



Universidade Federal do Rio de Janeiro

Rubens Lacerda Queiroz

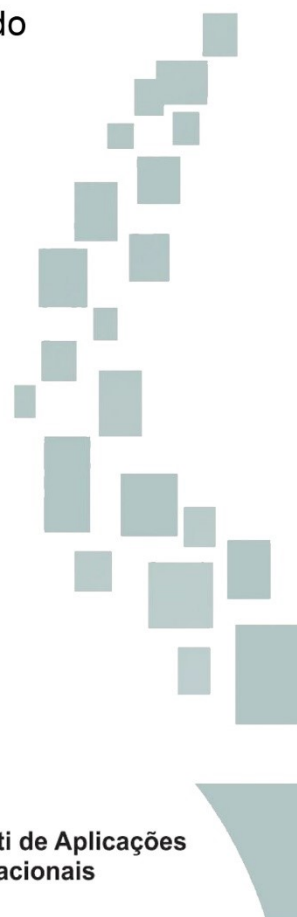
DISSERTAÇÃO DE MESTRADO

DuinoBlocks4Kids

Utilizando Tecnologia Livre e materiais de baixo custo para o exercício do Pensamento Computacional no Ensino Fundamental I por meio do aprendizado de programação aliado à Robótica Educacional.

Rio de Janeiro

2017





PPGI PROGRAMA
DE PÓS-GRADUAÇÃO
EM INFORMÁTICA

Universidade Federal do Rio de Janeiro

Rubens Lacerda Queiroz

DuinoBlocks4Kids: utilizando Tecnologia Livre e materiais de baixo custo para o exercício do Pensamento Computacional no Ensino Fundamental I por meio do aprendizado de programação aliado à Robótica Educacional.

Dissertação de Mestrado submetida ao Programa de Pós-graduação em Informática do Instituto de Matemática e do Instituto Tércio Pacitti da Universidade Federal do Rio de Janeiro - UFRJ, como parte dos requisitos necessários à obtenção do título de Mestre em Informática.

Orientador:

Fábio Ferrentini Sampaio, Ph.D.

Coorientadora:

Mônica Pereira dos Santos, Ph.D.

Área de Pesquisa:

Informática, Educação e Sociedade

Rio de Janeiro

2017

CIP - Catalogação na Publicação

Q3d Queiroz, Rubens Lacerda
DuinoBlocks4Kids: utilizando tecnologia livre e materiais de baixo custo para o exercício do pensamento computacional no ensino fundamental 1 por meio do aprendizado de programação aliado à robótica educacional / Rubens Lacerda Queiroz. -- Rio de Janeiro, 2017.
186 f.

Orientador: Fábio Ferrentini Sampaio.
Coorientadora: Mônica Pereira dos Santos.
Dissertação (mestrado) - Universidade Federal do Rio de Janeiro, Instituto Tércio Pacitti de Aplicações e Pesquisas Computacionais, Programa de Pós-Graduação em informática, 2017.

1. Pensamento Computacional. 2. Ensino de Programação. 3. Robótica Educacional. 4. Linguagem de Programação Visual. 5. Maturidade Cognitiva e Aprendizado de Programação. I. Sampaio, Fábio Ferrentini, orient. II. dos Santos, Mônica Pereira, coorient. III. Título.

DuinoBlocks4Kids: utilizando Tecnologia Livre e materiais de baixo custo para o exercício do Pensamento Computacional no Ensino Fundamental I por meio do aprendizado de programação aliado à Robótica Educacional

Rubens Lacerda Queiroz

Dissertação de Mestrado submetida ao Programa de Pós-graduação em Informática do Instituto de Matemática e do Instituto Tércio Pacitti da Universidade Federal do Rio de Janeiro - UFRJ, como parte dos requisitos necessários à obtenção do título de Mestre em Informática.

Aprovada em: Rio de Janeiro, 30 de março de 2017, por:



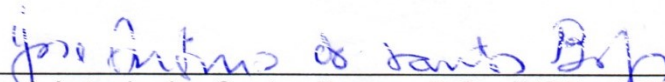
Orientador: Fábio Ferrentini Sampaio, Ph.D. (NCE e PPGI/UFRJ)



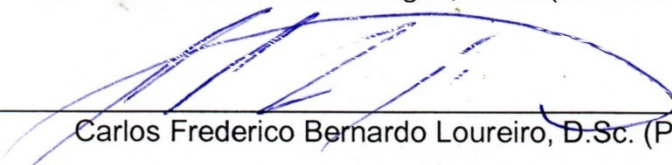
Coorientadora: Mônica Pereira dos Santos, Ph.D. (PPGE/UFRJ)



Priscila Machado Vieira Lima Ph.D. (NCE e PPGI/UFRJ)



Jose Antonio dos Santos Borges, D.Sc. (NCE e HCTE/UFRJ)



Carlos Frederico Bernardo Loureiro, D.Sc. (PPGE/UFRJ)

Aos meus pais

AGRADECIMENTOS

Agradeço aos meus orientadores, Prof. Fábio Ferrentini Sampaio e Prof^a. Mônica Pereira dos Santos, pelo companheirismo, pela confiança em mim depositada e pela parceria constante na realização desse trabalho. Aos professores do PPGI, pela preocupação incessante com o progresso dos alunos e pela busca em respeitar e aproveitar os seus potenciais individuais. Aos colaboradores do LlvRE, por todo o suporte dado ao projeto. Aos colegas do PPGI, pelo companheirismo e pelos conhecimentos compartilhados, em especial à Adriele Ribeiro, meu braço direito nas aulas da oficina de robótica. À equipe do Dosvox, pela agradável companhia e parceria e por terem tão gentilmente me acolhido em seu espaço, em especial ao Prof. Antonio Borges, por todo incentivo e apoio a mim dedicados nesse meu retorno à vida acadêmica. Aos meus colegas do PGTIAE, pela calorosa amizade e pelo incentivo dado ao meu ingresso no mestrado. Ao criador do Ardublockly (ambiente sobre o qual foi desenvolvido o DuinoBlocks4Kids), Carlos Pereira, pelo pronto e gentil atendimento à todas as minhas solicitações de suporte. Ao Prof. Eduardo Bento, da UFSJ, que, acreditando nesse projeto, deu início a uma parceria que certamente ajudará a difundir e ampliar as possibilidades de uso do DB4K. E, por fim, a minha família, que desde sempre deu todo suporte necessário aos meus projetos e sonhos, tendo sido meu principal alicerce, não somente nessa como em todas as minhas empreitadas.

“O importante é não parar de questionar. A curiosidade tem sua própria razão para existir. Não se pode deixar de ficar admirado ao se contemplar os mistérios da eternidade, da vida, da maravilhosa estrutura da realidade. Basta que se tente simplesmente compreender um pouco deste mistério a cada dia. Nunca perca a sagrada curiosidade”.

(Albert Einstein)

RESUMO

Queiroz, Rubens Lacerda. DuinoBlokcks4Kids: utilizando Tecnologia Livre e materiais de baixo custo para o exercício do Pensamento Computacional no Ensino Fundamental I por meio do aprendizado de programação aliado à Robótica Educacional. 2017. Dissertação (Mestrado em Informática) – Instituto de Matemática, Instituto Tércio Pacitti de Aplicações e Pesquisas Computacionais, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2017.

Este trabalho apresenta uma investigação acerca do desenvolvimento e exercício de habilidades do Pensamento Computacional em crianças do Ensino Fundamental I, a partir do aprendizado de conceitos básicos de programação, por meio de recursos baseados em Tecnologia Livre e materiais de baixo custo associados a estratégias pedagógicas alicerçadas na Robótica Educacional e desenvolvidas especificamente para esse público. Com esse propósito, buscou-se, por meio de um estudo de caso realizado com sete crianças do terceiro e quarto anos do Ensino Fundamental, levantar hipóteses acerca da possível relação existente entre a maturidade cognitiva de crianças nessa faixa etária e o aprendizado de estruturas básicas de programação, bem como verificar de que forma o aprendizado dessas estruturas, por meio das estratégias pedagógicas propostas por este trabalho, pode apoiar o exercício de determinadas habilidades do Pensamento Computacional. Foi desenvolvido, para a realização do estudo, um kit didático denominado DuinoBlocks4kids (DB4K), composto por um Ambiente de Programação Visual em Blocos para plataforma de prototipagem eletrônica Arduino, uma proposta metodológica, uma série de atividades e um conjunto de materiais de robótica. Pôde-se observar, como resultado do uso deste kit, a possibilidade do exercício das seguintes habilidades do Pensamento Computacional: capacidade de abstração, compreensão de fluxos de controle, depuração e detecção sistemática de erros, uso da lógica condicional e decomposição de problemas. Foram encontrados também indícios da existência de uma relação direta entre algumas características cognitivas de crianças no período operatório concreto (tais como a habilidade de sequenciar eventos ou ideias, a habilidade de realizar operações mentais a partir de experiências concretas, dentre outras) e as habilidades necessárias para a realização de determinadas tarefas relacionadas à programação de computadores.

Palavras-chave: Pensamento Computacional. Ensino de Programação. Robótica Educacional. Linguagem de Programação Visual. Maturidade Cognitiva e Aprendizado de Programação.

ABSTRACT

Queiroz, Rubens Lacerda. DuinoBlokcks4Kids: utilizando Tecnologia Livre e materiais de baixo custo para o exercício do Pensamento Computacional no Ensino Fundamental I por meio do aprendizado de programação aliado à Robótica Educacional. 2017. Dissertação (Mestrado em Informática) – Instituto de Matemática, Instituto Tércio Pacitti de Aplicações e Pesquisas Computacionais, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2017.

This work presents an investigation about the development and practice of some Computational Thinking abilities in primary school children from the learning of some basic concepts of computer programming, through resources based on Free Technology and low cost materials, combined with pedagogical strategies based on Educational Robotics specifically developed for this audience. With this purpose, by means of a case study carried out with seven children aged between 8 and 10, it is seeking to raise hypotheses about a possible relationship between cognitive maturity of children in this age group and the learning of basic structures of computer programming, as well as to verify how the learning of this structures, through the pedagogical strategies proposed by this work, can support the exercise of certain Computational Thinking skills. To carry out this study, a teaching kit called DuinoBlocks4Kids (DB4K), composed of a Visual Programming Language for Arduino, a methodological proposal, a series of activities and a set of robotic materials have been developed. As a result of the use of this kit it was possible to observe the possibility of exercising the following Computational Thinking skills: abstraction ability, understanding of flow of control, debugging and systematic error detection, the use of conditional logic and problem decomposition. It was also found indications of the existence of a direct relationship between some cognitive characteristics of children in the concrete operational stage (such as the ability to sequence events or ideas, the ability to carry out mental operations based on concrete experiences, among others) and the skills needed to perform certain tasks related to computer programming.

Keywords: Computational Thinking. Teaching computer programming. Educational Robotics. Visual Programming Language. Cognitive Maturity and Computer Programming Learning.

LISTA DE ILUSTRAÇÕES

Figura 1: Imagem vista pelo usuário por meio do vídeo apresentado pelo LabVad	
Robótica	50
Figura 2: Placa Arduino Uno e dispositivos controlados pelo experimento LabVad	
Robótica	51
Figura 3: Carrinho e braço robóticos programáveis via DB4K client-side	52
Figura 4: Caixinha mágica.....	53
Figura 5: O morcego robô e o peixe robô de garrafa pet.....	54
Figura 6: Cachorro robô de garrafa pet	54
Figura 7: Kit Arduino	55
Figura 8: Visão Geral da Interface do DB4k client-side	58
Figura 9: Diagrama de Classes simplificado da interface do DB4K	59
Figura 10: Blocos da categoria controle	60
Figura 11: Blocos da categoria LED	60
Figura 12: Blocos da categoria Motores	61
Figura 13: Blocos da categoria Display	61
Figura 14: Blocos da categoria Som	61
Figura 15: Blocos da categoria Sensores	62
Figura 16: Blocos da categoria Carrinho	62
Figura 17: Blocos da categoria Braço	62
Figura 18: Área de trabalho e blocos da categoria LED	63
Figura 19: Área para a exibição do programa textual (Wiring) correspondente ao programa criado com os blocos	64
Figura 20: Barra de navegação	65
Figura 21: Menu retrátil	65
Figura 22: Janela de configurações	66
Figura 23: Janela retrátil para exibição de mensagens do compilador Arduino	66
Figura 24: Diagrama de casos de uso com as principais funcionalidades do DB4K	67
Figura 25: Diagrama da principal sequência de atividades realizada pelas crianças quando da programação dos materiais de robótica com o DB4K	68
Figura 26: Exemplo de decomposição de problemas	79
Figura 27: Diagrama de Atividades da tarefa de depuração de erros	80

Figura 28: Exemplos de programas de um único comando.....	83
Figura 29: Exemplos de programas utilizando apenas a estrutura sequencial	85
Figura 30: Interpretação equivocada do enunciado do programa proposto	87
Figura 31: Exemplo de programa a ser executado pelo robô humano	89
Figura 32: Gabarito para construção do programa da palavra-código	89
Figura 33: Programa construído a partir de uma interpretação equivocada do enunciado.....	91
Figura 34: Exemplo de problema de entendimento do uso do bloco repetir	92
Figura 35: Não entendimento da sequência de comandos necessárias para fazer um LED piscar.....	93
Figura 36: Exemplos de programas a serem executados pelo robô humano	95
Figura 37: Dificuldade no entendimento dos comandos necessários para fazer um LED piscar	96
Figura 38: Programa incorreto corrigido com base na observação do resultado obtido a partir da sua execução	97
Figura 39: Sequência de blocos para fazer o LED vermelho piscar 2 vezes sem o uso do bloco repetir.	98
Figura 40: Sequência de blocos para fazer o LED vermelho piscar 2 vezes com o uso do bloco repetir	98
Figura 41: Uso do bloco repetir	100
Figura 42: Programa para escrever a sequência de números 5,3,8 no display de 7 segmentos.....	105
Figura 43: Programa resultante da observação do funcionamento da caixinha mágica	106
Figura 44: Programa para o robô humano utilizando o bloco enquanto	108
Figura 45: Programa exemplo utilizando o bloco enquanto	109
Figura 46: Programa com LCD faltando bloco esperar.....	110
Figura 47: Programa montado por Hilton	112
Figura 48: Programa de Daniel	113
Figura 49: Primeira e segunda tentativas de Bruno para fazer o LED piscar.....	113
Figura 50: Última tentativa de Bruno para fazer o LED piscar	114
Figura 51: Programa de Ivana.....	115
Figura 52: Programa a ser construído na segunda atividade	116

Figura 53: Programa do peixe robô	118
Figura 54: Programa de controle do morcego robô	121
Figura 55: Programa de Ivana para o morcego robô	122
Figura 56: Programa utilizado na revisão sobre a estrutura de repetição condicional	124
Figura 57: Programa de José para o morcego robô	125
Figura 58: Programa de Daniel para o morcego robô.....	125
Figura 59: Código base utilizado por Marcelo para programar o morcego robô	126
Figura 60: Programa básico do carrinho robô	128
Figura 61: Programa do carrinho modificado por José	128
Figura 62: Programa inicial de Daniel para o carrinho robô	129
Figura 63: Tabuleiro e peões para atividade com o braço robótico	131
Figura 64: Programa de controle do cachorro robô	131
Figura 65: Erro cometido por José na parte do programa responsável por mover o rabinho do cachorro robô.....	132
Figura 66: Código de José para a segunda etapa de construção do programa do cachorro robô	132
Figura 67: Código de Davi para segunda etapa do programa do cachorro robô	133
Figura 68: Uso equivocado do motor DC no programa do cachorro robô	133
Figura 69: Terceira e última parte do código para o programa do cachorro robô ...	135
Figura 70: Programa para o braço robótico	135
Figura 71: Primeira tentativa de Davi para o programa do braço robótico	136
Figura 72: Programa a ser descrito textualmente pelas crianças	140
Figura 73: Exercício de reconhecimento dos dispositivos	142
Figura 74: Programa com erros	143
Figura 75: Programa sem erros	144
Figura 76: Robô lutador desenhado por José	145
Figura 77: Robô desenhado por Daniel.....	145
Figura 78: Programa criado por Daniel para o seu robô	146
Figura 79: Boné robô desenhado por Hilton	147
Figura 80: Robô pipoqueiro-sorveteiro desenhado por Ivana	147
Figura 81: Robô jogador de tênis desenhado por Marcelo	148
Figura 82: Robô desenhado por Bruno	148

Figura 83: Smileyometer	158
Figura 84: Blocos ascender e apagar LED	159
Figura 85: Caracteres unicode utilizados para distinção dos parâmetros de intensidade	160
Figura 86: Botões flutuantes para abrir o código no IDE e compilar o código Wiring	160
Figura 87: Item de menu Novo Programa	161
Figura 88: Botão Finalizar Execução do Programa	162
Figura 89: Questionário aplicado após a primeira aula na qual foi utilizado o ambiente de programação (terceira aula da oficina)	162
Figura 90: Questionário aplicado após a nona aula da oficina	163
Figura 91: Exemplo de questionário de avaliação dos recursos pedagógicos	164
Figura 92: Generalização do uso dos dispositivos responsáveis pela locomoção do carrinho robótico	168
 Quadro 1: Mapa sintético das atividades e conteúdos trabalhados na oficina	75
Quadro 2: Escala Likert utilizada para a avaliação do aprendizado.....	76
Quadro 3: Exemplo de itens de referência para avaliação de habilidades específicas	76
Quadro 4: itens de referência para avaliação da habilidade de depuração e correção dos programas desenvolvidos	77

LISTA DE SIGLAS

DB4K	DuinoBlocks4Kids
FSF	<i>Free Software Foundation</i>
GINAPE	Grupo de Informática Aplicada à Educação da UFRJ
LabVad	Laboratório Virtual de Atividades Didáticas em Ciências e Robótica
LivRE	Laboratório de Inovações em Robótica Educacional
MIT	<i>Massachusetts Institute of Technology</i>
PPGI	Programa de Pós-Graduação em Informática
STEM	<i>Science, Technology, Engineering and Mathematics</i>
TIC	Tecnologias de Informação e Comunicação

