem suas contas do Scratch e distribuir as folhas *Aula 8 - Roteiro*, o professor explica aos alunos o projeto que eles desenvolverão na aula com o auxílio de uma animação demonstrando o projeto final em funcionamento. O professor deve relacionar o que está acontecendo nas animações com os blocos apresentados previamente para que os alunos tenham uma ideia inicial do que eles devem fazer.

O professor deve encorajar os alunos a tentar ao máximo implementar os passos e resolver os problemas que eles enfrentarem sozinhos.

Nas folhas *Aula 8 - Roteiro* foram incluídas algumas modificações que podem ser implementadas pelos alunos que terminarem de implementar as funcionalidades básicas do projeto.

Obs.: o projeto implementado pode ser acessado através do seguinte link: [Aula 8 - Coleta](#)

Aula 9 - Condicionais e Pong

Nesta aula os alunos continuam a prática e utilização das estruturas condicionais através do jogo Pong, onde eles devem controlar uma raquete para não deixar que uma bola caia no chão.

Objetivos	Aprender o que é uma condição no contexto da Computação
	Entender o que acontece quando uma condição é satisfeita ou não
	Aprender o que são as estruturas condicionais
	Aprender sobre as estruturas condicionais no contexto do Scratch
	Praticar a utilização da estrutura condicional SE
	Continuar a prática dos blocos de comandos previamente utilizados
	Criar um jogo do Pong
Atividade	Scratch - Jogo de Clicar
Recursos e Materiais	Computador para o professor
	Projektor de vídeo
	Computadores para os alunos utilizarem
	Conexão à Internet
	Turma criada no Scratch caso a versão web do Scratch seja utilizada
	Conta dos alunos criadas e adicionadas a turma do Scratch
	Estúdio da turma para os alunos compartilharem seus projetos (recomendado)
	Uma cópia da <i>Aula 9 - Roteiro</i> para cada aluno (recomenda-se que seja impresso colorido)
	<i>Aula 9 - Apresentação</i>

Descrição da Aula

A Aula 9 foi estruturada em 3 partes: *Revisão e Implementação do Jogo Pong*.

Revisão

O professor inicia a aula conduzindo uma rápida revisão das condicionais e suas finalidades e os blocos que os alunos utilizaram na aula anterior pela primeira vez.

Em seguida, devem ser introduzidos os blocos que serão utilizados pelos alunos para implementação do Jogo Pong durante o restante da aula e o próprio jogo através do vídeo ilustrativo no fim da *Aula 9 - Apresentação*.

Implementação do Jogo Pong

O professor avança para o último slide da *Aula 9 - Apresentação* para apresentar e explicar o jogo que os alunos desenvolverão na parte prática da aula. Finalmente após a apresentação do projeto, o professor pede para que os alunos acessem suas contas e entregue as folhas *Aula 8 - Roteiro* para cada um deles. Similarmente a aula anterior, frise novamente que os alunos devem tentar ao máximo desenvolver, testar, resolver os problemas sozinhos.

Caso um estúdio da turma tenha sido criado para os projetos da aula, peça que os alunos compartilhem seus projetos e os adicionem ao estúdio depois que eles terminarem a implementação do projeto.

Obs.: o projeto implementado pode ser acessado através do seguinte link: [Aula 8 - Pong](#)

Aula 10 - Introdução às Variáveis

Nesta aula os alunos aprendem sobre variáveis e praticam a sua utilização ao desenvolver um sistema de pontuação no Scratch para o *Jogo de Clicar*, *Jogo da Coleta* e o *Pong*.

Objetivos	Aprender o que são variáveis
	Entender quando variáveis devem ser utilizadas
	Aprender sobre os blocos de manipulação de variáveis no Scratch
	Praticar a manipulação de variáveis para armazenar pontos
	Continuar a prática dos blocos de comandos previamente utilizados
Atividade	Scratch - Jogo de Clicar e Pong
Recursos e Materiais	Computador para o professor
	Projector de vídeo
	Computadores para os alunos utilizarem
	Conexão à Internet
	Turma criada no Scratch caso a versão web do Scratch seja utilizada
	Conta dos alunos criadas e adicionadas a turma do Scratch
	Uma cópia da <i>Aula 10 - Roteiro</i> para cada aluno (recomenda-se que seja impresso colorido)
	<i>Aula 10 - Apresentação</i>

Descrição da Aula

Estruturamos esta aula em 2 partes: *Introdução às variáveis*, *Implementando sistemas de pontuação*.