SUMÁRIO

1 INTRODUÇÃO.....	15
1.1 MOTIVAÇÃO	15
1.2 OBJETIVOS.....	17
1.2.1 Objetivo Geral.....	17
1.2.2 Objetivos Específicos	18
1.3 RELEVÂNCIA	18
1.4 PERGUNTA DE PESQUISA.....	20
1.5 METODOLOGIA	20
1.5.1 Organização do estudo.....	24
2 FUNDAMENTAÇÃO TEÓRICA.....	26
2.1 ROBÓTICA EDUCACIONAL.....	26
2.2 ENSINO DE PROGRAMAÇÃO PARA CRIANÇAS	29
2.3 PENSAMENTO COMPUTACIONAL	31
2.4 O PERÍODO OPERATÓRIO CONCRETO E O APRENDIZADO DE PROGRAMAÇÃO	33
2.5 LINGUAGENS DE PROGRAMAÇÃO VISUAL PARA CRIANÇAS	39
2.6 SOFTWARE E HARDWARE LIVRES.....	41
2.7 ARDUINO	43
2.8 COMPUTAÇÃO DESPLUGADA	44
2.9 USO DE NARRATIVAS NO PROCESSO ENSINO-APRENDIZAGEM	46
2.10 TRABALHOS CORRELATOS	48
3 O DUINOBLOCKS4KIDS (DB4K)	50
3.1 ESCOLHA DOS MATERIAIS DE ROBÓTICA A SEREM MANIPULADOS PELO AMBIENTE	52
3.2 A ESCOLHA DAS FERRAMENTAS PARA CONSTRUÇÃO DO AMBIENTE ..	55
3.3 O AMBIENTE	58
4 A OFICINA.....	69
4.1 OBJETIVOS.....	69
4.2 PROPOSTA METODOLÓGICA	69
4.3 CONTEÚDOS	71
4.4 DURAÇÃO	72
4.5 PLANOS DE AULA.....	72
4.6 PARTICIPANTES.....	72
4.7 COLETA DE DADOS E AVALIAÇÃO.....	74
5 AVALIAÇÃO DO APRENDIZADO DAS ESTRUTURAS BÁSICAS DE PROGRAMAÇÃO E SUA RELAÇÃO COM ASPECTOS DO PENSAMENTO COMPUTACIONAL	76
5.1 PROGRAMAÇÃO SEQUENCIAL.....	82
5.1.1 Primeiro Encontro (Aula 1).....	82
5.1.2 Segundo Encontro (Aula 2).....	83
5.1.3 Terceiro Encontro (Aula 3).....	84

5.2 ESTRUTURA DE REPETIÇÃO CONTADA	88
5.2.1 Primeiro Encontro (Aula 4)	88
5.2.2 Segundo Encontro (Aula 5)	94
5.2.3 Terceiro Encontro (Aula 6)	99
5.2.4 Quarto Encontro (Aula 7)	101
5.2.5 Quinto Encontro (Aula 8)	103
5.3 ESTRUTURA DE REPETIÇÃO CONDICIONAL	107
5.3.1 Primeiro Encontro (Aula 9)	107
5.3.2 Segundo Encontro (Aula 10)	111
5.3.3 Terceiro Encontro (Aula 11)	119
5.3.4 Quarto Encontro (Aula 12)	123
5.3.5 Quinto Encontro (Aula 13)	129
5.4 AULA DE ENCERRAMENTO (AULA 14)	138
5.4.1 Atividades Desenvolvidas	138
5.4.2 Resultados	139
5.5 SÍNTESE DA AVALIAÇÃO DO APRENDIZADO	149
5.5.1 Uma visão geral do aproveitamento dos alunos	149
5.5.2 Construção de programas puramente sequenciais	149
5.5.3 Uso da Estrutura de Repetição Contada	151
5.5.4 Uso da Estrutura de Repetição Condicional	153
5.5.5 Atividades de fechamento	156
5.5.6 Sobre as dificuldades apresentadas	156
6 AVALIAÇÃO DA USABILIDADE DO AMBIENTE DE PROGRAMAÇÃO EM BLOCOS	157
6.1 RESULTADOS DAS OBSERVAÇÕES	158
6.1.1 Modificações realizadas nos elementos de interface e funcionalidades já existentes	159
6.1.2 Inclusão de novas funcionalidades	160
6.2 RESULTADOS DOS QUESTIONÁRIOS	162
7 AVALIAÇÃO DOS RECURSOS PEDAGÓGICOS	164
7.1 AS BRINCADEIRAS	164
7.2 AS NARRATIVAS	165
7.3 OS MATERIAIS DE ROBÓTICA	166
7.3.1 A caixinha mágica	166
7.3.2 Os robôs de garrafa pet	167
7.3.3 O carro robô e o braço robótico	168
8 CONSIDERAÇÕES FINAIS	170
8.2 PRINCIPAIS CONTRIBUIÇÕES	173
8.3 PUBLICAÇÕES	174
8.4 TRABALHOS FUTUROS	175
REFERÊNCIAS	176
APÊNDICE A – PLANOS DE AULA	186
APÊNDICE B – ROTEIRO DE AVALIAÇÃO DE USABILIDADE E FUN	236

1 INTRODUÇÃO

1.1 MOTIVAÇÃO

No Brasil, já há alguns anos, verifica-se um déficit na formação de profissionais [na área de TI]. [...]. Soma-se a este quadro a diminuição na procura por cursos superiores em Computação e TI e o alto índice de evasão nesses cursos [...] (BARCELOS e SILVEIRA, 2012).

Um estudo divulgado em março de 2013 pela Associação Brasileira de Empresas de Tecnologia da Informação e Comunicação (Brasscom)¹ mostra que, na época, a taxa de evasão escolar na área de TI era de 87% e que a concorrência por cursos neste ramo encontrava-se baixa. Diante deste cenário, a previsão era de que, para o ano de 2014, embora o mercado do Distrito Federal e de sete estados (São Paulo, Rio de Janeiro, Paraná, Rio Grande do Sul, Minas Gerais, Bahia e Pernambuco) fossem ter uma demanda por 78 mil profissionais do setor, somente 33 mil estudantes iriam concluir seus cursos (BRASSCON, 2013). Em pesquisa mais recente, divulgada pelo jornal Estadão em março de 2016, dados da Associação para a Promoção da Excelência do Software Brasileiro (Softex)² indicam que o Brasil pode chegar a um déficit de 408 mil profissionais de TI até 2020. No cenário mundial, de acordo com a organização Code.org³, 1,4 milhão de vagas para o setor devem ser abertas em todo o mundo até o referido ano, mas, apenas 400 mil delas serão preenchidas (ESTADÃO, 2016).

Diferentes estudos acadêmicos apontam, como uma das possíveis causas desse quadro, um frequente insucesso nas disciplinas de programação de computadores em capacitar os alunos a implementarem a solução de um determinado problema através do uso de alguma linguagem de programação, mesmo nos casos em que o problema dado apresenta um baixo grau de complexidade (GOMES, HENRIQUES e MENDES, 2008). Muito dessa dificuldade existe em decorrência da não introdução da Ciência da Computação como parte das ciências básicas, o que faz com que ela normalmente não seja abordada nas séries fundamentais (BARCELOS e SILVEIRA, 2012).

Outro aspecto que tem recebido crescente atenção em relação a esse tema, e que acaba por apontar a programação de computadores como sendo uma nova

¹ <http://www.brasscom.org.br/>

² <http://www.softex.br/>

³ <https://code.org/>

habilidade básica, tal qual a escrita, a leitura e a aritmética, é a importância do desenvolvimento dessa habilidade como meio de possibilitar a todo cidadão ser produtor e não apenas consumidor de tecnologia digital, oportunizando assim, entre outras coisas, a mobilidade social em um mundo fortemente orientado por esse tipo de tecnologia (WHITEHOUSE, 2016).

Diante deste cenário, diferentes autores, como Mitchel Resnick e outros (2009), vêm defendendo a necessidade de se incorporar o ensino de programação já nos primeiros anos do Ensino Básico (K-12 nos EUA), como forma de desenvolver nos alunos competências e habilidades relacionadas ao Pensamento Computacional: uma forma de pensamento característica dos cientistas da computação, mas, universalmente aplicável, que envolve um conjunto de atitudes e habilidades, tais como o uso da recursividade, generalização e decomposição na solução de problemas tanto técnico-científicos quanto da vida cotidiana (WING, 2006).

Para que iniciativas nesse sentido sejam adotadas, torna-se necessário o desenvolvimento de recursos idealizados especificamente para o ensino de programação de computadores a crianças. Como resposta a essa demanda, uma série de estratégias vêm sendo pensadas e implementadas, entre elas, a associação da Robótica Educacional com Ambientes de Programação Visual em Blocos (entre outros tipos de Linguagens de Programação Visual⁴).

O GINAPE, Grupo de Informática Aplicada à Educação do NCE/UFRJ, através do seu Laboratório de Inovações em Robótica Educacional (LIVRE⁵) vem, há algum tempo, desenvolvendo trabalhos ligados à Robótica Educacional e ao ensino de programação. Estes trabalhos estão ligados a um projeto “central”, chamado LabVad - Laboratório Virtual de Atividades Didáticas em Ciências e Robótica⁶ - uma plataforma que disponibiliza diferentes experimentos controláveis remotamente (SOUZA *et al*, 2014).

Como exemplos dos trabalhos relacionados à Robótica Educacional desenvolvidos pelo LIVRE, pode-se citar as dissertações de mestrado de Pinto (2011) que nos apresenta um estudo sobre a aplicação de uma arquitetura pedagógica em um curso de Robótica Educacional, a de Alves (2013), na qual é feita a especificação

⁴ Linguagens de Programação Visual, dentre as quais se incluem os Ambientes de Programação em Blocos, são aquelas nas quais a "sintaxe (semanticamente significativa) inclui expressões visuais" (BURNETT, 1999, p.1) permitindo que os usuários manipulem graficamente os elementos do programa ao invés de especificá-los textualmente (ver capítulo 2.5).

⁵ www.nce.ufrj.br/ginape/livre/

⁶ <http://labvad.nce.ufrj.br/>

e implementação de um Ambiente de Programação em Blocos desenvolvido especificamente para a programação de placas de prototipagem eletrônica Arduino, e a de Dargains (2015) onde é feito um estudo exploratório sobre o uso da Robótica Educacional no ensino de programação introdutória.

Os trabalhos acima mencionados, aos quais somam-se outros sobre o mesmo tema desenvolvidos pelo LlvRE/LabVad, fornecem uma base científica sólida sobre a qual é possível se desenvolver uma gama de novos trabalhos, entre eles, aqueles que possibilitem a extensão dos benefícios alcançados pelas pesquisas anteriores, voltadas mais especificamente ao público jovem e adulto, a alunos do Ensino Fundamental I.

Uma revisão recente da literatura na área de ambientes de programação visual voltados ao ensino de programação com robótica, realizada pela equipe do LlvRE, apontou para a inexistência de linguagens visuais voltadas à programação de placas de prototipagem eletrônica Arduino (tecnologia adotada pelo projeto LabVad), pensadas e desenvolvidas especificamente para o ensino de programação a crianças. Como resposta a essa demanda, surge a proposta do desenvolvimento do DuinoBlocks4Kids (DB4K), um kit didático baseado em Tecnologia Livre, voltado ao ensino de programação para crianças do Ensino Fundamental I, composto por um Ambiente de Programação Visual baseado em blocos, um conjunto de materiais de Robótica Educacional e uma série de atividades.

1.2 OBJETIVOS

1.2.1 Objetivo Geral

O objetivo geral deste estudo é investigar a possibilidade de desenvolvimento e exercício de habilidades do Pensamento Computacional em crianças do Ensino Fundamental I (pertencentes a comunidades de baixa renda) a partir do aprendizado de conceitos básicos de programação por meio de recursos baseados em Tecnologia Livre e materiais de baixo custo, associados a estratégias pedagógicas alicerçadas na Robótica Educacional, pensadas e desenvolvidas especificamente para esse público.

1.2.2 Objetivos Específicos

- Desenvolvimento de um Ambiente de Programação Visual em Blocos para programação de placas de prototipagem eletrônica Arduino desenhado especificamente para o público infantil.
- Integração do Ambiente de Programação em Blocos desenvolvido à plataforma LabVad.
- Elaboração de materiais de Robótica Educacional a partir do uso de sucata e materiais descartáveis ou de baixo custo (como garrafas pet e potes plásticos) a serem utilizados em conjunto com o Ambiente de Programação em Blocos.
- Desenho de um conjunto de atividades didáticas e estratégias pedagógicas que deem suporte ao desenvolvimento e exercício de habilidades do Pensamento Computacional por meio do aprendizado de conceitos básicos de programação a partir do uso do ambiente de programação e dos materiais de robótica desenvolvidos.
- Realização de uma oficina para aplicação do kit didático produzido.
- Avaliação dos resultados obtidos na oficina.

1.3 RELEVÂNCIA

Na sociedade informatizada em que vivemos, as crianças já nascem imersas no mundo digital, sendo conhecidas, devido à sua aparente fluência com as novas tecnologias, como “nativos digitais” (PRENSKY, 2001). Entretanto, Mitchel Resnick (2014) argumenta que, para se adquirir fluência em relação às novas tecnologias, não é suficiente saber usar com naturalidade as Tecnologias de Informação e Comunicação (TIC), é necessário que se saiba “projetar, criar e se expressar através dessas tecnologias” e, para tanto, é imprescindível saber programar. Além disso, a aprendizagem de programação é uma importante ferramenta de apoio ao desenvolvimento do Pensamento Computacional (GROVER e PEA, 2013) que, por meio do reforço e aprimoramento de uma série de habilidades intelectuais, estende o alcance dos seus benefícios a uma ampla gama de domínios (WING, 2011).

No Brasil, o Ministério da Educação (MEC), por considerar o letramento digital um direito a ser assegurado, de grande relevância para a cidadania e para a atuação crítica na vida social, definiu, no texto mais recente da Base Nacional Comum Curricular (BNCC), um tema integrador denominado “Culturas Digitais e Computação”. Este tema está relacionado ao “uso pedagógico das novas tecnologias da comunicação e [à] exploração dessas novas tecnologias para a compreensão do mundo e para a atuação nele”. (MEC, 2016, p.51). De acordo com o MEC (2016), em uma sociedade cada vez mais tecnologicamente organizada, torna-se imperativo que a escola passe a considerar as potencialidades do uso dos recursos tecnológicos para o alcance de suas metas.

Também preocupada com a temática da introdução das Ciências da Computação no Ensino Básico, a Sociedade Brasileira de Computação (SBC) realizou, no ano de 2016, duas reuniões com o MEC para discutir a definição de diretrizes curriculares nacionais específicas para a área da computação, apresentando, durante esses encontros, diversas justificativas sobre a importância da inclusão de conteúdos de computação na Educação Básica, tanto no Ensino Fundamental quanto no Ensino Médio (SBC, 2016a; SBC, 2016b).

Esta preocupação, tanto do Governo Federal quanto da SBC, em relação à introdução da computação na Educação Básica e de sua importância como ferramenta de atuação do aluno sobre o mundo, cria um ambiente favorável à utilização do aprendizado de programação como ferramenta de potencialização daquilo que o aluno pode criar e aprender com o computador.

No tocante à Robótica Educacional, estudos acerca dos kits utilizados no Brasil para o ensino de programação a crianças, como os realizados por Costa Jr. e Guedes (2015) e por de França e do Amaral (2013), apontam o uso mais frequente de quatro kits: Lego Mindstorms⁷, Modelix⁸, Robomind⁹ e Arduino¹⁰. Os três primeiros são proprietários e possuem material pedagógico próprio. O Arduino é baseado em Tecnologia Livre (ver capítulo 2.6), mas, não foram encontrados materiais

⁷ <https://www.lego.com/en-us/mindstorms>

⁸ <http://modelix.cc/>

⁹ <http://www.robomind.com.br/>

¹⁰ <https://www.arduino.cc/>

pedagógicos para esse kit no levantamento feito por Costa Jr. e Guedes (2015). Essa inexistência (ou pouca oferta) de material didático apropriado para o uso de Tecnologia Livre no aprendizado de programação com robótica no Nível Fundamental, aliada ao custo relativamente elevado das soluções proprietárias citadas, pode vir a contribuir com um agravamento da exclusão digital a qual comumente estão expostas as crianças residentes em comunidades de baixa renda e matriculadas na rede pública de ensino (notoriamente, carente de recursos).

Tais aspectos tornam bastante nítida a relevância da inclusão do ensino de programação de computadores desde as primeiras séries do Ensino Fundamental, bem como o desenvolvimento de ferramentas economicamente acessíveis que deem o suporte adequado a esse aprendizado de modo que ele possa ocorrer de forma efetiva e prazerosa. O DuinoBlocks4Kids vem exatamente ao encontro dessa demanda.

1.4 PERGUNTA DE PESQUISA

O Ambiente de Programação em Blocos DB4K, em conjunto com os materiais de robótica e as atividades didáticas propostas por esta pesquisa, propicia o desenvolvimento e exercício de habilidades do Pensamento Computacional em crianças do Ensino Fundamental I, residentes em comunidades de baixa renda, a partir do aprendizado de conceitos básicos de programação?

1.5 METODOLOGIA

No que concerne à metodologia de pesquisa adotada, este trabalho pode ser caracterizado como um estudo de caso único, exploratório, de caráter qualitativo. Conduzir uma pesquisa por meio de um estudo de caso torna-se interessante quando o seu “foco se encontra em fenômenos contemporâneos inseridos em algum contexto da vida real” (YIN, 2001, p.19), sendo exatamente o caso desse trabalho.

Um estudo de caso pode ser entendido como uma instância específica destinada a ilustrar um princípio mais genérico. (NISBET e WATT, 1984 *apud* COHEN, MANION e MORRISON, 2007). Essa instância é um sistema limitado, como por

exemplo: um sujeito, uma sala de aula, uma comunidade, etc., e provê um exemplo único de pessoas reais em situações reais (COHEN, MANION e MORRISON, 2007). No caso desta pesquisa, a instância estudada foi um grupo de crianças do terceiro e quarto anos do Ensino Fundamental I participantes de uma oficina de aprendizado de programação de computadores por meio da Robótica Educacional.

“Todo o bom desenho de estudo de caso incorpora uma teoria, que serve como plano geral da investigação, da busca de dados e da sua interpretação” (YACUZZI, 2005 *apud* MEIRINHOS e OSÓRIO, 2010, p.55). Desenvolver uma teoria antes que se faça coleta de dados é essencial para que se possa delinear de maneira mais clara quais dados devem ser coletados e que estratégias utilizar em relação à análise dos mesmos (YIN, 2001). Nesse sentido, para nortear a coleta e análise de dados, lançou-se mão, neste trabalho, da teoria de desenvolvimento cognitivo de Jean Piaget, do Construcionismo de Seymour Papert e do conceito de Pensamento Computacional delineado por Jeannette Wing (ver capítulo 2).

A utilização da teoria, ao realizar estudos de caso, não apenas representa uma ajuda imensa na definição do projeto de pesquisa e na coleta de dados adequados, como também torna-se o veículo principal para a generalização dos resultados do estudo de caso (YIN, 2001, p.54).

Os estudos de caso possuem algumas características bastante específicas que cabem aqui ser destacadas, e que fazem desta uma excelente ferramenta de observação e interpretação de contextos educacionais (MERRIAM, 1988 *apud* SOUZA, 2007):

- Se preocupam em oferecer uma descrição rica e vívida de eventos relevantes para o caso.
 - Fornecem uma narrativa cronológica desses eventos.
 - Combinam a descrição dos eventos com sua análise.
 - Focam em um ator ou em um grupo de atores e procuram entender a percepção destes em relação aos eventos.
 - Destacam eventos específicos que são relevantes para o caso.
 - O pesquisador se envolve integralmente com o caso.
 - É feita uma tentativa de retratar a riqueza do caso quando da escrita do relato.
- (HITCHCOCK e HUGHES, 1995, p.317 *apud* COHEN, MANION e MORRISON, 2007, p.253, tradução nossa).

Diferentemente do que acontece nos experimentos, onde manipula-se variáveis e se determina as suas relações causais, ou em *surveys*, nas quais são utilizados questionários padronizados aplicados à grandes amostras de indivíduos, em um estudo de caso, o pesquisador tipicamente usa a observação como fonte de

dados (COHEN, MANION e MORRISON, 2007), sendo que, “estudos de observação são superiores a experimentos e a *surveys* quando os dados estão sendo coletados em procedimentos não verbais” (BAILEY, 1994, p.243 *apud* COHEN, MANION e MORRISON, 2007, p.260, tradução nossa), como no caso das atividades de programação e manipulação de dispositivos de robótica utilizadas como fonte de dados para esta pesquisa.

Quando um estudo de caso é utilizado como piloto para outros estudos, ou novas perguntas de pesquisa, ele é denominado estudo de caso exploratório (YIN, 2001 *apud* COHEN, MANION e MORRISON, 2007). Busca-se, por meio da pesquisa exploratória, “descobrir ideias e intuições, na tentativa de adquirir maior familiaridade com o fenômeno pesquisado” (SELLTIZ, WRIGHTSMAN e COOK, 1965 *apud* OLIVEIRA, 2011, p.20), “gerar mais informações que possam ser adquiridas para a realização de futuras pesquisas conclusivas” (ZIKMUND, 2000 *apud* OLIVEIRA, 2011, p.21). Isto é, estudos exploratórios “são o prelúdio para uma investigação subsequente, [...] podendo buscar hipóteses e proposições relevantes para orientar estudos posteriores ” (MEIRINHOS e OSÓRIO, 2010, p.57), sendo exatamente esta uma das propostas desse trabalho.

Neste tipo de pesquisa, trabalha-se normalmente com amostras pequenas e a análise dos dados é feita de forma qualitativa (MALHOTRA, 2001 *apud* OLIVEIRA, 2011).

Na perspectiva qualitativa, o ambiente natural é a fonte direta de dados e o pesquisador, o principal instrumento, sendo que os dados coletados são predominantemente descritivos. Além disso [...] a preocupação com o processo é muito maior do que com o produto, ou seja, o interesse do pesquisador ao estudar um determinado problema é verificar "como" ele se manifesta nas atividades, nos procedimentos e nas interações cotidianas. Outro aspecto é que a análise dos dados tende a seguir um processo indutivo – a pesquisa qualitativa é emergente em vez de estritamente pré-configurada. (CRESWELL, 2007 *apud* AUGUSTO *et al*, 2013, p.748)

Cohen, Manion e Morrison (2007), nos apontam algumas maneiras de organizar e apresentar a análise de dados qualitativos, podendo essa organização ser feita por “questões” ou por indivíduos. Optou-se neste trabalho por organizar a análise por questões, considerando-se as estruturas básicas de programação trabalhadas durante a oficina como as questões guias. Dentro dessas questões foi feita então uma avaliação por indivíduo, apresentando-se a resposta de cada um dos sujeitos da pesquisa aos desafios propostos em relação à estrutura de programação trabalhada. Por fim, a partir dessa análise individualizada, foi realizado (de acordo com o que

sugerem os mesmos autores) um segundo nível de análise, tomando como base o cruzamento das respostas individuais, “em busca de temas, padrões de resposta, [...], concordâncias e discordâncias, para comparar indivíduos e as questões por eles levantadas, ou seja, para sintetizar os dados” (COHEN, MANION e MORRISON, 2007, p.467, tradução nossa).

Como já mencionado, a perspectiva qualitativa tem na figura do investigador o seu principal instrumento e, em um estudo de caso, a fonte de dados tipicamente utilizada é a observação. No que se refere ao método de observação empregado, o investigador pode, dentre outros, assumir o papel de observador participante ou não participante, sendo a observação participante o procedimento geralmente mais utilizado nos estudos de caso (GÓMEZ, FLORES e JIMÉNEZ, 1999 *apud* MEIRINHOS e OSÓRIO, 2010).

Na observação participante o investigador pode assumir uma variedade de atribuições, e participar, de fato, dos eventos sendo estudados (YIN, 2001), como no caso desta pesquisa, na qual o investigador assumiu o papel de professor durante a oficina realizada para a coleta de dados, participando assim, ativamente, do fenômeno investigado.

A observação participante não é uma tarefa fácil pois, como visto, exige do pesquisador o desempenho de um duplo papel: de investigador e de participante (MEIRINHOS e OSÓRIO, 2010). No entanto,

[...] o esforço investido, será suficientemente compensado com a qualidade da informação obtida através deste procedimento. O observador participante pode aproximar-se num sentido mais profundo e fundamental às pessoas e comunidades estudadas e aos problemas que as preocupam. Esta aproximação que situa o investigador no papel dos participantes, permite obter percepções da realidade estudada que dificilmente se poderiam conseguir sem se implicar de maneira efetiva (GÓMEZ, FLORES e JIMÉNEZ, 1999, p. 165-166 *apud* MEIRINHOS e OSÓRIO, 2010, p.61)

Apesar dessas vantagens, a observação participante possui o problema da interferência, ou seja, ela tem uma propensão a apresentar “problemas de viés do observador” (COHEN, MANION e MORRISON, 2007, p.256). “A melhor maneira de controlar esta interferência é [o observador ter] consciência de como sua presença afeta o grupo e até que ponto este fato pode ser minimizado ou, inclusive, analisado como dado da pesquisa” (GOLDENBERG, 2004, p.55).

1.5.1 Organização do estudo

Para que se pudesse construir o cenário adequado à realização do estudo de caso aqui apresentado, e se ter à disposição as ferramentas necessárias às investigações pretendidas, este trabalho foi organizado em onze etapas, cada uma objetivando dar o suporte necessário para a adequada execução da etapa posterior e culminando com a realização do estudo de caso em si:

- 1) Estudo dos trabalhos sobre Robótica Educacional desenvolvidos pelo LIVRE.
- 2) Seleção de um conjunto de materiais de robótica, dentre os existentes no mercado, a serem manipulados pelo DB4K, tendo como requisitos básicos: o baixo custo e a compatibilidade com os dispositivos presentes no LabVad Robótica (ver capítulo 3).
- 3) Estudo de ambientes de Programação Visual voltados a crianças.
- 4) Projeto do ambiente de programação DB4K.
- 5) Escolha do ambiente sobre o qual construir o DB4K.
- 6) Construção do Ambiente de Programação Visual em Blocos DB4K.
- 7) Definição das estratégias pedagógicas a serem adotadas para o ensino dos conceitos básicos de programação e robótica contemplados pelo DB4K.
- 8) Desenho de um conjunto de atividades didáticas.
- 9) Desenvolvimento de materiais de robótica auxiliares a serem utilizados nas atividades didáticas planejadas.
- 10) Desenvolvimento de um roteiro de avaliação.
- 11) Realização de uma oficina com alunos do terceiro e quarto anos do Ensino Fundamental, pertencentes a uma comunidade de baixa renda e devidamente matriculados em escolas municipais da cidade do Rio de Janeiro, buscando avaliar qualitativamente:
 - O desenvolvimento e/ou exercício de habilidades do Pensamento Computacional a partir do aprendizado dos conceitos básicos de programação e robótica contemplados na oficina.

- A Usabilidade do ambiente de Programação Visual em Blocos Implementado.
- Os materiais de robótica (tanto os adquiridos no mercado quanto os desenvolvidos especificamente para a oficina).
- As atividades didáticas e estratégias pedagógicas propostas.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 ROBÓTICA EDUCACIONAL

A palavra robô, mencionada pela primeira vez em 1920 na peça R.U.R (*Rossum's Universal Robots*) do escritor tcheco Karel Capek e lançada no Brasil pela editora Hedra com o título de "A Fábrica de Robôs" (TCHÁPEK, 2012), tem origem na palavra tcheca *robota* que significa "trabalho compulsório". Na peça de Capek, um cientista descobre uma maneira de "dar vida" a máquinas de aparência humana capazes de fazer tudo em lugar do homem. Em 1950, Isaac Asimov aborda novamente o tema em seu famoso livro de ficção científica "Eu, Robô" (ASIMOV, 2014), nos apresentando robôs humanoides altamente desenvolvidos. E é este robô, com aparência humana, que povoa o imaginário mundial (ALFARO, 2006; SOUZA e DUARTE, 2015).

No entanto, robôs, de fato, são basicamente qualquer estrutura mecânica autômata¹¹ multifuncional reprogramável operada por meio de circuitos integrados¹², controles eletro-hidráulicos e engrenagens, projetada para movimentar, de diversas formas, uma série de materiais ou dispositivos especializados (SOUZA e DUARTE, 2015; KLOC, KOSCIANSKI e PILATTI, 2009). A construção dessas estruturas e sua manipulação por meio de sistemas baseados em lógica de programação constituem o ramo da ciência denominado robótica (SOUZA e DUARTE, 2015). A robótica encontra atualmente aplicação em diversas áreas, como a indústria, medicina, exploração espacial e também a educação. Quando utilizada com propósitos educacionais a robótica recebe a denominação de Robótica Educacional (SALVINI, KORSAH e NOURBAKHS, 2015).

A ideia do uso da robótica em educação se baseia fortemente na Teoria Construcionista de Seymour Papert, que une a Teoria Construtivista de Jean Piaget - com quem trabalhou entre 1958 e 1963 na Universidade de Genebra, Suíça - ao uso do computador na educação (PINTO, 2011). Papert (1993) adicionou à teoria de

¹¹ 1 - "Máquina ou engenho composto de mecanismo que lhe imprime determinados movimentos (p.ex., um relógio, certos tipos de brinquedo etc.)" (AUTÔMATO, 2001).

2 - "Aparelho com aparência humana, ou de outros seres animados, que reproduz seus movimentos por meios mecânicos ou eletrônicos" (AUTÔMATO, 2001).

¹² Em eletrônica, um circuito integrado (também conhecido como CI, microchip ou chip) é um circuito eletrônico que incorpora miniaturas de diversos componentes.

Piaget (1974), de que a manipulação de objetos é a chave para as crianças construírem seu próprio conhecimento, a ideia de que essa construção se dá de forma mais efetiva quando o aprendiz se engaja de maneira consciente na construção de algo tangível (PAPERT, 1993 *apud* ALIMISIS, 2013).

Dentro desse contexto, Papert criou o LOGO, um software que permite aos usuários, através de linhas de código, movimentar uma “tartaruga”, um animal cibernético que pode ser tanto um objeto virtual (presente na tela do computador) quanto um objeto físico manipulável (PAPERT, 1993; SOLOMON e PAPERT, 1976). Essa tartaruga deixa um “rastro” (uma linha desenhada) por onde “anda” permitindo ao usuário ter um *feedback* imediato dos comandos dados por ele ao computador, uma vez que, ao mesmo tempo em que move a “tartaruga robô”, o usuário constrói elementos gráficos por meio do “rastro” deixado por ela (ALIMISIS *et al*, 2007; SOARES e BORGES, 2011). É neste processo, de reflexão do usuário sobre os resultados concretos dos comandos dados por ele ao computador a partir da observação dos elementos gráficos construídos por meio dos movimentos da tartaruga, que se estabelece a construção do conhecimento.

Desde sua criação por Papert até os dias de hoje, o LOGO, em todas as suas implementações, vem sendo a principal ferramenta de aplicação do “modo de pensar” construcionista, e traz em si uma semente do uso da robótica na educação por meio da utilização da sua tartaruga robô. No entanto, embora as primeiras versões do LOGO fizessem uso de uma tartaruga robô real que se movimentava sobre uma folha de papel (PAPERT, 1993), as versões comumente utilizadas são puramente virtuais, oferecendo uma “arquitetura robótica” pobre e algumas limitações, como a não possibilidade de se observar os possíveis efeitos do ambiente real sobre os experimentos realizados. Além disso, por se tratar de um ambiente 2D, o LOGO impossibilita a realização de tarefas mais “elaboradas” características da robótica propriamente dita que, deste modo, acaba por apresentar-se como sendo uma evolução natural do uso do ambiente de programação LOGO como ferramenta de suporte ao Construcionismo (ALIMISIS *et al*, 2007).

O próprio Papert nos apresenta em seus estudos a proposta da Robótica Educacional como uma nova disciplina de cunho construcionista:

O esboço desta nova disciplina surgirá gradualmente, e o problema de situá-la no contexto da Escola e no ambiente de aprendizagem maior, será melhor apresentado quando o tivermos na nossa frente. Apresento aqui uma

definição preliminar da disciplina – porém apenas como uma semente para discussão – como aquele grão de conhecimento necessário para que uma criança invente (e, evidentemente, construa) entidades com qualidades evocativamente semelhantes à vida dos mísseis inteligentes. Se este grão constituísse a disciplina inteira um nome adequado seria “engenharia de controle” ou até mesmo “robótica” (PAPERT, 1994, p.160).

No entanto, até pouco tempo, o uso da robótica na educação ficava inviabilizado pela falta de acesso a robôs, ou aos componentes necessários à sua construção, devido principalmente ao seu alto custo. Mas, recentes avanços em relação ao preço, tamanho, performance e capacidade dos componentes utilizados para a construção e manipulação de robôs fizeram surgir no mercado uma série de Kits de Robótica Educacional, alguns proprietários, como o Lego Mindstorms¹³ e outros baseados em Hardware Livre (ver 2.6) como o Arduino (ver 2.7), resultando num perceptível crescimento no uso da robótica na área educativa.

Além dos ganhos para aprendizagem relacionados ao caráter construcionista do uso da robótica na educação já abordados aqui, por tratar-se de uma disciplina de caráter multidisciplinar, a robótica permite também que os alunos trabalhem uma grande diversidade de competências e habilidades, à medida que engloba, em um único objeto de estudo, diferentes áreas do conhecimento como matemática, eletrônica, mecânica e programação de computadores. O uso da robótica no processo de ensino aprendizagem também estimula a colaboração, a habilidade de trabalhar em grupo, a elaboração de estratégias para a solução de problemas e o desenvolvimento do pensamento formal¹⁴ (SOUZA e DUARTE, 2015; LÓPEZ-RODRÍGUEZ e CUESTA, 2015; PEREZ *et al*, 2013; RUS, 2006; DENIS e HUBERT, 2001), além de promover uma melhora na capacidade de raciocínio lógico dos alunos, conforme apontam os estudos de Buechley, Eisenberg e Elumeze (2007) e Barbero e Demo (2011).

Robôs instigam naturalmente a curiosidade de crianças e adultos (PEREZ *et al*, 2013) motivo este, entre outros tantos, que vem fazendo com que

A Robótica Educacional [...]venha] emergindo como uma ferramenta de aprendizagem capaz de oferecer atividades práticas e divertidas em um ambiente de aprendizagem atrativo, alimentando o interesse e a curiosidade dos alunos (EGUCHI, 2010 *apud* ALIMISIS, 2013, p.63, tradução nossa).

¹³ <https://www.lego.com/en-us/mindstorms>

¹⁴ “[O pensamento formal] constitui um sistema complexo, mas coerente, relativamente diferente da lógica da criança: constitui a essência da lógica dos adultos cultivados assim como a base das formas elementares do pensamento científico” (PIAGET, 1972, p.6 *apud* MARCHAND, 2002, p.191).

No entanto, embora diferentes estudos venham apresentando resultados promissores em relação ao uso da robótica na educação, por ser esta uma prática bastante recente, existem ainda muitas áreas obscuras envolvendo esse campo de estudo, o que acaba por fazer com que o papel do uso de robôs no processo ensino aprendizagem não esteja perfeitamente claro (SALVINI, KORSAH e NOURBAKHSH, 2015). Isto faz com que trabalhos como este se apresentem como sendo de significativa relevância para o auxílio no esclarecimento de questões ainda em aberto no tocante ao uso da robótica na educação.

2.2 ENSINO DE PROGRAMAÇÃO PARA CRIANÇAS

Embora o tema “ensino de programação para crianças” tenha emergido com mais força nos últimos anos, este não é necessariamente recente. Como visto na seção anterior, já no final dos anos sessenta, Seymour Papert desenvolvia trabalhos nesse sentido por meio da utilização da linguagem LOGO. Desde então, suas pesquisas em relação ao uso de informática na educação avançaram em constante evolução dando origem a uma série de publicações sobre o tema, como os livros “*Mindstorms: Children, Computers and Powerful Ideas*”, lançado na década de 80 (PAPERT, 1993) e “*The Children's Machine: Rethinking School In The Age Of The Computer*”, da década de 90 (PAPERT, 1994).

A revisão da bibliografia nos permite encontrar outros trabalhos das décadas de 80 e 90 dedicados ao estudo dos possíveis benefícios advindos do ensino de programação de computadores a crianças, como o trabalho de Clements e Gullo (1984), que apresenta uma análise dos efeitos da prática de programação de computadores na cognição de crianças com seis anos de idade. Os pós-testes desse trabalho mostraram um ganho significativo em relação ao pensamento reflexivo¹⁵ e divergente¹⁶ das crianças que participaram das atividades de programação de computadores, enquanto que outro grupo, que usou o computador para atividades de

¹⁵ “[Pensamento reflexivo é] a espécie de pensamento que consiste em examinar mentalmente o assunto e dar-lhe consideração séria e consecutiva” (DEWEY, 1959, p.13).