Introdução às variáveis

Antes do professor apresentar o conceito de variáveis e suas utilizações, é interessante que uma rápida revisão dos principais scripts que os alunos tiveram que desenvolver na aula anterior seja conduzida. Na *Aula 10 - Apresentação*, foram incluídos os scripts responsáveis pela configuração do estado inicial do jogo quando a bandeira verde é clicada, pela movimentação da bola e por fazer com que a bola quique na raquete (i.e.: paddle).

Em seguida, o professor apresenta a definição formal de variáveis, indicando que elas são responsáveis pelo armazenamento de informações. O professor diz que as variáveis serão utilizadas pelos alunos na aula para que eles criem um sistema de pontuação para ambos o jogo de clicar, jogo da coleta e o pong desenvolvidos nas aulas anteriores. O restante dos slides na *Aula 10 - Apresentação* contém o passo a passo que os alunos devem seguir para criar, renomear, exibir o valor das variáveis na tela.

Por fim, o professor introduz os dois blocos de comando para fazer a manipulação de variáveis que os alunos utilizarão na atividade prática (Figura A.2).



Figura A.2: Comandos que os alunos utilizarão para manipular as variáveis

O professor explica o funcionamento dos comandos *adicione [valor] a [variável]* e *mude [variável] para [valor]*, indicando que os alunos podem digitar valores nos campos em branco e que eles podem clicar em *minha variável* para selecionar qual variável será manipulada pelo comando.

Implementando sistemas de pontuação

Finalmente, o professor apresenta aos alunos o que eles farão na aula utilizando o último slide da *Aula 10 - Apresentação* para ilustrar o funcionamento das variáveis e quando elas são incrementadas durante os jogos.

Na atividade prática, os alunos terão que incrementar os valores das variáveis quando o personagem for clicado no *Jogo de Clicar*, quando o personagem coletar o objeto no *Jogo da Coleta* e quando a bola tocar na raquete no *Pong*. Além disso, eles terão que mudar o valor das variáveis para 0 quando o jogo for iniciado, caso contrário, o jogo é iniciado com o valor da variável armazenado através de execuções anteriores do jogo.

O professor, então, distribui as folhas *Aula 10 - Roteiro* aos alunos para eles darem início a atividade prática enquanto eles estão acessando suas contas.

Caso alguns dos alunos não tenham conseguido terminar a implementação do Pong na aula anterior, os ajude rapidamente após o início da atividade prática.

Recomendação: *a princípio, distribua apenas as partes 1, 2 E 3 da primeira folha do roteiro aos alunos. Quando os alunos terminarem estas duas partes, distribua a parte MODIFICAÇÕES da primeira folha. E finalmente, a segunda folha do roteiro com outras sugestões para que eles tentem implementar algumas delas.*

Obs.: *Acesse os projetos através seguintes links:*

[Aula 10 - Jogo de Clicar com Pontuação](#)

[Aula 10 - Jogo da Coleta com Pontuação](#)

[Aula 10 - Pong com Pontuação](#)

Aula 11 - Perguntas e Respostas I

Nesta aula os alunos aprendem a utilizar os sensores *pergunte* e *resposta* para criar um jogo de perguntas e respostas simples com perguntas e respostas específicas.

Objetivos	Aprender sobre os sensores <i>pergunte</i> e <i>resposta</i>
	Praticar a utilização do se e senão
	Continuar a prática dos blocos de comandos previamente utilizados
	Criar um jogo de perguntas e respostas simples
Atividade	Scratch - Jogo de Perguntas e Respostas I
Recursos e Materiais	Computador para o professor
	Projetor de vídeo
	Computadores para os alunos utilizarem
	Conexão à Internet
	Turma criada no Scratch caso a versão web do Scratch seja utilizada
	Conta dos alunos criadas e adicionadas a turma do Scratch
	Uma cópia da <i>Aula 11 - Roteiro</i> para cada aluno (recomenda-se que seja impresso colorido)
<i>Aula 11 - Apresentação</i>	

Descrição da Aula

Estruturamos esta aula em 2 partes: *Revisão sobre variáveis*, *Introdução aos Sensores* e *Criando um jogo de perguntas e respostas simples*.