¹⁶ O pensamento divergente está relacionado à criatividade. Ele é responsável pela solução de problemas para os quais ainda não exista uma solução conhecida. Esse tipo de pensamento estaria associado ao levantamento de hipóteses, à capacidade de se imaginar uma variedade de soluções diferentes para um determinado problema (PELAES, 2010).

Instrução Assistida por Computador (*Computer Assisted Instruction* - CAI), não apresentou ganhos em relação às avaliações pré-teste.

Os estudos de Papert, e de Clements e Gullo, apontam para a existência de ganhos reais promovidos pelo ensino de programação, tanto na qualidade da aprendizagem de modo geral, quanto em relação a aspectos cognitivos específicos, o que vem a reforçar a ideia de ser esta uma boa prática a ser adotada na Educação Básica.

Além das características positivas apontadas por esses estudos, relacionadas à melhora na qualidade do aprendizado, a crescente e cada vez mais acelerada inserção das TIC no dia a dia de toda a sociedade acabou por trazer uma nova dimensão ao aprendizado de programação por crianças: Atualmente, saber programar tornou-se tão essencial quanto saber escrever (RESNICK, 2014).

Para Resnick, uma pessoa não aprende a escrever porque vai, necessariamente, tornar-se um escritor profissional, mas sim, porque saber escrever é algo útil em todos os aspectos de nossas vidas, pois através da escrita nos é possível expressar nossas ideias de uma maneira diferente. Por meio dela podemos ver e fazer os outros verem o mundo de diferentes maneiras. Enfim, podemos criar e transformar o mundo a partir desta criação. Do mesmo modo, nos dias de hoje, aprender a programar não é somente útil para quem vai tornar-se um programador profissional, um cientista da computação, mas sim, para que possamos nos tornar de fato fluentes nas novas tecnologias e, deste modo, as utilizarmos para organizar e expressar nossas ideias. Assim como saber escrever, saber programar é hoje algo útil em todos os aspectos de nossas vidas.

Esse entendimento de que o aprendizado da programação de computadores é algo essencial na sociedade atual tem feito surgir uma série de iniciativas no sentido de tornar esse aprendizado possível e acessível a qualquer pessoa, desde os primeiros anos da vida escolar até a vida adulta. Algumas dessas iniciativas são a Codecademy¹⁷, Code.org¹⁸, Programaê¹⁹, Code Club Brasil²⁰ e o próprio Scratch²¹ de Mitchel Resnick.

¹⁷ <https://www.codecademy.com/pt>

¹⁸ <https://code.org/>

¹⁹ <http://www.fundacaoolemann.org.br/programae/>

²⁰ <http://codeclubbrasil.org/>

²¹ <https://scratch.mit.edu/>

Uma rápida pesquisa na internet nos permite encontrar também inúmeras reportagens que apontam o crescente número de escolas brasileiras, em sua grande maioria escolas da Rede Particular de Ensino, que vêm incluindo o ensino de programação no seu currículo, estando esta prática, muitas vezes, associada a atividades de Robótica Educacional.

Como já mencionado no capítulo 1.3, diante deste quadro, torna-se fundamental o desenvolvimento de alternativas que busquem possibilitar que essas iniciativas sejam estendidas também ao Ensino Público, buscando-se evitar com isso uma ampliação ainda maior do *gap* existente entre a qualidade do ensino público e a do ensino privado.

O DuinoBlocks4Kids se insere exatamente neste contexto, como uma ferramenta gratuita de apoio ao ensino de programação para crianças, baseada em software livre e associada a hardware de baixo custo (ver 2.6).

2.3 PENSAMENTO COMPUTACIONAL

Jan Cuny, Larry Snyder e Janet Wing definem o Pensamento Computacional (*Computational Thinking*) como sendo "o processo de pensamento envolvido na formulação de problemas e de suas soluções de maneira que essas soluções estejam representadas de forma a poderem ser efetivamente executadas por um agente processador de informações" (WING, 2011, tradução nossa).

Para Hemmendinger (2010), o objetivo do desenvolvimento do Pensamento Computacional não é o de fazer com que todos passem a pensar como cientistas da computação, mas sim, habilitar as pessoas a aplicarem esta maneira específica de raciocinar na busca por novos questionamentos e na solução de diversos tipos de problemas nas mais variadas áreas do conhecimento (BARR e STEPHENSON, 2011).

“Pensar Computacionalmente” é reformular um problema aparentemente difícil de maneira a “transformá-lo” em um que saibamos resolver, é prevenir erros e estar pronto para corrigi-los, revisando cada etapa realizada na busca pela solução de um problema. É saber planejar na presença de incertezas e entender que é possível trabalhar de forma segura com problemas complexos sem precisar conhecer todos os seus detalhes (WING, 2006).

Como já mencionado, os benefícios da capacidade de Pensar Computacionalmente transportam-se para qualquer domínio por meio do aumento e reforço de uma série de habilidades intelectuais (WING, 2011). E, embora não haja ainda consenso acerca do conjunto exato dessas habilidades, de acordo com Grover e Pea (2013), a maior parte dos pesquisadores e educadores da área de Ciências da Computação têm aceitado, de forma bastante ampla, esta forma de pensamento como sendo compreendida pelos seguintes elementos:

- Abstração e generalização de padrões (Incluindo modelagem e simulações).
- Processamento sistemático de informações.
- Sistemas simbólicos e representações.
- Noções algorítmicas de fluxo de controle.
- Decomposição estruturada de problemas.
- Pensamento paralelo, recursivo e iterativo.
- Lógica condicional.
- Condicionantes de eficiência e performance.
- Depuração e detecção sistemática de erros.

Como se pode observar, o Pensamento Computacional abarca muitos elementos que, para serem trabalhados, necessitam do exercício e desenvolvimento de uma gama ampla de habilidades. Para se abarcar todo esse conjunto de habilidades torna-se necessária a realização de um trabalho extenso e aprofundado de diversos aspectos relacionados às Ciências da Computação, ou seja: Pensar Computacionalmente, ao contrário do que se pode imaginar em um primeiro momento, não se restringe à habilidade de programar computadores. No entanto, tem-se na programação (competência fundamental da Ciência da Computação) uma importante ferramenta de apoio ao desenvolvimento e exercício de algumas das habilidades cognitivas características do Pensamento Computacional (GROVER e PEA, 2013), e é nesse âmbito que se insere essa pesquisa.

Por se tratar de um tema relativamente novo, há muito ainda que se estudar acerca das possíveis maneiras de se trabalhar o Pensamento Computacional nas escolas e de que forma o aprendizado de programação, já nos primeiros anos do Ensino Fundamental, pode contribuir para o seu desenvolvimento. Espera-se, desse

modo, que esta pesquisa possa auxiliar na busca por essas descobertas, e oferecer também ferramentas que venham a colaborar para tornar o exercício de algumas habilidades cognitivas relacionadas ao Pensamento Computacional, por meio da programação de computadores, mais acessível a crianças pertencentes a famílias de baixa renda.

2.4 O PERÍODO OPERATÓRIO CONCRETO E O APRENDIZADO DE PROGRAMAÇÃO

Esta seção tem por objetivo oferecer uma breve visão da teoria de Jean Piaget no que se refere a determinadas características cognitivas de crianças com idade entre oito e dez anos, por ser esta a faixa etária dos sujeitos desta pesquisa. A escolha deste teórico como referência para a observação de aspectos relacionados à maturidade cognitiva deve-se ao fato do aprendizado de programação associado à Robótica Educacional ser fortemente baseado na Teoria Construcionista de Seymour Papert, que, conforme visto na seção 2.1, une a Teoria Construtivista de Jean Piaget ao uso do computador na educação. Busca-se também estabelecer aqui uma relação entre as habilidades cognitivas características de crianças operacionais concretas e as competências necessárias para se realizar determinadas atividades relacionadas ao aprendizado de programação de computadores.

Jean Piaget distingue quatro períodos gerais no que se refere ao desenvolvimento cognitivo, sendo eles: sensório motor [0 a 2 anos], pré-operatório [2 a 7/8 anos], operatório concreto [7/8 a 11/12 anos], operatório formal [11/12 anos em diante] (MOREIRA, 1999).

Todos os indivíduos passam por todas essas fases ou períodos, nessa sequência, porém o início e o término de cada uma delas dependem das características biológicas do indivíduo e de fatores educacionais, sociais. Portanto, a divisão nessas faixas etárias é uma referência, e não uma norma rígida (FURTADO, BOCK e TEIXEIRA, 2001, p.102).

O período operatório concreto, que tem início por volta dos sete/oito anos, caracteriza-se como sendo uma fase de transição entre a ação e as estruturas lógicas mais gerais, como classificação, seriação, ordenação, agrupamento, etc. (SOUZA e WECHSLER, 2014), ou seja, as crianças operacionais concretas começam a realizar operações mentalmente e não apenas por meio de ações físicas como acontecia no período pré-operatório (TERRA, 2010). Em outras palavras: "o sujeito se torna capaz

de reconstruir no plano da representação o que já havia construído no plano da ação" (SOUZA e WECHSLER, 2014, p.144).

O período é dito "operatório" por ter início nele, como mencionado anteriormente, o desenvolvimento de uma nova habilidade mental: as "operações". Isso significa que a criança desenvolve a capacidade de realizar uma ação (física ou mental) voltada para um objetivo específico e reverter, ou retroceder, essa ação de qualquer ponto a qualquer ponto entre seu início e fim. Esse tipo de habilidade mental pode ser observado, por exemplo, quando uma criança, ao montar um quebra-cabeças e perceber que algumas peças não foram colocadas no lugar correto, consegue retirar apenas as peças posicionadas incorretamente e recomeçar o trabalho deste ponto em diante, utilizando novas peças no lugar das que foram retiradas ou reorganizando-as de outra forma rumo ao objetivo inicialmente planejado (FURTADO, BOCK e TEIXEIRA, 2001).

Outro aspecto fundamental em relação a esse período é que, para realizar as "operações", a criança recorre sempre a "objetos concretos" presentes ou já experimentados (daí a designação operatório concreto), sendo bastante limitada a realização de operações concretas em direção ao ausente. Para antecipar o ausente a criança tem que partir do concreto, diferente do que acontece no período seguinte (operatório formal), quando o real é percebido como um caso particular do possível. Ou seja, no período operatório concreto, os conhecimentos não são construídos a partir de uma definição, mas sim, de uma situação, daquilo que é perceptivo, sendo necessária a comparação daquilo que está sendo aprendido com o que já é conhecido ou está sendo fisicamente visualizado. Deste modo, mesmo a capacidade de reflexão, qual seja: de ponderar antes de agir, resgatar o passado, antecipar o futuro, considerar mais de um ponto de vista ao mesmo tempo, etc.; que também tem início nessa fase, sempre ocorre a partir de situações presentes ou já experienciadas pela criança (FURTADO, BOCK e TEIXEIRA, 2001; MOREIRA, 1999; SOUZA e WECHSLER, 2014; PEDROZO, 2014).

As crianças nesse período possuem, entre outras, as seguintes habilidades:

- Estabelecer corretamente as relações de causa e efeito e de meio e fim.
- Sequenciar ideias ou eventos.
- Trabalhar com ideias sob dois pontos de vista, simultaneamente.
- Formar o conceito de número (no início do período operatório concreto, sua noção de número está vinculada a uma correspondência com o objeto concreto). (FURTADO, BOCK e TEIXEIRA, 2001, p.102).
- Pensar, simultaneamente, no todo e nas suas partes. (MOREIRA, 1999, p.98).

Algumas das características cognitivas do período operatório concreto aqui mencionadas permitem supor serem adequadas a crianças dessa idade a prática de algumas atividades relacionadas à programação de computadores, como por exemplo:

- A habilidade de sequenciar eventos ou ideias possibilitaria o desenvolvimento de programas puramente sequenciais, que, grosso modo, nada mais são do que um sequenciamento de comandos simples “posicionados” de forma a produzir um resultado esperado.
- Por construírem seu conhecimento a partir de uma ação, e não de um conceito, é necessário que as crianças operacionais concretas vejam “fisicamente” o resultado da execução de um comando para então poder concebê-la mentalmente. Mas, uma vez isso feito, a habilidade delas de realizar operações mentais (a partir de experiências concretas) e não apenas por meio de ações físicas, e de pensar simultaneamente no todo e nas suas partes, possibilitaria a essas crianças construírem mentalmente a sequência de eventos resultantes de um conjunto de comandos utilizados no programa. Ou seja, essa habilidade permitiria às crianças operacionais concretas a construção de um trecho completo de código (composto por um determinado número de comandos) e a “visualização mental” do resultado da execução desse trecho de código sem a necessidade de observarem “fisicamente”, a todo instante, o resultado da execução individual de cada um dos comandos.
- A capacidade de estabelecer corretamente relações de causa e efeito, meio e fim, possibilitaria trabalhar-se com crianças no período operatório concreto a ideia de processamento, qual seja, de que os comandos presentes em um programa são o “meio” pelos quais se obtém o resultado observado quando da execução desse programa, ou seja, o “fim”.
- Ser capaz de trabalhar simultaneamente com mais de um ponto de vista sobre uma mesma ideia, possibilitaria às crianças operacionais concretas perceberem e compreenderem que um mesmo problema pode ter mais de uma solução, e que uma mesma solução pode ser aplicada a mais de um problema, ou seja, que um mesmo resultado pode

ser obtido a partir de diferentes programas e que um mesmo programa pode ser utilizado para solucionar mais de um problema.

- A reversibilidade possibilitaria a essas crianças serem capazes de, a partir da constatação de que algo no resultado obtido não se deu como o esperado, modificar ou retirar do programa apenas os comandos que elas acreditam ser a causa do problema, construindo o programa correto a partir desse ponto, ou seja, sem desfazer-se de todo o código desenvolvido anteriormente para reconstruir, a partir do começo, passo a passo, um novo programa. Além disto, permitiria a elas fazerem o “reaproveitamento de código” quando os objetivos de um programa sendo construído diferirem apenas parcialmente de um programa previamente desenvolvido.

E, por fim, a capacidade de formar o conceito de número nos leva a uma questão chave no que se refere à programação de computadores e ao desenvolvimento do Pensamento Computacional: a capacidade de abstração.

Os números, em sua concepção infinita, são abstrações possíveis de serem construídas somente a partir do que Piaget denominou como sendo uma abstração reflexiva (ou reflexionante): uma construção feita pela mente sem a necessidade de um referencial concreto. Este tipo de abstração é característica do período operatório formal, mas, já se encontra presente em estágios anteriores, nos quais a abstração dominante é a empírica (ou simples), que consiste na capacidade da criança de focalizar em uma determinada propriedade de um objeto e ignorar as demais (KAMII, 1992). "A abstração empírica apoia-se nos objetos físicos ou nos aspectos materiais da própria ação" (PIAJET, 1980 *apud* TAXA, 2001, p.27).

No início do estágio operatório concreto a abstração reflexiva não pode ocorrer independente da empírica, de modo que, como já mencionado, a noção de número, nesse momento, encontra-se vinculada a uma equivalência com o objeto concreto. No entanto, no decorrer do seu desenvolvimento cognitivo, a criança adquire a capacidade de utilizar a abstração reflexiva sem que esta dependa da abstração empírica. Somente assim ela poderá, por exemplo, entender o número 10.000 sem que para isso precise ter visto ou contado 10.000 objetos em um conjunto (FURTADO, BOCK e TEIXEIRA, 2001; KAMII, 1992).

Ou seja, ainda que de forma limitada, crianças no período operatório concreto já possuem a habilidade de trabalhar com abstrações, o que tornaria possível desenvolver-se atividades de programação que propiciem a prática, e com isso o aprimoramento, desta habilidade tão fundamental para o Pensamento Computacional. No entanto, há que se estar atento para perceber se a capacidade de abstração exigida para essas atividades é compatível com a maturidade cognitiva das crianças às quais elas se destinam. Principalmente ao se levar em consideração que algumas dessas crianças poderão apresentar ainda, em relação a determinados aspectos cognitivos, uma maturidade característica do período pré-operatório.

De modo geral, para crianças no período operatório concreto, o uso da abstração empírica, ou seja, a construção de raciocínios a partir da abstração de objetos pertencentes ao seu universo, é uma tarefa rotineira. Já o pensamento hipotético-dedutivo, característico do período operatório formal, para o qual seria necessário a construção de abstrações a partir de hipóteses, e não com base em situações familiares, tende a não aparecer nesse período (LISTER, 2011).

Por possuir uma capacidade de abstração ainda restrita, quando defrontadas com conceitos muito abstratos, crianças no período operatório concreto tendem a

[...] reduzir o nível de abstração de modo a tornar esses conceitos [...] significativos e mentalmente acessíveis [...] lidando para isso com exemplos específicos ao invés de lidar com todo um conjunto genericamente conceituado (HAZZAN, 2008 *apud* LISTER, 2011, p.15, tradução nossa).

A abordagem conceitual [dessas crianças] de forma alguma é "ininteligente", e certamente resulta em tentativas de solução que são mais racionais e relevantes do que aquelas que as crianças pré-operacionais são capazes de produzir. No entanto, essas tentativas abraçam intimamente o terreno da realidade empírica detectada, e especulações [...] ocorrem com muita dificuldade e somente como último recurso. [...] para o "pensador operatório concreto", o "reino da possibilidade abstrata" é visto como incerto e apenas uma extensão do "reino da realidade palpável", muito mais seguro e certo (FLAVELL, MILLER e MILLER, 1989 *apud* LISTER, 2011, p.15, tradução nossa).

Desse modo, determinados conceitos utilizados em computação, como a generalização e a recursividade, não seriam, em princípio, passíveis de serem trabalhados no período operatório concreto. No entanto, algumas estruturas de programação, como as de repetição (contada e condicional) e de decisão (simples e composta), possuem aspectos que podem ser observados ou experienciados, tanto por meio de brincadeiras que não necessitem do uso do computador quanto com a realização de atividades de programação cujos resultados possam ser visualizados e

apresentem de forma bastante evidente uma relação direta com o “comando” que os produziu. Assim sendo, é bastante razoável acreditar que a realização dessas abstrações seja perfeitamente acessível a crianças do período operatório concreto, uma vez que não é exigido que elas sejam construídas exclusivamente a partir de conceitos excessivamente abstratos.

É também interessante perceber que existem alguns conceitos das ciências da computação relacionados à abstração que possuem correspondência direta com a abstração empírica. Este tipo de abstração se encaixa perfeitamente em um dos aspectos relacionados à abstração apresentados por (KRAMER, 2007) como sendo fundamentais para os cientistas da computação, qual seja: o processo de remover detalhes para simplificar e focar a atenção, seja “retirando alguma coisa” ou desconsiderando uma ou mais propriedades de um objeto complexo para prestar atenção em outros.

Um exemplo específico do uso desse tipo de abstração nas ciências da computação pode ser observado na definição de abstração utilizada no paradigma de programação orientada a objetos: “Abstração é a supressão intencional, ou ocultação, de determinados detalhes de um processo ou artefato, com o intuito de evidenciar com mais clareza outros aspectos, detalhes ou estruturas” (BUDD, 2001, p.15, tradução nossa). O mesmo se aplica ao conceito de “níveis de abstração”, definido por Wing (2006) como sendo uma habilidade fundamental para que se possa desenvolver o Pensamento Computacional.

[O uso de níveis de abstração possibilita] que detalhes de implementação em níveis mais baixos de um projeto sejam ignorados ou simplificados, tornando assim mais simples o desenho de componentes em níveis mais altos. Os detalhes de um HD, por exemplo, que é dividido em trilhas e setores, são abstraídos pelo sistema operacional (SO) de modo que o disco apareça para uma aplicação de software como um conjunto de arquivos de tamanhos variáveis. Um programador de aplicações pode então criar, escrever e ler arquivos sem ter conhecimento de como o HD é construído e organizado. (SMITH e NAIR, 2005, p.1, tradução nossa).

Como se pode observar, de acordo com os pressupostos da teoria de Jean Piaget, parece ser perfeitamente possível se desenvolver, com crianças do Ensino Fundamental I, atividades relacionadas ao aprendizado de programação e ao exercício de habilidades cognitivas do Pensamento Computacional.

2.5 LINGUAGENS DE PROGRAMAÇÃO VISUAL PARA CRIANÇAS

Junto com os primeiros computadores pessoais (PCs), no fim dos anos setenta, surgiu também um grande interesse de se utilizar os computadores nas escolas para o aprendizado de programação. Assim, nessa época, inúmeras escolas americanas ensinaram milhares de crianças a escreverem pequenos programas em linguagens como LOGO (ver 2.1) ou BASIC²². No entanto, embora muitos alunos e professores tenham se entusiasmado com a ideia, com o decorrer dos anos, a maior parte das escolas passou a utilizar os computadores para outros fins que não o ensino de programação (RESNICK *et al*, 2009).

Mas, por que as iniciativas de ensino de programação para crianças, como o uso do LOGO, proposto por Papert, não “vingaram”? Por algumas razões, dentre elas o fato de que as linguagens de programação da época eram muito difíceis, de maneira que muitas crianças simplesmente não conseguiam aprender sua sintaxe²³. Além disso, os programas desenvolvidos não tinham conexão alguma com os interesses das crianças, eram, por exemplo, programas desenhados para gerar listas de números ou traçar algumas linhas na tela, sendo que muitas vezes não era possível à criança perceber, por si só, se o resultado que o programa por ela escrito havia produzido era realmente o resultado esperado, e tampouco esses resultados davam margem para a discussão de outras questões que não a simples verificação da correção ou não do programa escrito (RESNICK *et al*, 2009).

No decorrer do tempo, algumas iniciativas surgiram no sentido de se tentar vencer essas dificuldades. No que se refere ao entendimento da sintaxe das linguagens de programação textuais, uma alternativa encontrada foi o uso de Linguagens de Programação Visual (*Visual Programming Language* - VPL), ou seja, linguagens nas quais “a sintaxe (semanticamente significativa) inclui expressões visuais” (BURNETT, 1999, p.1, tradução nossa).

²² BASIC, acrônimo para “*Beginner’s All-purpose Symbolic Instruction Code*”, é uma linguagem de programação de “fácil” aprendizagem, desenvolvida na década de 60 para fins didáticos, e que se tornou muito popular nos anos 70 e 80 com o advento dos computadores pessoais.

²³ Sintaxe, em linguagens de programação, diz respeito ao conjunto de regras que define a forma de uma linguagem, estabelecendo como são compostas as suas estruturas básicas, ou seja, quais palavras e símbolos fazem parte desta linguagem e de que forma eles podem ser combinados e arranjados.

Com base nesse paradigma, o MIT (*Massachusetts Institute of Technology*) Media Lab²⁴, iniciou, em 2003, o desenvolvimento do Scratch²⁵, uma linguagem de programação que tinha por objetivo permitir que qualquer pessoa, de qualquer idade, pudesse programar. Tempos depois, em 2007, o MIT criou um site para o Scratch, por meio do qual os usuários podiam compartilhar os projetos por eles desenvolvidos, todos com código aberto. O sucesso foi tal que, em 2009, apenas dois anos depois do lançamento do site, usuários de todas as partes do mundo, na sua maioria crianças e jovens de oito a dezesseis anos, já faziam upload de mais de 1500 projetos por dia (RESNICK *et al*, 2009). Mais recentemente, no ano de 2013, o MIT Media Lab lançou o ScratchJr.²⁶ (FLANNERY *et al*, 2013), uma versão do Scratch voltada ao ensino de conceitos de programação a crianças de cinco a sete anos.

Um dos principais conceitos utilizados para o projeto do Scratch foi o uso de “blocos de encaixar”. Baseados na experiência de trabalho com a Lego, onde ajudaram no desenvolvimento do Lego MindStorms²⁷, os criadores do Scratch decidiram pelo desenvolvimento de uma linguagem de programação constituída de blocos de encaixar, o que permitiria às crianças organizarem e reorganizarem suas ideias, montar e desmontar o seu programa, da mesma forma como criavam e recriavam objetos diferentes a partir das inúmeras maneiras possíveis de encaixar e combinar os blocos Lego. Além disso, os efeitos do uso de cada bloco e da combinação destes poderia ser observado de imediato por meio de elementos gráficos (*sprites*) presentes na interface do programa.

O Scratch é atualmente a principal referência no que diz respeito ao ensino de programação para crianças, permitindo que as mesmas trabalhem conceitos de matemática e computação, bem como desenvolvam a criatividade, o pensamento sistemático e a habilidade de trabalhar em grupo (RESNICK *et al*, 2009).

Além do Scratch, outros projetos de grande vulto, voltados ao ensino de programação para crianças, também adotam o conceito da programação visual por meio de blocos de encaixar. Dentre eles podemos citar o Code.org²⁸, que possui suas próprias ferramentas de ensino de programação por blocos e o Programa²⁹ e Code

²⁴ <https://ilk.media.mit.edu/>

²⁵ <https://scratch.mit.edu/>

²⁶ <http://www.scratchjr.org/>

²⁷ <https://www.lego.com/en-us/mindstorms/>

²⁸ <https://code.org/>

²⁹ <http://www.fundacaolemann.org.br/programae/>

Club Brasil³⁰, que utilizam o Scratch, dentre outras ferramentas, em seus programas de aprendizado de programação. A adoção, por esses projetos, de ambientes de Programação Visual em Blocos, sugerem ser esta uma abordagem bastante acertada no que diz respeito ao ensino de programação para crianças, o que veio a motivar o uso deste mesmo paradigma no desenvolvimento do DuinoBlocks4Kids.

Outro aspecto importante, no que diz respeito ao desenvolvimento de uma Linguagem de Programação Visual em Blocos voltada a crianças do Ensino Fundamental I para a programação de placas de prototipagem eletrônica Arduino, pode ser observado no que Papert dizia em relação às características que uma linguagem de programação deveria ter. Segundo Papert “[...] as linguagens de programação deveriam ter um ‘ piso baixo ’ (fácil de começar) e um ‘ teto alto ’ (oportunidades de se criar projetos com complexidade crescente com o passar do tempo)” (RESNICK *et al*, 2009, p.63, tradução nossa).

No que se refere à programação de placas de prototipagem eletrônica Arduino, conta-se atualmente com algumas ferramentas que permitem a programação dessas placas por meio da linguagem textual *Wiring*³¹, como o IDE³² fornecido pelo projeto Arduino³³, e alguns ambientes de programação em Blocos voltados ao público jovem e adulto, como o DuinoBlocks (ALVES, 2013) e o Ardublockly³⁴. Esses ambientes podem ser entendidos como sendo o “teto alto” descrito por Papert. O DuinoBlocks4Kids se integraria a esse conjunto de ferramentas como sendo o “piso baixo”, buscando facilitar, o quanto possível, o primeiro contato dos usuários com o aprendizado de programação de computadores por meio da Robótica Educacional com o uso da plataforma de prototipagem eletrônica Arduino.

2.6 SOFTWARE E HARDWARE LIVRES

O Movimento Software Livre é "baseado no princípio do compartilhamento do conhecimento e na solidariedade praticada pela inteligência coletiva" (DA SILVEIRA,

³⁰ <http://codeclubbrasil.org/>

³¹ Linguagem nativa para a programação das placas de prototipagem eletrônica Arduino, derivada da linguagem C/C++.

³² Um IDE (*Integrated Development Environment*) é um software que integra várias ferramentas de apoio ao desenvolvimento de programas de computador. O IDE auxilia o programador a editar o código do programa e posteriormente traduz esse código em uma linguagem que o computador possa “entender”, de modo que o programa possa ser executado pelo processador.

³³ <https://www.arduino.cc/en/Main/Software>

³⁴ <https://github.com/carlosperate/ardublockly>

2003, p.36). Ele surgiu de forma oficial e organizada, em 1985, com a criação da FSF (*Free Software Foundation*)³⁵. A FSF teve origem no descontentamento de Richard Stallman, do MIT, com a prática do desenvolvimento de softwares proprietários, que impedia que o código fonte de um software pudesse ser acessado e utilizado livremente para o desenvolvimento de novas aplicações (DA SILVEIRA, 2003).

Foi um movimento que começou pequeno, mas que atualmente conta com milhares de colaboradores e disponibiliza a toda comunidade global uma gama significativa de softwares com código aberto de grande qualidade, entre eles, sistemas operacionais, como algumas versões do sistema operacional Linux, entre elas o Ubuntu³⁶, linguagens de programação como Python³⁷ e Java³⁸, *suites office* como o LibreOffice³⁹ e ferramentas para o desenvolvimento de sites, como o Joomla⁴⁰. Além desses projetos de maior vulto e amplamente conhecidos, existem também, disponíveis em repositórios de Software Livre como o GitHub⁴¹, um número incalculável de softwares, cujos códigos podem ser livremente baixados, utilizados, modificados e redistribuídos, permitindo a qualquer pessoa auxiliar, em qualquer tempo, na melhoria dos softwares lá disponibilizados, bem como desenvolver, individual ou coletivamente, novos programas baseados nos códigos desses softwares.

O movimento Software Livre conta com algumas organizações internacionais responsáveis por promover e preservar a liberdade desses softwares, definindo, entre outras coisas, os termos relativos às licenças dos mesmos. A licença mais conhecida atualmente é a GNU GPL (*General Public License*), definida pela FSF. Essa licença mantém os direitos autorais do desenvolvedor do programa, mas, dá uma série de liberdades aos usuários, entre as quais, o direito de estudar, alterar, copiar e redistribuir o software (NETO, 2010).

Outro movimento relacionado à democratização do acesso à tecnologia é o Movimento Hardware Livre. Hardware Livre é um hardware eletrônico projetado e oferecido da mesma maneira que um Software Livre. Um exemplo de hardware desse tipo são as placas de prototipagem eletrônica Arduino (ver 2.7), tecnologia adotada

³⁵ <http://www.fsf.org/>

³⁶ <http://www.ubuntu.com/>

³⁷ <https://www.python.org/>

³⁸ <https://www.java.com/>

³⁹ <https://pt-br.libreoffice.org/>

⁴⁰ <https://www.joomla.org/>

⁴¹ <https://github.com/>

pelo LabVad. Os esquemas para o desenvolvimento dos componentes eletrônicos das placas Arduino são disponibilizados a toda a comunidade, de modo que qualquer pessoa possa construir seus próprios dispositivos baseados nesses desenhos, o que acaba por popularizar e, conseqüentemente, baratear muito o custo desses hardwares (SOUZA *et al*, 2014).

Um aspecto bastante importante em relação ao desenvolvimento de aplicações a partir do uso de Software Livre e da associação destes ao uso de hardware de baixo custo, em contraponto ao uso de soluções proprietárias, é a democratização do acesso à tecnologia.

Como já mencionado, é fato notório que existe hoje no Brasil uma defasagem entre a qualidade dos Ensinos Público e Privado de Nível Básico. Investir em projetos que busquem possibilitar às escolas públicas oferecerem a seus alunos um processo de inclusão digital baseado em práticas relacionadas ao uso de informática na educação com qualidade competitiva em relação àquelas utilizadas pelas escolas particulares (no que a utilização de Tecnologia Livre é um forte aliado), é uma forma de se buscar contribuir com o combate aos danos sociais resultantes das diferenças na qualidade do processo de ensino-aprendizagem hoje observadas.

O Projeto LabVad é uma dessas iniciativas, estando atualmente baseado exclusivamente no uso de Software Livre e Hardware de baixo custo, tendo sido também estas as tecnologias utilizadas como base para o desenvolvimento do DuinoBlocks4Kids.

2.7 ARDUINO

Arduino é um projeto de Hardware Livre desenvolvido por um grupo de cinco pesquisadores de diferentes nacionalidades no Interaction Design Institute Ivrea⁴², localizado na cidade de Ivrea, Itália, que tinha por objetivo a criação de uma plataforma de prototipagem eletrônica que fosse simples de ser utilizada por qualquer estudante, inclusive crianças, de maneira que eles pudessem facilmente criar “coisas” que funcionassem sem a necessidade de frequentar um curso completo de eletrônica (BANZI, 2012).

⁴² <https://interactionivrea.org/>

Por ser um hardware de código aberto, todos os arquivos com os esquemas dos circuitos estão disponíveis para *download*. Além disso, toda a documentação e metodologias de ensino também estão disponíveis e podem ser livremente utilizadas. Só o que é protegido é o nome da marca (BANZI, 2012).

As placas de prototipagem eletrônica Arduino, e aquelas projetadas a partir do desenho destas, têm permitido o desenvolvimento de uma série de projetos interessantes que, de outra forma, possivelmente não existiriam. Como exemplo, temos o desenvolvimento de projetos de *drones* de hardware de código aberto que permitiram que os mesmos pudessem ser produzidos a muito baixo custo por uma *startup* chamada Matternet⁴³ para serem utilizados em uma série de ações humanitárias, entre elas, o transporte de pequenos objetos, como medicamentos, entre aldeias africanas (BANZI, 2012; ALLEN, 2016).

Além de projetos de “grande porte”, como o da Matternet, existem também inúmeros exemplos de projetos desenvolvidos por crianças e que podem ser facilmente encontrados em sites e comunidades virtuais dedicadas ao compartilhamento de projetos baseados em placas Arduino, como a Hackster.io⁴⁴, FacaComArduino⁴⁵ e Laboratório de Garagem⁴⁶.

O seu baixo custo, qualidade, flexibilidade e facilidade de uso, têm feito das placas de prototipagem eletrônica Arduino uma excelente opção para o desenvolvimento de projetos na área de Robótica Educacional.

2.8 COMPUTAÇÃO DESPLUGADA

Computação Desplugada - *CS Unplugged*⁴⁷ (BELL *et al*, 2009) é uma técnica desenvolvida por Tim Bell, Ian H. Witten e Mike Fellows que deu origem ao livro "*Computer Science Unplugged: An enrichment and extension programme for primary-aged children*"⁴⁸. A ideia central da Computação Desplugada é "ensinar os fundamentos da computação de forma lúdica, sem o uso de computadores, sem distrações e detalhes técnicos em demasia" (VIEIRA, PASSOS e BARRETO, 2013).

⁴³ <https://mttr.net/>

⁴⁴ <https://www.hackster.io/arduino>

⁴⁵ <http://www.facomarduino.info/>

⁴⁶ <http://labdegaragem.com/>

⁴⁷ <http://csunplugged.org/>

⁴⁸ <https://ir.canterbury.ac.nz/handle/10092/247>

As atividades são desenvolvidas com base em materiais usados pelos alunos no dia a dia, como lápis de cor, cartolinas, utensílios em geral, entre outros. Esses materiais são empregados em atividades físicas nas quais os alunos se movem e gesticulam buscando incorporar as operações realizadas por um computador (PAUL, 2015).

A técnica tem se tornado cada vez mais popular entre professores de ciências da computação em diversos países, já tendo sido traduzida em doze línguas (VIEIRA, PASSOS e BARRETO, 2013; BELL *et al*, 2009). Outro ponto que reforça a grande aceitação da técnica é a inclusão de muitas de suas atividades no currículo para o ensino de computação nos Ensinos Fundamental e Médio (k-12) da ACM (*Association for Computing Machinery*)⁴⁹ (TAUB, ARMONI e BEN-ARI, 2012).

As atividades propostas pela *CS unplugged* são mais voltadas ao entendimento de como o computador funciona, de como determinados programas funcionam, e não ao aprendizado do desenvolvimento de programas de computador, abordando temas como números binários, detecção de erros, algoritmos de busca e ordenação, roteamento em redes e autômatos finitos. No entanto, alguns trabalhos têm feito adaptações da técnica, propondo novas atividades para o aprendizado de conceitos de ciências da computação, tanto contemplados quanto não contemplados pela obra.

Como exemplos destes trabalhos podemos citar as pesquisas de Vieira, Passos e Barreto (2013), Scaico e outros (2012) e Setzer (2005). Vieira, Passos e Barreto (2013) apresentam uma proposta do uso de peças teatrais para ensinar alguns princípios de computação trabalhados no livro *CS unplugged*. Scaico et e outros (2012) substituem o uso de "cartões com pontinhos" proposto por Bell e outros (2009) para o ensino de números binários, por um conjunto de lâmpadas que acendem e apagam. Já Setzer (2005) no seu trabalho "O Computador a Papel: Uma Atividade Pedagógica para a Introdução de Conceitos Básicos de Computadores", anterior ao trabalho de BELL e outros, usa o conceito de computação sem computador para ensinar aos alunos noções básicas da estrutura interna de um computador por meio de sua linguagem de máquina. Para tanto, os alunos simulam, por meio de um teatrinho por eles encenado, todas as unidades de um computador hipotético simplificado. Do mesmo modo, este trabalho não adota as atividades propostas no

⁴⁹ Uma organização internacional científica e educacional dedicada ao avanço das artes, ciências e aplicações das tecnologias da informação.

livro *CS Unplugged*, mas, utiliza o conceito de computação desplugada para o desenho de atividades de apoio ao aprendizado de algumas estruturas básicas utilizadas na lógica de programação.