Revisão sobre variáveis

O professor conduz nos minutos iniciais da aula uma pequena revisão das variáveis e dos dois blocos de manipulação de variáveis utilizadas na aula anterior para implementar o sistema de pontuação para os três projetos.

Introdução aos Sensores

O professor explica brevemente o que são os sensores no Scratch, lembrando os alunos dos sensores que eles utilizaram nas aulas anteriores: *tocando em* e *tecla pressionada*.

Em seguida, o professor apresenta dois novos sensores que serão utilizados na aula: *pergunte* e *resposta*. O sensor *pergunte* faz o personagem dizer algo e logo em seguida exibe uma caixa de texto para que a resposta seja digitada. A resposta digitada pelo usuário é então armazenada no sensor *resposta*, sendo que este sensor é basicamente uma variável especial no Scratch que receberá o que foi digitado pelo usuário na caixa de texto da pergunta.

Criando um jogo de perguntas e respostas simples

Por fim, o professor apresenta à turma a animação de um jogo de perguntas e respostas contendo duas perguntas sobre a Computação. O professor deve chamar atenção para a caixa de texto que aparece no jogo, onde o usuário digita as respostas, e que a resposta digitada deve ser exatamente a mesma que a resposta correta, ou seja, a inclusão de acentos é necessária caso a resposta correta contenha acentos gráficos.

O último slide da *Aula 11 - Apresentação* contém um link com perguntas da disciplina de ciências, juntamente com suas respostas, para que os alunos utilizem caso eles queiram. No entanto, o professor pode encorajá-los a incluir perguntas relacionadas a temas de interesse, sendo elas relacionadas às disciplinas da escola ou não.

Finalmente, as folhas *Aula 11 - Roteiro* são distribuídas aos alunos e eles acessam suas contas para dar início a atividade prática. Da mesma forma que nos roteiros das aulas anteriores, propomos algumas modificações a serem implementadas para os alunos que conseguirem finalizar o desenvolvimento do jogo antes da aula terminar. As modificações propostas não são dependentes, então o professor pode deixar que os

alunos escolham quais delas eles querem tentar fazer. As seguintes modificações foram propostas:

- fazer o personagem introduzir o jogo.
- perguntar ao jogador se ele está pronto ou não para iniciar o jogo.
- adicionar cenários diferentes para cada pergunta.
- adicionar um sistema de pontuação
- fazer o jogador perder se ele errar alguma pergunta.

Recomendação: *separe as modificações das tarefas básicas da aula e distribua as modificações apenas para os alunos que terminaram as duas primeiras partes do roteiro.*

Obs.: *o projeto de exemplo com as modificações implementadas pode ser acessado através do seguinte link:*

[Aula 11 - Perguntas e Respostas I](#)

Aula 12 - Perguntas e Respostas II

Nesta aula os alunos desenvolvem um jogo de perguntas e respostas dinâmicas para praticar a utilização de variáveis para fazer as perguntas e fazer a verificação das respostas.

Objetivos	Aprender sobre os sensores <i>pergunte</i> e <i>resposta</i>
	Aprender sobre os operadores <i>*</i> , <i>=</i> , <i>junte</i> , <i>números aleatórios</i>
	Praticar a utilização do se e senão
	Continuar a prática dos blocos de comandos previamente utilizados
	Criar um jogo de perguntas e respostas para prática de tabuada
Atividade	Scratch - Jogo de Perguntas e Respostas II
Recursos e Materiais	Computador para o professor
	Projector de vídeo
	Computadores para os alunos utilizarem
	Conexão à Internet
	Turma criada no Scratch caso a versão web do Scratch seja utilizada
	Conta dos alunos criadas e adicionadas a turma do Scratch
	Uma cópia da <i>Aula 13 - Roteiro</i> para cada aluno (recomenda-se que seja impresso colorido)
	<i>Aula 13 - Apresentação</i>

Descrição da Aula

Estruturamos esta aula em 3 partes: *Revisão*, *Introdução aos Operadores* e *Criando um jogo de perguntas e respostas dinâmicas*.