A realização de atividades de aprendizado de programação baseadas na técnica de computação desplugada pode ser caracterizado como um método de aprendizagem lúdica por meio de brincadeiras, um recurso pedagógico que, ao proporcionar prazer e alegria à criança,

[...] colabora com o processo educacional porque coloca o aluno em uma situação de boa receptividade; ele está fazendo algo que gosta, se dispersa menos e concentra-se para aproveitar ao máximo estes momentos. O aluno passa a ver o adulto (professor) de uma forma mais próxima, não é o adulto que espera dele um comportamento sério que o faça compreender as 'coisas difíceis' que eles estão ensinando. Mas é o adulto que de forma leve e alegre entra "no mundo da criança" para transmitir aquilo que sabe (DOHME, 2003, p.114).

[...] [Uma vez que se entende] a natureza viva da criança: sua predisposição ao movimento, ao riso, à fantasia, à espontaneidade [...]as atividades lúdicas não aparecem simplesmente como algo que vai agradar às crianças, mas como algo que vai aumentar a sua motivação em participar, conseqüentemente, que vai aumentar a sua capacidade de assimilação, o seu aproveitamento (DOHME, 2003, p.120).

2.9 USO DE NARRATIVAS NO PROCESSO ENSINO-APRENDIZAGEM

Diferentes autores interpretam os termos narrativa e história (*story*) de maneiras distintas. Walter Fisher (1987) define narrativa como “qualquer relato verbal ou não verbal que possui uma sequência de eventos para os quais os ouvintes atribuem um significado” (BUTCHER, 2006, p.199, tradução nossa). Brooks (1997) define história (*story*) como sendo uma descrição geral ou abstrata de uma coleção de eventos, pessoas e/ou coisas, entre os quais é estabelecida uma relação causal e temporal. Para Szurmak e Thuna (2013) uma história (*story*) distingue-se de uma narrativa pelo fato da primeira apresentar uma estrutura linear “quadro a quadro”, enquanto que a narrativa permite que se foque em determinados detalhes ao se fazer um *zoom* em eventos específicos, o que possibilita ao espectador entrar e sair da “linha de tempo da história”.

No âmbito deste trabalho, o termo narrativa pode ser entendido como uma combinação dessas definições: Um relato verbal ficcional apresentado ora de forma linear - por meio de uma sequência de eventos que propiciem uma visão mais geral

da situação sendo relatada; ora de forma não linear - focando em determinados detalhes dos acontecimentos, cenários e personagens envolvidos; de modo a possibilitar ao expectador uma atribuição de significado mais consciente a esses elementos e um maior aprofundamento no entendimento das relações causais e temporais existentes entre eles.

As narrativas constituem um modo de entendimento de novas experiências, funcionando como um ponto de referência a partir do qual pode-se criar uma imagem mental das mesmas (CONNELLY e CLANDININ, 1990 *apud* BUTCHER, 2006). No âmbito da educação, o uso de narrativas cria situações de aprendizado e desenvolve formas criativas de se resolver problemas (BUTCHER, 2006), configurando-se como uma ferramenta poderosa e versátil de ativação do aprendizado e engajamento dos alunos (SZURMAK e THUNA, 2013). Dentre os benefícios relacionados ao uso de narrativas no ensino, possuem especial relevância:

- Narrativas tornam algo abstrato mais concreto/direto.
 - Narrativas contextualizam as informações à medida em que criam um arcabouço no qual os alunos podem acomodar os novos conhecimentos (e assim ampliar sua retenção e entendimento).
 - Narrativas possibilitam aos estudantes terem experiências emocionais mais imediatas com as quais eles podem se relacionar (e, conseqüentemente, lembrar)
- (SZURMAK e THUNA, 2013, p.550, tradução nossa).

No contexto do aprendizado de ciência, tecnologia, engenharia e matemática (STEM) Subramaniam e outros (2012) apresentam uma proposta do uso de narrativas para atrair jovens, oferecendo várias atividades que possibilitem a eles imaginarem-se no papel de cientistas. O aprendizado se dá a medida que, por meio da interação entre imaginação e fatos científicos, os alunos tornam-se capazes de embasar cientificamente o desenho dos seus personagens e as ações por eles realizadas nas histórias.

Mais especificamente no que se refere à Robótica Educacional, Rusk e outros (2008) veem no uso de narrativas uma maneira de se atrair mais crianças para esse tipo de atividade. De acordo com esses autores, pesquisadores do Harvard's Project Zero⁵⁰ (SHOTWELL, WOLF e GARDNER, 1979) identificaram dois "tipos" de crianças conforme o modo como elas interagem com seus brinquedos: Crianças *Patterners* (modeladoras/configuradoras) e *Dramatists* (dramaturgas). As crianças *Patterners*

⁵⁰ <http://www.pz.harvard.edu/>

são aquelas que tem mais interesse em investigar o funcionamento dos objetos, preferindo brincar com blocos e quebra-cabeças. Já as crianças *Dramatistis* preferem brincar com bonecos e bichinhos de brinquedo, criando histórias de faz de conta e interagindo com os colegas. Normalmente, as atividades relacionadas à Robótica Educacional tendem a ser mais atraentes para crianças *Patternner*. No entanto, o uso de narrativas pode tornar esse tipo de atividade atraente também para crianças *Dramatisitis* (RUSK *et al*, 2008).

Os benefícios trazidos pelo uso de narrativas no processo ensino-aprendizagem, tanto num sentido mais amplo, quanto de maneira mais específica no ensino de STEM e Robótica Educacional (conforme pode-se observar nos trabalhos aqui citados) apontam o uso de narrativas como sendo uma estratégia bastante interessante a ser adotada no desenho das atividades a serem utilizadas para o ensino de programação com o DB4K.

2.10 TRABALHOS CORRELATOS

Como já mencionado, não foi encontrado na literatura um Ambiente de Programação Visual voltado à programação de placas de prototipagem eletrônica Arduino (tecnologia adotada neste trabalho), pensado e desenvolvido especificamente para o ensino de programação a crianças do Ensino Fundamental I. Ou seja, um Ambiente de Programação Visual composto por blocos graficamente mais atrativos e dotados de uma semântica menos abstrata que a empregada nos ambientes de programação que trabalham com o mesmo hardware.

Como exemplos de Ambientes de Programação Visual para placas Arduino atualmente existentes, podemos citar o DuinoBlocks (ALVES, 2013), que serviu de inspiração para o desenvolvimento do DB4K; O S4A - Scratch for Arduino ⁵¹, um *fork*⁵² do Scratch que permite a programação de placas Arduino; e o Ardublockly⁵³, ambiente sobre o qual foi desenvolvido o DuinoBlocks4Kids (ver capítulo 3.2).

⁵¹ <http://s4a.cat/>

⁵² *Forks* são softwares desenvolvidos a partir do código fonte de outros softwares, dando origem a um novo projeto da mesma “linha”, mas, independente do projeto do qual se originou.

⁵³ <https://github.com/carlosperate/ardublockly>

Algumas características adotadas pelo DuinoBlocks4Kids, não presentes nos ambientes acima listados, e que possuem como objetivo tornar o seu uso mais acessível e aprazível a crianças do Ensino Fundamental I são:

- Blocos de programação desenhados de modo a apresentarem uma semântica diretamente relacionada com os dispositivos sendo manipulados e com os efeitos por eles causados sobre estes dispositivos.
- Supressão de detalhes relacionados ao hardware, como pinagens e valores de níveis de tensão.
- Uso de linguagem icônica.
- Conjunto " enxuto " de blocos.
- Simplificação dos valores dos parâmetros utilizados nos blocos, como por exemplo: temperatura: alta ou baixa; luz: muita ou pouca; velocidade: rápida, média ou devagar.

Uma vez não tendo sido encontrados trabalhos que apresentassem um Ambiente de Programação Visual em Blocos como essas características voltado à programação de placas de prototipagem eletrônica Arduino, não foram também encontrados estudos acerca do aprendizado de programação de computadores do Ensino Fundamental I com base nessa associação de tecnologias.

3 O DUINOBLOCKS4KIDS (DB4K)

Como mencionado no início deste trabalho, o desenvolvimento do DuinoBlocks4Kids surgiu do desejo de se estender o alcance dos trabalhos desenvolvidos pelo LIVRE/LabVad a crianças do Ensino Fundamental I.

O LabVad - Laboratório Virtual de Atividades Didáticas em Ciências e Robótica⁵⁴, tem como objetivo principal, a partir de pesquisas acerca da utilização das TIC na educação, fomentar práticas pedagógicas que tirem proveito da ideia de "aprender fazendo" a partir do desenvolvimento de atividades experimentais (PINTO, 2011). Para auxiliar na viabilização de atividades desse tipo, tão importantes na Educação Básica, sobretudo nas disciplinas de ciências naturais (Física, Química e Biologia), o LabVad adotou, como estratégia norteadora, a utilização de laboratórios remotos, ou seja, experimentos reais sendo executados a distância por meio de computadores ligados à internet (PINTO, 2011). Uma das vantagens desse tipo de solução é possibilitar a execução de experimentos reais sem que para isso seja necessária a aquisição de equipamentos especiais ou alocação de espaço para a sua realização.

Dentre outros experimentos, o ambiente LabVad possui um laboratório de Robótica Educacional denominado LabVad Robótica (SOUZA *et al*, 2014), que busca fornecer ao usuário uma experiência de programação para robótica muito próxima da que ele teria se estivesse realizando um experimento localmente.

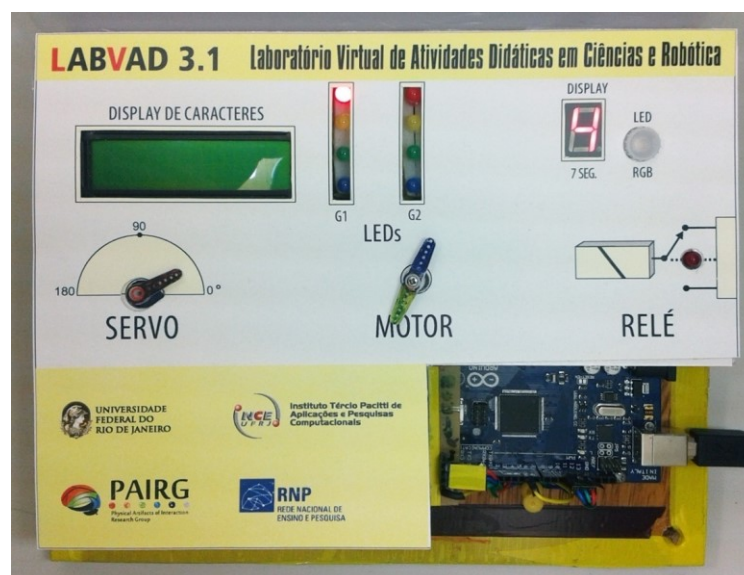


Figura 1: Imagem vista pelo usuário por meio do vídeo apresentado pelo LabVad Robótica

⁵⁴ <http://labvad.nce.ufrj.br>

A Figura 1 apresenta a imagem vista pelo usuário durante a realização dos experimentos no LabVad Robótica, por meio de um vídeo em tempo real apresentado na interface da plataforma LabVad, e a Figura 2 detalha os dispositivos presente neste laboratório, sendo eles: um conjunto de LEDs⁵⁵, um LED RGB⁵⁶, um display de 7 segmentos⁵⁷, um display LCD⁵⁸, um servo motor⁵⁹, um motor DC⁶⁰ e um relé⁶¹.

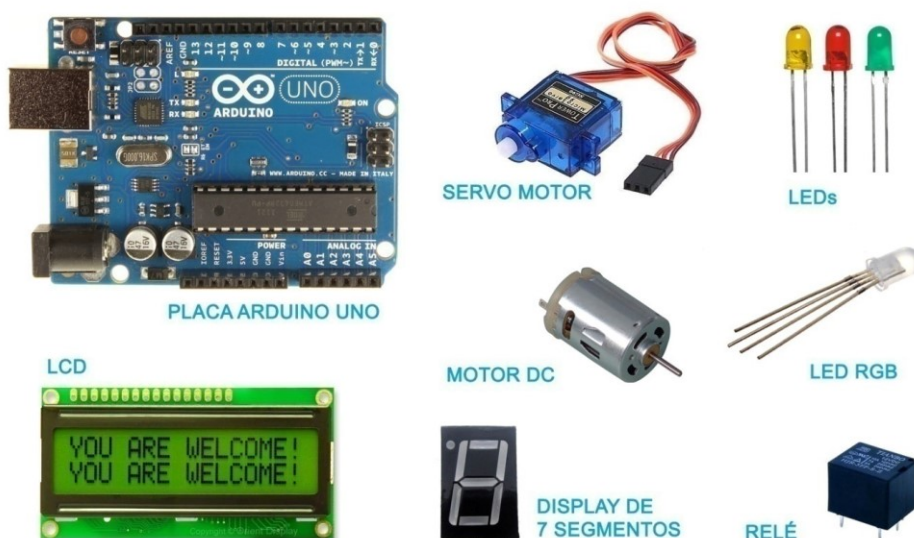


Figura 2: Placa Arduino Uno e dispositivos controlados pelo experimento LabVad Robótica

O DB4K foi projetado de modo a permitir a sua integração com o LabVad, tanto no que se refere às tecnologias utilizadas quanto aos experimentos a serem manipulados por meio do ambiente.

Além da utilização do DB4K para o controle, via internet, dos experimentos disponibilizados pelo LabVad Robótica, buscou-se possibilitar também a sua utilização

⁵⁵ O LED é um diodo emissor de luz (*Light Emitter Diode*), um componente eletrônico que tem a propriedade de transformar energia elétrica em luz.

⁵⁶ LED RGB é um tipo especial de LED que pode emitir três cores diferentes de luz, vermelha (*Red*), verde (*Green*) e azul (*Blue*), individualmente ou combinadas, podendo desta forma gerar uma variedade de possibilidades de cor.

⁵⁷ Display de 7 segmentos é um tipo de display usado para exibir uma informação alfanumérica (letras e números).

⁵⁸ Display LCD, ou display de cristal líquido (*Liquid Crystal Display*), é um painel eletrônico utilizado para a apresentação de textos, imagens e vídeos. O LCD utilizado pelo LabVad exibe apenas textos.

⁵⁹ Servo motor é um dispositivo eletromecânico que apresenta um movimento proporcional a um “comando”. Ao receber um “comando” ele modifica a sua posição movendo-se para a posição desejada.

⁶⁰ Motor DC, ou motor de corrente contínua (*Direct Current Motor*), é um dispositivo elétrico que converte corrente elétrica contínua (energia elétrica) em movimento (energia mecânica).

⁶¹ Relé é um dispositivo eletromecânico utilizado para ligar e desligar outros dispositivos. É uma espécie de interruptor controlado eletronicamente.

em uma versão *client-side*, ou seja, que pudesse ser executada no computador do usuário, sem a necessidade de conexão com a internet, e capaz de controlar, diretamente em uma placa Arduino conectada ao computador, outros experimentos além daqueles disponibilizados pelo LabVad. Este trabalho trata especificamente dessa versão do ambiente DB4K.

3.1 ESCOLHA DOS MATERIAIS DE ROBÓTICA A SEREM MANIPULADOS PELO AMBIENTE

O Ambiente DB4K *client-side* permite ao usuário, via programação, a manipulação de: um conjunto de LEDs, um LED RGB, um display de 7 segmentos, um display LCD, um servo motor e um motor DC, todos também programáveis pela interface do DB4K presente no LabVad. Além desses, o DB4K *client-side*, possibilita também a programação de um conjunto de sensores, de um carrinho robô e de um braço robótico (Figura 3), o que vem a possibilitar o desenvolvimento de uma gama bastante variada de atividades relacionadas ao ensino de conceitos básicos de programação.

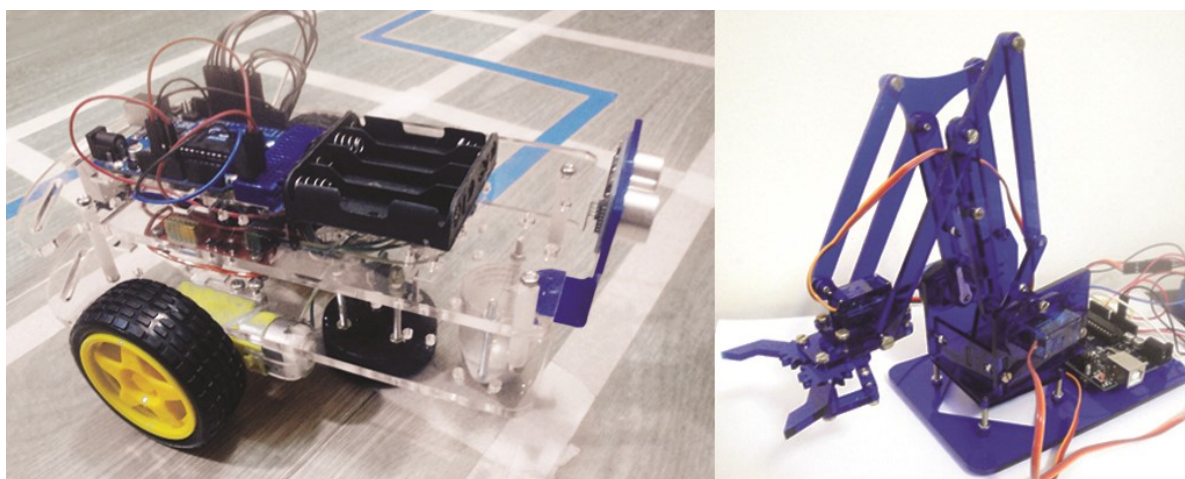


Figura 3: Carrinho e braço robóticos programáveis via DB4K *client-side*⁶²

Para manter-se a compatibilidade com o LabVad Robótica e para tornar possível ao usuário o controle dos dispositivos a serem manipulados pelo DB4K, sem que para isso seja necessária a montagem de alguns circuitos relativamente

⁶² Tanto o carrinho quanto o braço robótico adotados pelo projeto são facilmente encontrados no mercado a preços bastante acessíveis

complexos necessários à conexão conjunta dos atuadores e sensores à placa Arduino, optou-se pelo desenvolvimento de um artefato batizado como "a caixinha mágica" (Figura 4). Trata-se de um pote de material plástico no qual encontram-se acoplados um conjunto de atuadores e sensores conectados a uma placa de prototipagem eletrônica Arduino. O pote é transparente e pode ser facilmente aberto, permitindo que o usuário observe como é feita a conexão dos dispositivos à placa controladora, podendo todos os seus elementos serem afixados e desafixados sempre que desejável.