Revisão

O professor inicia a aula com uma curta revisão sobre os sensores *pergunte* e *resposta* utilizados na aula anterior.

Introdução aos Operadores

O professor explica a funcionalidade dos operadores que os alunos terão que utilizar na aula para criar o jogo de perguntas e respostas de tabuada: *igual* ($=$), *vezes* ($*$), *número aleatório entre* e *junte*. Destes blocos, o bloco *junte* é o mais complicado de entender e utilizar, podendo ser confuso para os alunos visualizarem o que está acontecendo quando ele é utilizado.

Para facilitar o entendimento por parte dos alunos em relação aos operadores e a como eles podem ser combinados, foram incluídos slides com exemplos de combinação dos operadores *igual* ($=$), *vezes* ($*$) e *junte*. Além disso, para o operador *junte*, incluímos dois exemplos de uso no Scratch para que os alunos pensem em qual será o resultado.

Criando um jogo de perguntas e respostas dinâmicas

Após apresentar os operadores e exemplos de usos aos estudantes, o professor introduz o jogo que eles criarão. O professor deve fazer um paralelo entre o jogo de perguntas e respostas desenvolvido na aula passada com o jogo de perguntas e respostas que será desenvolvido nesta aula. A grande diferença é que no jogo da Aula 11, as perguntas eram específicas e sempre as mesmas e, consequentemente, também as respostas também eram. Contudo, no jogo da Aula 12, as perguntas sendo exibidas dependerão dos valores aleatórios armazenados nas variáveis, logo, tanto as perguntas quanto as respostas terão valores diferentes a cada iteração do jogo. Desta forma, a verificação das respostas deve ser feita de tal forma que ela funcionará para quaisquer sejam os valores das variáveis. Utilize a animação do jogo incluída na *Aula 12 - Apresentação* para ilustrar o funcionamento do jogo para os alunos.

Por fim é feita a distribuição das folhas *Aula 12 - Roteiro* aos alunos para que eles deem início ao desenvolvimento da primeira versão do jogo. Para o roteiro da Aula 12, nós propomos três versões do jogo de perguntas e respostas de matemática,

sendo que a primeira é a mais simples de ser implementada. Na versão I, os alunos devem fazer o personagem perguntar qual o resultado da multiplicação de um número aleatório com um número fixo escolhido por eles. Na segunda versão, o personagem deve perguntar o resultado da multiplicação de dois números aleatórios. E por fim, na terceira versão, o personagem primeiro pergunta ao usuário qual a tabuada que ele quer praticar para que o personagem sempre pergunte o resultado da multiplicação de um número aleatório com o número informado pelo usuário.

Recomendação: *distribua a folha com as modificações apenas quando os alunos terminarem de implementar a primeira versão do jogo.*

Obs.: *As três versões do Jogo de Perguntas e Respostas de Matemática podem ser acessada através dos links abaixo:*

[Aula 12 - Perguntas e Respostas v1](#)

[Aula 12 - Perguntas e Respostas v2](#)

[Aula 12 - Perguntas e Respostas v3](#)

Aula 13 - Jogo de Desviar

Nesta aula os alunos aprendem sobre clones no Scratch e desenvolvem um jogo em que eles devem controlar um personagem para evitar objetos caindo do céu. Estes objetos são criados utilizando os blocos responsáveis por gerenciar os clones.

Objetivos	Aprender sobre os blocos de gerenciamento de clones
	Continuar a prática dos blocos de comandos previamente utilizados
	Criar um jogo de desviar para praticar a utilização dos blocos de clones
Atividade	Scratch - Jogo de Perguntas e Respostas II
Recursos e Materiais	Computador para o professor
	Projektor de vídeo
	Computadores para os alunos utilizarem
	Conexão à Internet
	Turma criada no Scratch caso a versão web do Scratch seja utilizada
	Conta dos alunos criadas e adicionadas a turma do Scratch
	Uma cópia da <i>Aula 13 - Roteiro</i> para cada aluno (recomenda-se que seja impresso colorido)
	<i>Aula 13 - Apresentação</i>

Descrição da Aula

Estruturamos esta aula em 2 partes: *Introdução aos Clones* e *Criando um jogo de desviar*.

Introdução aos Clones

O professor apresenta aos alunos o conceito de clones no contexto do Scratch. Os clones são cópias de personagens ou objetos, que apresentarão o mesmo comportamento após serem criados. Por exemplo, no jogo de desviar, o jogador controla um personagem para evitar os objetos caindo do céu. Esses objetos sempre farão as mesmas coisas, logo, os blocos para gerenciar clones podem ser utilizados para criar esses objetos em tempo real enquanto o jogo está sendo executado.

Os blocos de gerenciamento de clones a serem apresentados são:

- **crie clone de [este ator]:** cria um clone do ator onde este bloco é utilizado
- **quando eu começar como um clone:** utilizado para programar o que o clone faz depois que ele é criado. Note que a forma deste bloco é a mesma do bloco *Quando a bandeira verde for clicada*. Chame a atenção dos alunos para isso e diga que ele funciona de maneira similar.
- **apague este clone:** exclui o clone em questão

Criando um jogo de perguntas e respostas dinâmicas

Após o conceito de clones e os blocos para gerenciá-los serem apresentados aos estudantes, o professor introduz o jogo que será criado na aula. Utilize a animação do jogo incluída na *Aula 13 - Apresentação* para ilustrar o funcionamento do jogo para os alunos.

Obs.: o personagem no jogo será movimentado da mesma forma que o personagem que eles programaram no *Jogo da Coleta* na *Aula 8*. Assim, acrescentamos um slide após o slide ilustrando o funcionamento do jogo para relembrar os alunos do funcionamento do script que controla o personagem.

Por fim é feita a distribuição das folhas *Aula 13 - Roteiro* aos alunos para que eles deem início ao desenvolvimento do jogo de desviar. O roteiro foi dividido em três seções. Na primeira seção, os alunos programam o objeto que será clonado. Na segunda seção, os alunos programam o personagem que será controlado pelo jogador. Por fim, na última seção, propomos algumas modificações para o jogo como a adição de tempo

para o jogador saber quando tempo ele conseguiu sobreviver, a adição de outros objetos que devem ser evitados e a modificação de como os objetos se movimentam no cenário.

Obs.: *As versões do Jogo de Desviar implementados podem ser acessadas através dos seguintes links:*

[Aula 13 - Jogo de Desviar](#)

[Aula 13 - Jogo de Desviar com Modificações](#)

Aula 14 - Árvores de Decisão e Estórias I

Nesta aula os alunos aprendem sobre as árvores de decisão e as colocam em prática na definição de um roteiro de uma estória a ser implementada no Scratch.

Objetivos	Aprender as árvores de decisão
	Criar roteiro para uma estória simples com dois tipos de finais utilizando as árvores de decisão
	Aprender sobre os blocos de transmissão e recepção de mensagens
	Continuar a prática dos blocos de comandos previamente utilizados
	Criar uma estória animada utilizando o roteiro criado previamente
Atividades	Criando roteiro para estória
	Scratch - Estória
Recursos e Materiais	Computador para o professor
	Projetor de vídeo
	Computadores para os alunos utilizarem
	Conexão à Internet
	Turma criada no Scratch caso a versão web do Scratch seja utilizada
	Conta dos alunos criadas e adicionadas a turma do Scratch
	Uma cópia da <i>Aula 14 - Árvore de Decisão</i> para cada aluno
	<i>Aula 14 - Apresentação</i>

Descrição da Aula

Esta aula está estruturada em 4 partes: *introdução às árvores de decisão, criando um roteiro para estória, introdução à transmissão de mensagens no Scratch, criando uma estória animada no Scratch.*

Introdução às árvores de decisão

O professor apresenta aos alunos o conceito de árvores de decisão e demonstra um exemplo de aplicação de árvore de decisão para criação de um roteiro de uma estória. Nesse exemplo, é mostrado a interação entre um morcego e um urso. O urso faz perguntas ao morcego e o fluxo da estória é guiado pelas respostas dadas pelo morcego.

Criando um roteiro para estória

Nesta parte da aula, o professor distribuí uma cópia da folha *Aula 14 - Árvore de Decisão* aos alunos para que eles definam uma estória simples, utilizando a estrutura da árvore de decisão apresentada na primeira parte da aula.