Figura 4: Caixinha mágica⁶³

A caixinha mágica possui:

- 1 display LCD;
- 1 display de 7 segmentos;
- 4 LEDs;
- 1 LED RGB;
- 1 buzzer;
- 1 motor DC;
- 1 servo motor;
- 1 sensor de distância;
- 1 sensor de luz;
- 1 sensor de temperatura;
- 1 placa Arduino mega;
- 1 cabo USB para conexão da placa ao computador.

⁶³ A caixinha foi inspirada em uma "maleta" desenvolvida pelo prof. Serafim Brandão Pinto (NCE/UFRJ).

Além da caixinha mágica, foram desenvolvidos também alguns materiais a serem utilizados nas atividades com narrativas (ver capítulo 4.2), como um protótipo de peixe robô, um morcego robô (Figura 5) e um cachorro robô (Figura 6). Todos esses materiais podem ser facilmente abertos de modo que as crianças possam observar os dispositivos utilizados em sua construção, bem como a conexão dos mesmos com a placa Arduino. Alguns dos materiais, como no caso do morcego robô, possuem partes que são entregues já montadas pelo professor e outras que são construídas e montadas pelos alunos.

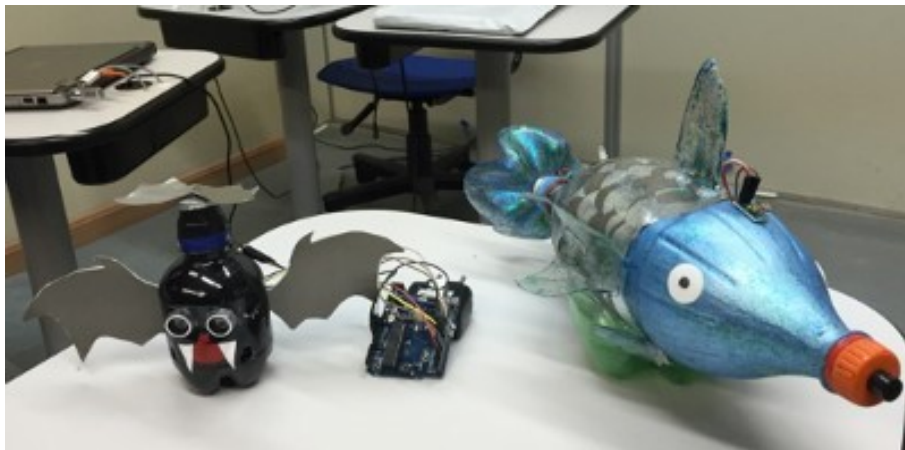


Figura 5: O morcego robô⁶⁴ e o peixe robô de garrafa pet

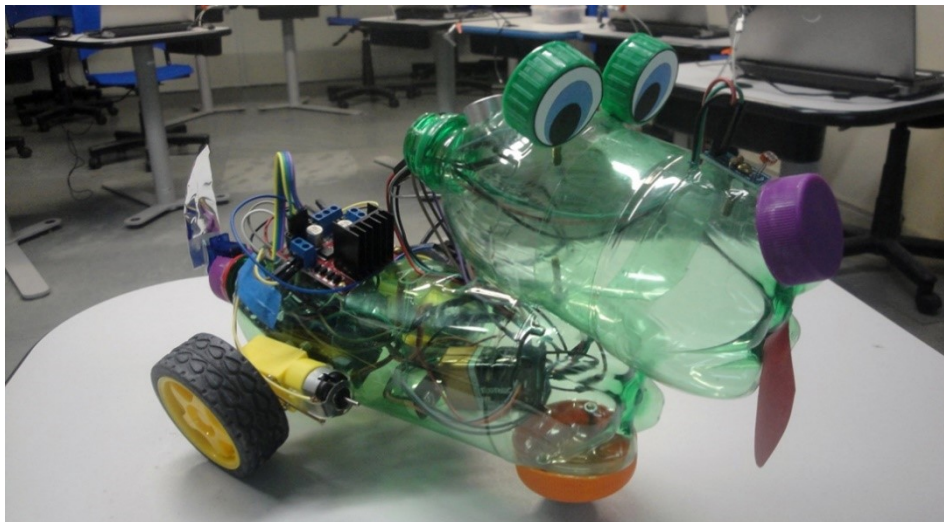


Figura 6: Cachorro robô de garrafa pet⁶⁵

⁶⁴ Design inspirado no morcego de garrafa pet do canal Planeta Viável: <https://www.youtube.com/watch?v=FhjedXHY4Z4>

⁶⁵ Desing inspirado no cachorro de garrafa pet do blog da “prô francieli”: <http://profrancieli.blogspot.com.br/2010/11/cachorro-de-garrafa-pet.html>

Foi adotado também um Kit Arduino Básico (Figura 7) para a montagem de pequenos circuitos, este kit é composto por:

- 1 conjunto de LEDs;
- 1 buzzer;
- 1 motor DC;
- 1 servo motor;
- 1 placa Arduino Uno;
- 1 sensor de distância;
- 1 cabo USB para conexão da placa ao computador;
- 1 conjunto de jumpers.



Figura 7: Kit Arduino

3.2 A ESCOLHA DAS FERRAMENTAS PARA CONSTRUÇÃO DO AMBIENTE

O desenvolvimento do Ambiente de Programação Visual em Blocos DB4K teve como inspiração inicial o DuinoBlocks (ALVES, 2013), um Ambiente de Programação em Blocos desenvolvido pelo LIVRE, desenhado especificamente para controle de plataformas de prototipagem eletrônica Arduino e inteiramente baseado em tecnologia WEB e Software Livre.

Desse modo, a opção inicial para o desenvolvimento do DB4k foi a realização de um *fork*⁶⁶ do DuinoBlocks. No entanto, dificuldades encontradas no entendimento

⁶⁶ *Forks* são softwares desenvolvidos a partir do código fonte de outros softwares, dando origem a um novo projeto da mesma “linha”, mas, independente do projeto que lhe deu origem.

da lógica utilizada para a criação dos blocos do DuinoBlocks, bem como para a geração de código *Wiring*⁶⁷ a partir destes blocos, aliadas ao fato do DuinoBlocks não ter sido projetado para permitir a utilização de ícones como um elemento gráfico de seus blocos, apontaram uma possível inviabilização do desenvolvimento do DB4K dentro do prazo pretendido. Diante deste quadro, partiu-se em busca de ferramentas alternativas que pudessem viabilizar o projeto, tendo-se sempre em vista os principais requisitos do ambiente:

- Ser baseado em Tecnologia Livre.
- Usar tecnologia WEB (para viabilizar sua integração com a plataforma LabVad).

Uma possível opção encontrada foi o S4A – Scratch for Arduino⁶⁸. O S4A é uma modificação do Scratch⁶⁹ que permite a programação de placas de prototipagem eletrônica Arduino a partir de blocos desenhados especificamente para esse fim. Além de se tratar de uma extensão do Scratch, o S4A possibilita o envio direto dos programas nele desenvolvidos para a placa Arduino, funcionalidade esta não presente na versão atual do DuinoBlocks. Essas duas características fizeram do S4A um forte candidato a ser utilizado como base para desenvolvimento do DB4K.

No entanto, o S4A utiliza, em seu código fonte, a linguagem Squeak⁷⁰, um ambiente de programação de código aberto desenvolvido com base na linguagem SmallTalk⁷¹. O Squeak não é uma linguagem voltada ao desenvolvimento de aplicações WEB, o que dificultaria uma possível integração do DB4K com o LabVad. Além disso, seria necessário despende um tempo razoável para a aprendizagem da linguagem Squeak/Smalltalk de modo a possibilitar as adaptações necessárias dos blocos Arduino do S4A, bem como outras questões que só poderiam ser verificadas depois de um estudo mais aprofundado do ambiente. Diante deste cenário que, mais uma vez apontava para uma possível inviabilização da execução do projeto dentro do prazo pretendido, partiu-se em busca de novas alternativas.

⁶⁷ Linguagem nativa para a programação das placas de prototipagem eletrônica Arduino, derivada da linguagem C/C++.

⁶⁸ <http://s4a.cat/>

⁶⁹ <https://scratch.mit.edu/>

⁷⁰ <http://squeak.org/>

⁷¹ <http://smalltalk.gnu.org/>

Chegou-se então, após algumas pesquisas, à Blockly⁷², uma biblioteca de código aberto desenvolvida pela Google Developers⁷³, baseada em tecnologia WEB, e voltada à construção de ambientes de Programação Visual. Além de ser uma biblioteca pensada exatamente para servir de base ao desenvolvimento de ambientes de Programação Visual em Blocos, e de suportar os requisitos fundamentais do ambiente DuinoBlocks4Kids, quais sejam: ser baseada em Tecnologia Livre e usar tecnologia WEB, a biblioteca Blockly possui uma série de características que atendiam a outros requisitos importantes do DB4K:

- Possibilita a utilização de imagens nos blocos.
- Permite a geração e exibição de código textual a partir dos blocos.
- Os ambientes criados com a biblioteca Blockly podem ser utilizados tanto em um servidor WEB quanto no lado cliente (ou seja, *off-line*).
- Possibilita a inserção de comentários no código a partir dos blocos.
- É compatível com os navegadores Chrome, Firefox, Safari, Opera e Internet Explorer, e também com diversos sistemas operacionais, inclusive Android e IOS, o que permite a sua utilização em uma diversidade de dispositivos móveis.

A biblioteca Blockly possui também uma extensa documentação, tutoriais para o uso de todas as suas funcionalidades, uma biblioteca de exemplos e um grupo de discussão bastante ativo, o que facilita muito a obtenção de suporte por parte do desenvolvedor, e foi por meio desse grupo de discussão que se chegou ao Ardublockly⁷⁴, um Ambiente de Programação em Blocos para Arduino desenvolvido a partir da biblioteca Blockly. O Ardublockly veio somar às vantagens já existentes no uso da Blockly, os seguintes atributos:

- A utilização, em sua interface, da biblioteca Materialize⁷⁵, uma biblioteca de estilos CSS, também disponibilizada pela Google Developers, que dá suporte à criação de interfaces responsivas⁷⁶ de grande impacto visual.

⁷² <https://developers.google.com/blockly/>

⁷³ <https://developers.google.com/>

⁷⁴ <https://github.com/carlosperate/ardublockly>

⁷⁵ <http://materializecss.com/>

⁷⁶ Interfaces responsivas são aquelas que adéquam o seu layout ao dispositivo a partir da qual estão sendo acessadas (PCs, Tabletes, Smartphones, etc.), ou ao tamanho da “janela” na qual estão sendo apresentadas. Esse tipo de interface ajusta o seu conteúdo de maneira a melhor apresentá-lo em cada tamanho de tela, melhorando assim sua usabilidade.

- Funções de apoio à geração de código *Wiring*⁷⁷.
- O envio dos programas nele criados diretamente para uma placa Arduino conectada ao computador (funcionalidade presente apenas na versão *client-side*).

Esse amplo conjunto de características positivas fizeram do Ardublockly a opção definitiva de ambiente sobre o qual se desenvolver o DB4K.

3.3 O AMBIENTE

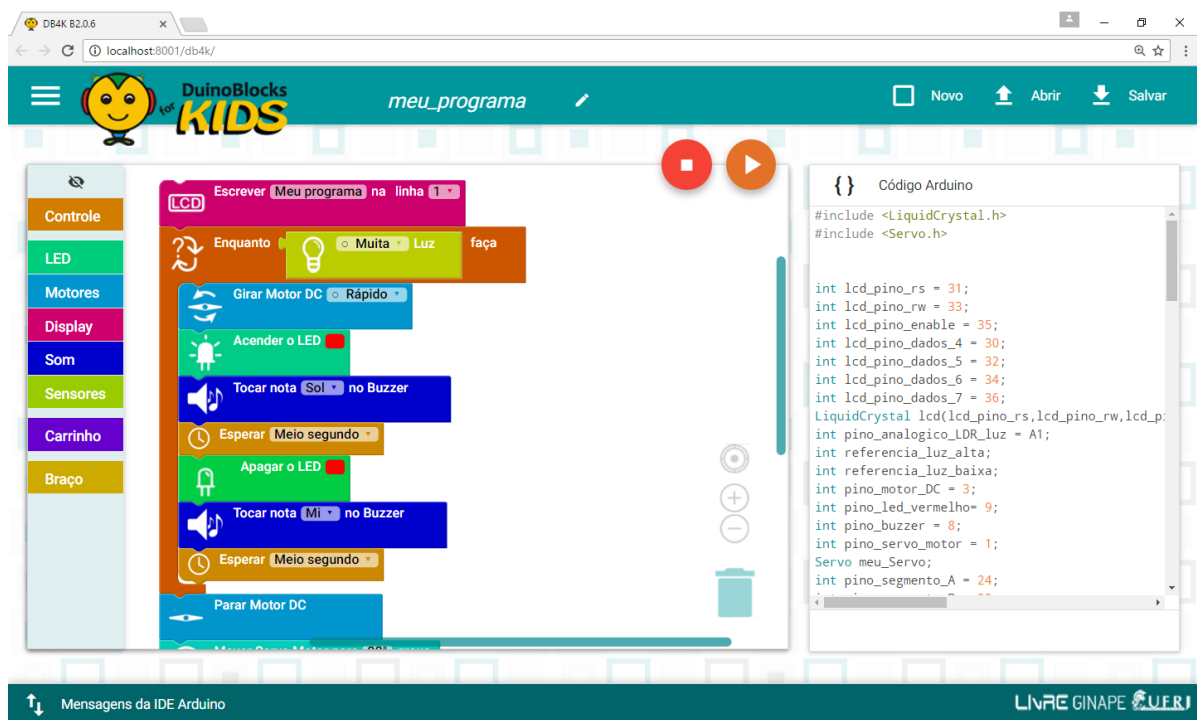


Figura 8: Visão Geral da Interface do DB4k *client-side*

O DuinoBlocks4Kids aproveitou muitos dos elementos da interface do Ardublockly (embora algumas funcionalidades tenham sido incluídas ou modificadas). Já os blocos utilizados para a construção dos programas, elemento central do DB4K, foram totalmente redesenhados e programados. A Figura 8 apresenta uma visão geral da interface do DB4K, e o diagrama de classes presente na Figura 9 fornece, de maneira simplificada, uma visão geral dos elementos de interface com o usuário presentes no ambiente.

⁷⁷ Linguagem nativa para a programação das placas de prototipagem eletrônica Arduino.

Figura 9: Diagrama de Classes simplificado da interface do DB4K

Os blocos do DB4K foram construídos de modo a apresentarem uma semântica diretamente relacionada com os dispositivos sendo manipulados e com os efeitos por eles causados sobre estes dispositivos. Por exemplo, para acender um LED, desenhou-se o bloco *acender LED*; para escrever no display de LCD, desenhou-se o bloco *escrever < texto > na linha < l > do LCD*. Além dos blocos diretamente relacionados com os dispositivos a serem manipulados, foram desenhados também blocos para as estruturas de controle utilizadas em programação, como repetição e decisão. As Figuras 10 a 17 apresentam todos os blocos disponíveis no DB4K.

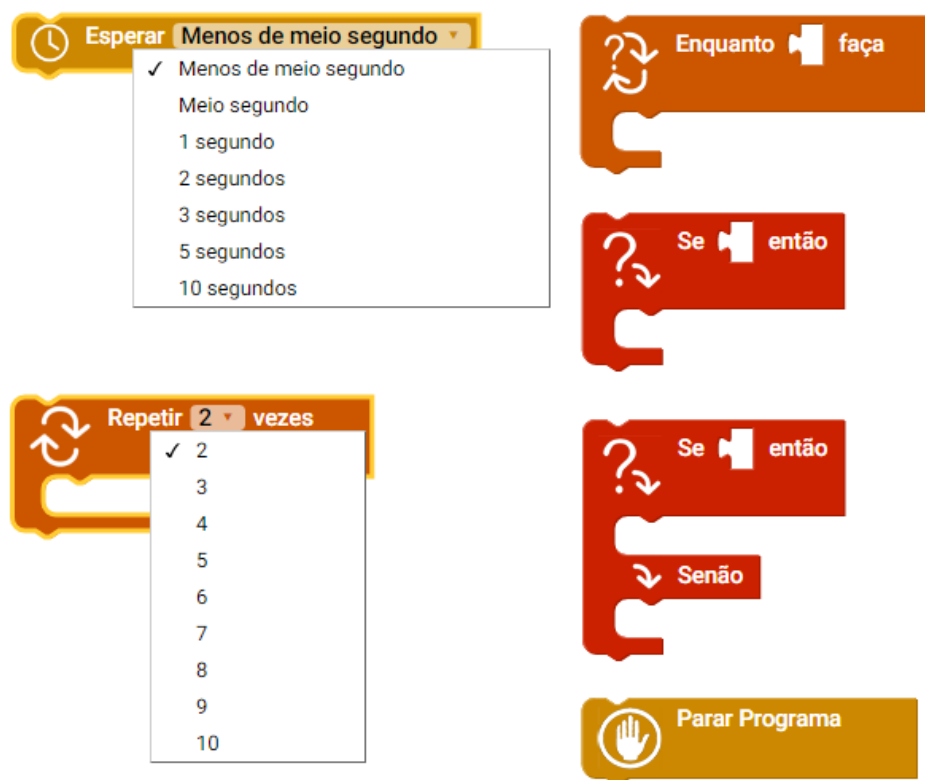


Figura 10: Blocos da categoria controle (Os blocos condicionais utilizam blocos da categoria sensores como fator condicionante (Figura 15))