O professor deve dizer aos alunos que a estória dos alunos devem acontecer de forma similar ao exemplo. Ou seja, os alunos devem pensar em uma estória com dois personagens, onde um dos personagens faz uma pergunta ao outro. O personagem pode dar duas respostas e cada resposta leva a um fim diferente.

Recomendação.: *volte ao slide que ilustra o exemplo da árvore de decisão para que os alunos o consulte caso eles tenham dúvidas*

Introdução à Transmissão de Mensagens no Scratch

Na terceira parte da aula, o professor introduz dois blocos de eventos, os quais serão necessários para que os alunos implementem suas estórias no Scratch. Os blocos a serem apresentados são:

- **transmita [mensagem]:** transmite determinada mensagem aos objetos dos projetos
- **quando eu receber [mensagem]:** utilizado similarmente ao blocos *quando bandeira verde for clicada* para implementar o que o personagem fará quando ele receber determinada mensagem

A *Aula 14 - Apresentação* ilustra os blocos e como o usuário pode criar novas mensagens e nomeá-las. Exemplos de utilização dos blocos para implementar a per-

gunta final do lado esquerdo da árvore de decisão da estória do urso e morcego também foram incluídos na apresentação para facilitar o entendimento dos alunos.

Criando uma estória animada no Scratch

Finalmente, na última parte da aula os alunos devem implementar suas respectivas estórias no Scratch. Deixe os últimos slides da *Aula 14 - Apresentação* com os exemplos de implementação visíveis para que os alunos consultem caso necessário.

Obs.: *não há roteiros para esta parte dado que as estórias dos alunos serão diferentes. Assim, os alunos podem ter que implementar diferentes tipos de interações (e.g.: mudança de cenários, inclusão de sons, inclusão de animações) ou incluir outros personagens/objetos que também deverão transmitir ou receber mensagens.*

Obs.: *As implementação da estória do urso e do morcego podem ser acessadas através dos seguintes links:*

[Aulas 14 e 15 - Estória Simplificada](#)

[Aulas 14 e 15 - Estória Completa](#)

Aula 15 - Árvores de Decisão e Estórias II

Nesta aula é dada continuação a implementação e extensão da estória animada dos alunos. É esperado que todos os alunos consigam ao menos finalizar a parte simplificada da estória, ou seja, estória com uma pergunta e dois fins.

Objetivos	Aprender as árvores de decisão
	Criar roteiro para uma estória simples com dois tipos de finais utilizando as árvores de decisão
	Aprender sobre os blocos de transmissão e recepção de mensagens
	Continuar a prática dos blocos de comandos previamente utilizados
	Criar uma estória animada utilizando o roteiro criado previamente
Atividade	Criando roteiro para estória
Recursos e Materiais	Computador para o professor
	Projector de vídeo
	Computadores para os alunos utilizarem
	Conexão à Internet
	Turma criada no Scratch caso a versão web do Scratch seja utilizada
	Conta dos alunos criadas e adicionadas a turma do Scratch
	Roteiros das estórias criados pelos alunos na Aula 14
<i>Aula 15 - Apresentação</i>	

Descrição da Aula

A Aula 15 foi dividida em 2 partes: *revisão* e *continuação a implementação da estória*

Revisão

Nos minutos iniciais da aula o professor faz uma revisão dos tópicos apresentados na aula anterior: árvores de decisão, blocos de transmissão de mensagens e os exemplos de uso dos blocos.

Continuação a implementação da estória

Em seguida, o professor distribui os roteiros das estórias definidas pelos alunos na aula anterior e pede que eles deem continuidade às atividades iniciadas na aula anterior. Os alunos devem finalizar a implementação da estória simplificada definida por eles na Aula 14.

Para os alunos que conseguirem terminar a implementação de suas estórias simples, disponibilize a implementação completa da estória do urso e morcego para que os alunos testem e, em seguida, peça que eles tentem estender seus projetos e deixá-los mais sofisticados. Algumas sugestões que o professor pode dar são:

- inclusão de continuações para a estória depois dos dois fins originais
- inclusão de cenário para representar o fim da estória
- inclusão de diálogos iniciais entre os personagens da estória

Obs.: *As implementação da estória do urso e do morcego podem ser acessadas através dos seguintes links:*

[Aulas 14 e 15 - Estória Simplificada](#)

[Aulas 14 e 15 - Estória Completa](#)

Apêndice B

Questionário de Perfil

1. Nome: _____
2. Data de Nascimento: _____
3. Antes deste GTD, você já tinha utilizado o computador?
☐ sim ☐ não
4. Você tem computador em casa (pode ser de mesa - desktop, ou móvel - notebook ou laptop)
☐ sim ☐ não
5. Antes deste GTD, onde você normalmente utilizava o computador?
☐ Em casa
☐ Casa de amigos
☐ Casa de parentes
☐ Escola
☐ Outro: _____
6. Quantos dias por SEMANA você normalmente utiliza o computador FORA da escola?
☐ Menos de 3 dias na semana (Apenas no fim de semana)
☐ Menos de 3 dias na semana (Qualquer dia da semana)
☐ De 3 a 4 dias na semana
☐ De 5 a 6 dias na semana
☐ Todos os dias
7. Para quê você normalmente usa o computador?
☐ Estudar
☐ Fazer trabalhos da escola (das matérias)
☐ Atividades do GTD
☐ Jogar
☐ Assistir vídeos
☐ Ler notícias
☐ Conversar com amigos e familiares
☐ Redes sociais (facebook, instagram, snapchat, twitter, etc.)
☐ Assistir vídeos
☐ Outro: _____

8. Para qual dessas atividades você MAIS usa o computador?

- ☐ Estudar
- ☐ Fazer trabalhos da escola (das matérias)
- ☐ Atividades do GTD
- ☐ Jogar
- ☐ Assistir vídeos
- ☐ Ler notícias
- ☐ Conversar com amigos e familiares
- ☐ Redes sociais (facebook, instagram, snapchat, twitter, etc.)
- ☐ Assistir vídeos
- ☐ Outro: _____

9. Você tem celular?

- ☐ sim ☐ não

10. Você usa o celular?

- ☐ Não
- ☐ Sim, eu tenho meu próprio celular
- ☐ Sim, eu uso celular dos meus pais ou irmãos
- ☐ Sim, eu uso celular de outros parentes (primos, tios, avós)
- ☐ Sim, eu uso celular de amigos
- ☐ Outro: _____

11. Quais destas atividades você normalmente faz no celular?

- ☐ Estudar
- ☐ Fazer trabalhos da escola (das matérias)
- ☐ Atividades do GTD
- ☐ Jogar
- ☐ Assistir vídeos
- ☐ Ler notícias
- ☐ Conversar com amigos e familiares
- ☐ Redes sociais (facebook, instagram, snapchat, twitter, etc.)
- ☐ Assistir vídeos
- ☐ Outro: _____

Apêndice C

Questionário de Avaliação

1. Atividade: _____

2. Data: _____

3. Nome (opcional): _____

4. Sexo: ☐ M ☐ F

5. Quanto você gostou da atividade?

☐ Detestei ☐ Não gostei ☐ Mais ou menos ☐ Gostei ☐ Adorei

6. O que você achou da atividade?

☐ Muito difícil ☐ Difícil ☐ Nem fácil e nem difícil ☐ Fácil ☐ Muito fácil

6. Comentário sobre a atividade ou sugestões:

Apêndice D

Roteiro: Grupos Focais 2019/01

Conhecimento Prévio

Algoritmos e Programação (antes do GTD)

O que sabiam, como aprenderam, quais sistemas conheciam

Atividades e Ferramentas

Atividades que gostou mais / Motivos

Atividades que gostou menos / Motivos

Sobre as outras, quais gostou ou não

Discutir atividades não mencionadas pelos alunos. Mostrar figuras se necessário

Aprendizado

Quanto acha que aprendeu no GTD?

Falar 2 coisas que não sabia e aprendeu (achou interessante)

Sobre o GTD

Expectativas antes de escolher o GTD

Teve coisas que queria ter visto e não viu

Se fosse continuar o GTD:

Teria interesse em continuar / Motivos

Se sim, que sugestões teria

Apêndice E

Roteiro: Grupos Focais 2019/02

Conhecimento Prévio

Expectativas antes de escolher o GTD

Algoritmos e Programação (antes do GTD)

O que sabiam, como aprenderam, quais sistemas conheciam

Atividades e Ferramentas

Atividades que gostou mais / Motivos

Atividades que gostou menos / Motivos

Sobre as outras, quais gostou ou não?

Discutir atividades não mencionadas pelos alunos. (Neste momento se precisar pode mostrar as figuras)

Alguma atividade foi muito difícil?

Aprendizado

Quanto acha que aprendeu no GTD?

Falar 2 coisas que não sabia e aprendeu (achou interessante)

Roteiros (atividades do Scratch)

Em geral, eram simples de entender e seguir

Acharam que foi útil para realizar a atividade?

O que faziam se não entendiam um passo? Seguiam o passo assim mesmo?

Sobre o GTD

No futuro você gostaria de aprender mais sobre pensamento computacional, programação, ou algum outro assunto relacionado a computação?

SE SIM: O quê? Por quê?

SE NÃO: Por quê?

Teve coisas que queria ter visto e não viu?

Se fosse continuar o GTD:

Teria interesse em continuar / Motivos

SE SIM: que sugestões teria

Apêndice F

Roteiro: Entrevista com alunos que não continuaram participando

Quais matérias você mais gosta na escola?

Por que você mais gosta?

Quais matérias você menos gosta na escola?

Por que você menos gosta?

Quais GTDs você está fazendo esse semestre?

Você está fazendo os GTDs que queria? Que marcou como primeira opção?

SE SIM: Por que você escolheu eles?

SE NÃO: Quais você queria? Por que?

Quanto ao GTD de PC I, você gostou de ter participado?

Você acha que foi fácil ou difícil? Quais dificuldades você teve?

Por que você não quis continuar no GTD de PC II?

Foi porque quis fazer outros GTDs? Foi porque não gostou?

O que você leva em consideração para escolher os GTDs?

Seus amigos estão em quais GTDs? Nos mesmos que você?

Você indicou o GTD de pensamento computacional para algum colega?

SE SIM: Para quem? O que você falou para ele?

SE NÃO: Por que você não indicou?

No futuro você gostaria de aprender mais sobre pensamento computacional, programação, ou algum outro assunto relacionado a computação?

SE SIM: O quê? Por que?

SE NÃO: Por que?

Apêndice G

Roteiro: Entrevista com os alunos que continuaram participando

Sobre os Alunos

Disciplinas que gostam. Por quê?

Disciplinas que não gostam. Por quê?

GTDs que estão fazendo.

Você está fazendo os GTDs que queria? Tinha outros que teria gostado de fazer?

SE SIM: Por que você escolheu eles?

SE NÃO: Quais você queria? Por quê?

O que você leva em consideração para escolher os GTDs?

Seus amigos estão em quais GTDs? Nos mesmos que você?

Sobre as Atividades

Atividades que chamaram que gostaram mais. Por quê?

Atividades que chamaram que gostaram menos. Por quê?

Discutir atividades não mencionadas pelos alunos. (Neste momento se precisar pode mostrar as figuras)

Revisão, Jogo de Desviar, Jogo de Perguntas, Jogo de Perguntas de Matemática, Projeto próprio, Estória

Vocês acham que os roteiros das atividades ajudaram? Por quê?

Estratégias utilizadas para resolver os problemas dados no GTD.

O que faziam quando tinham dúvidas sobre o que fazer?
Tentativa e erro?
Pensar no problema e em como resolvê-lo?
Pedir ajuda? Você acha que pedia ajuda mesmo antes de pensar em como resolver o problema? E antes de tentar fazer sozinho?

Aprendizado

Quanto acha que aprendeu no GTD?

O que você não sabia e aprendeu?
O que achou interessante? Por quê?
Saberia explicar para um colega que não fez o GTD? (Como explicaria?)

Sobre o GTD 1 e GTD 2

O que você achou do GTD 2 em relação ao GTD 1?

Diversão
Dificuldade

Se fosse ter de novo o GTD, você o indicaria para algum colega?

O que diria a ele sobre o GTD?
Gostou de ter continuado no GTD 2? Por que?
Sugestões e comentários como tornar o GTD mais legal

Perspectivas sobre o Futuro

O que quer ser quando crescer?

SIM: o que você quer ser
Por quê?

No futuro você gostaria de aprender mais sobre pensamento computacional, programação, ou algum outro assunto relacionado a computação?

SE SIM: O quê? Por quê?
SE NÃO: Por quê?