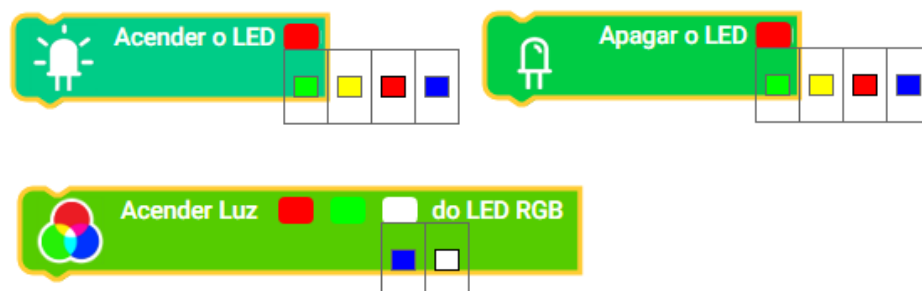


Figura 11: Blocos da categoria LED



Figura 12: Blocos da categoria Motores



Figura 13: Blocos da categoria Display

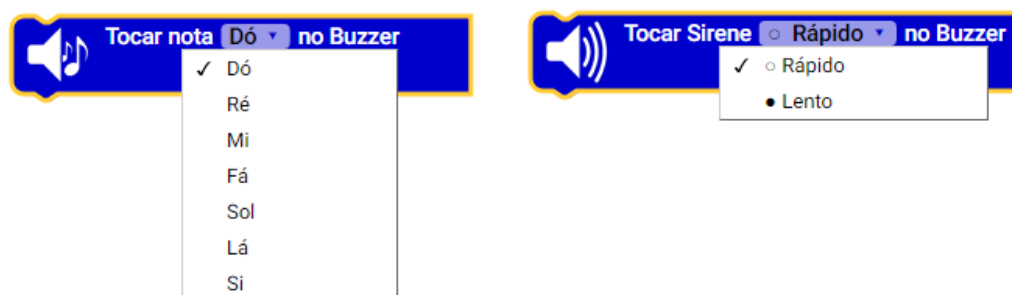


Figura 14: Blocos da categoria Som

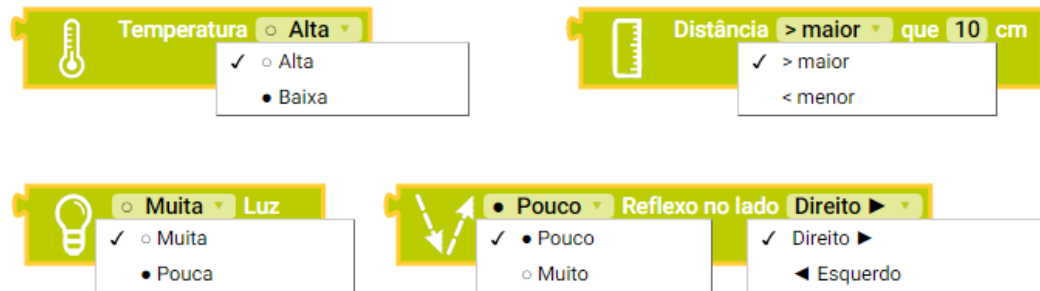


Figura 15: Blocos da categoria Sensores

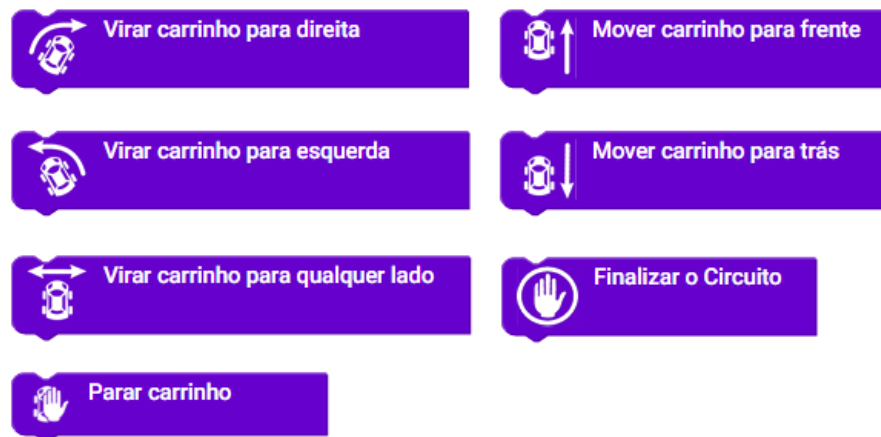


Figura 16: Blocos da categoria Carrinho



Figura 17: Blocos da categoria Braço

A exemplo do que ocorre com o ScratchJr⁷⁸ e com o Lego Mindstorms⁷⁹, optou-se por utilizar no DB4K, tanto uma representação textual quanto icônica para descrever as "funções" dos blocos⁸⁰. Em relação às cores utilizadas no ambiente, em

⁷⁸ <https://www.scratchjr.org/>

⁷⁹ <http://mindstorms.lego.com/>

⁸⁰ O que pode auxiliar o uso do ambiente por parte de usuários ainda "pouco à vontade" com a leitura e escrita.

contraposição às cores neutras utilizadas nos planos de fundo, na barra de navegação e nas barras de títulos das janelas retráteis, decidiu-se por adotar cores vivas para os blocos e seus respectivos itens de menu, direcionando a atenção do usuário exatamente para esses que são os elementos fundamentais do DB4K.

O acesso aos blocos é feito a partir do menu de blocos presente na área de trabalho (Figura 18). Cada item deste menu representa uma categoria de blocos e dá acesso a um submenu onde são disponibilizados os blocos correspondentes a esta categoria. Pode-se observar na Figura 18 como são apresentados para o usuário os blocos da categoria LED. Estão também presentes na área de trabalho um botão para enviar o programa para a placa Arduino e um botão para interromper a execução do programa na placa, uma lixeira para exclusão de blocos e um conjunto de três botões que possibilitam aplicar zoom (positivo e negativo) aos blocos.

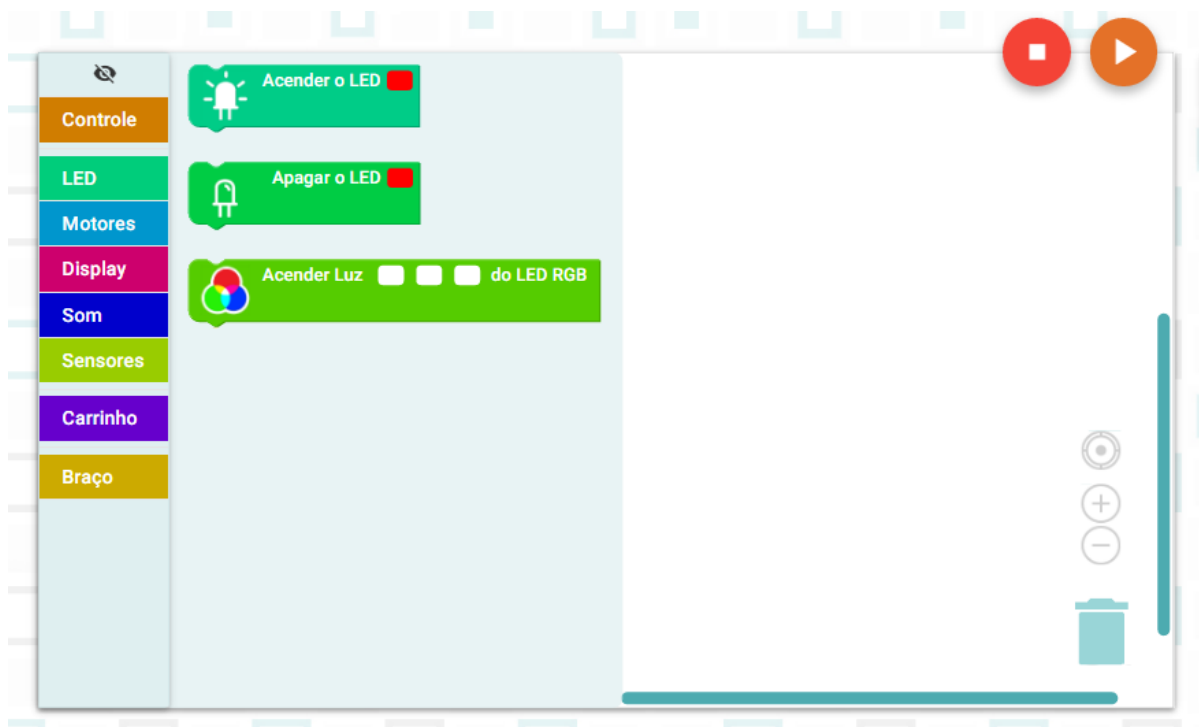


Figura 18: Área de trabalho e blocos da categoria LED

Além da área de trabalho para a construção dos programas com os blocos, para que o usuário possa, caso deseje, ter acesso à linguagem textual *Wiring*⁸¹ associada a cada bloco do DuinoBlocks4Kids, decidiu-se por disponibilizar, na interface do DB4K, uma área para a exibição do programa textual correspondente ao

⁸¹ Linguagem nativa para a programação das placas de prototipagem eletrônica Arduino, derivada da linguagem C/C++.

programa em blocos (Figura 19). Essa área também está presente na interface do Ardublocky e foi implementada por meio de uma janela retrátil (ou seja, seu conteúdo pode ser exibido ou ocultado).

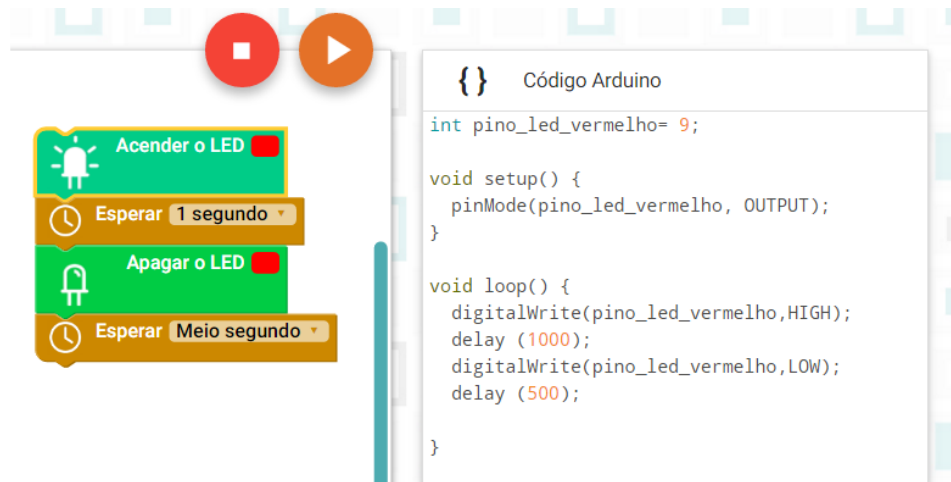


Figura 19: Área para a exibição do programa textual (*Wiring*) correspondente ao programa criado com os blocos

A interface possui também uma barra de navegação (Figura 20) onde é possível nomear, salvar, abrir e criar um novo programa, além de acessar um menu “retrátil” (Figura 21), que disponibiliza aos usuários os seguintes itens:

- Programa: dá acesso às opções que possibilitam criar um novo programa, salvar um programa ou abrir um programa salvo previamente.
- Código Arduino: Dá acesso às opções que permitem salvar o código *Wiring* correspondente ao programa em blocos criado no DB4K ou abrir o código *Wiring* no IDE do Arduino.
- Configurações: abre uma janela (Figura 22) onde é possível configurar a localização do compilador Arduino, a porta serial a ser utilizada para comunicação com a placa Arduino, o modelo de placa Arduino conectado à porta serial e a pasta de trabalho⁸².
- Atualizar Pinos: força a recarga (*reload*) dos arquivos utilizados pelo DB4K, atualizando os números dos pinos para os valores presentes no arquivo de configuração de pinos. Esses valores indicam em quais pinos

⁸² Sempre que o usuário envia um programa para execução ou solicita que o código *Wiring* seja aberto no IDE Arduino, o DB4K salva na pasta de trabalho um arquivo chamado db4kSketch.ino com o código *Wiring* correspondente ao programa em blocos construído pelo usuário. Este arquivo é a fonte do código que o sistema carrega na placa ou abre no IDE Arduino.

da placa conectada à porta serial cada dispositivo (motor, display, LED, etc.) está ligado.

- Sobre: abre uma janela com informações sobre o DB4K.
- Exemplos: dá acesso à alguns exemplos de programas simples como: piscar LED e Girar motor DC.

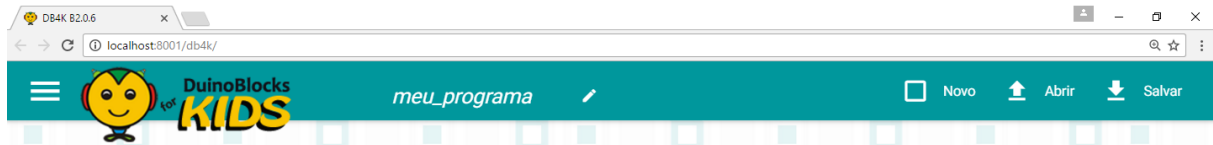


Figura 20: Barra de navegação

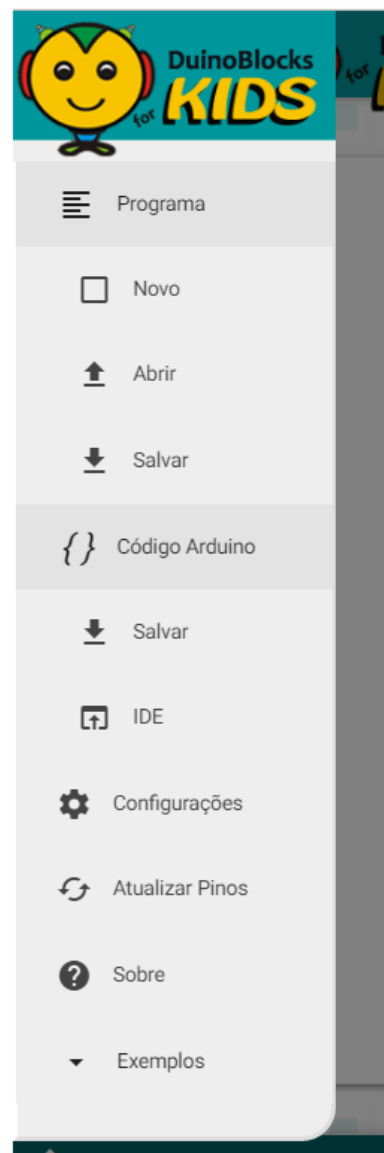


Figura 21: Menu retrátil

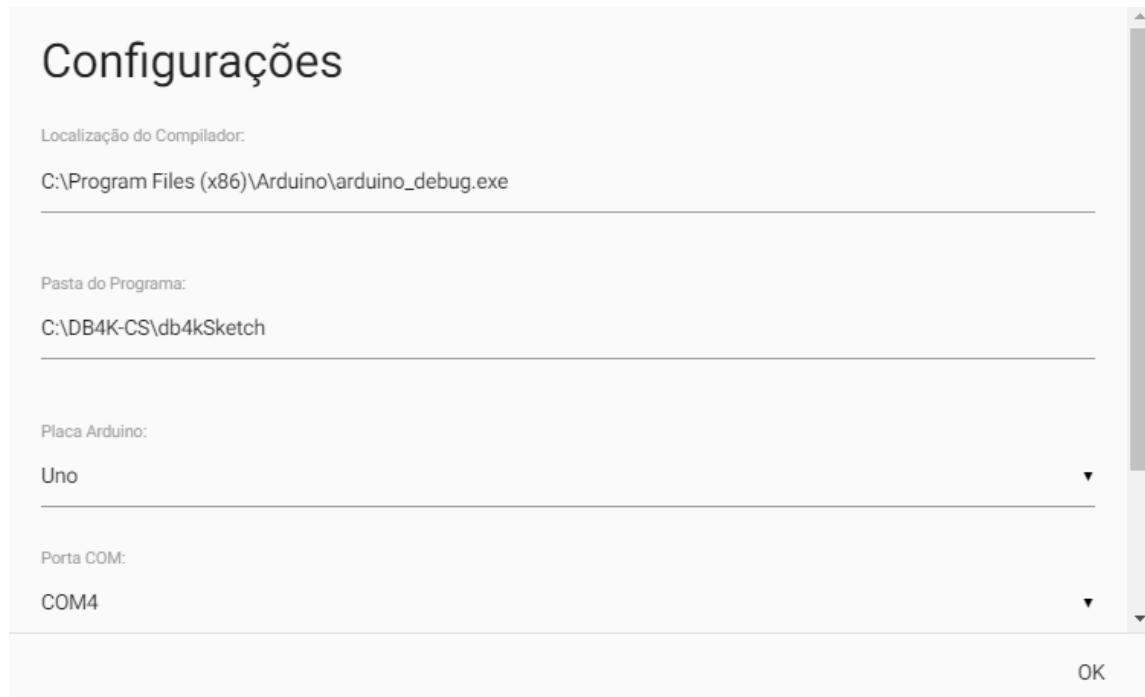


Figura 22: Janela de configurações

Na base da interface, o sistema disponibiliza uma janela retrátil (Figura 23) por meio da qual é possível visualizar mensagens enviadas pelo software Arduino durante o processo de compilação do programa criado.

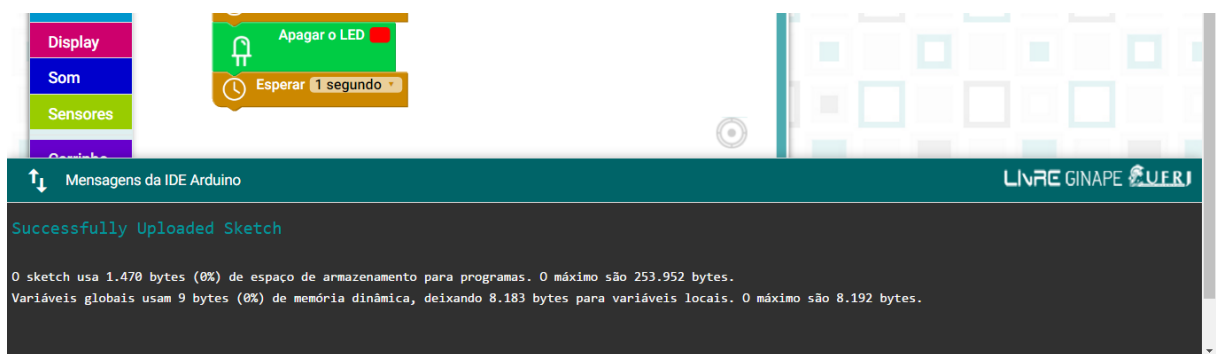


Figura 23: Janela retrátil para exibição de mensagens do compilador Arduino

O diagrama de casos de uso presente na Figura 24 mostra as principais funcionalidades do DB4K, tendo sido destacadas com a cor azul as mais comumente utilizadas pelas crianças.

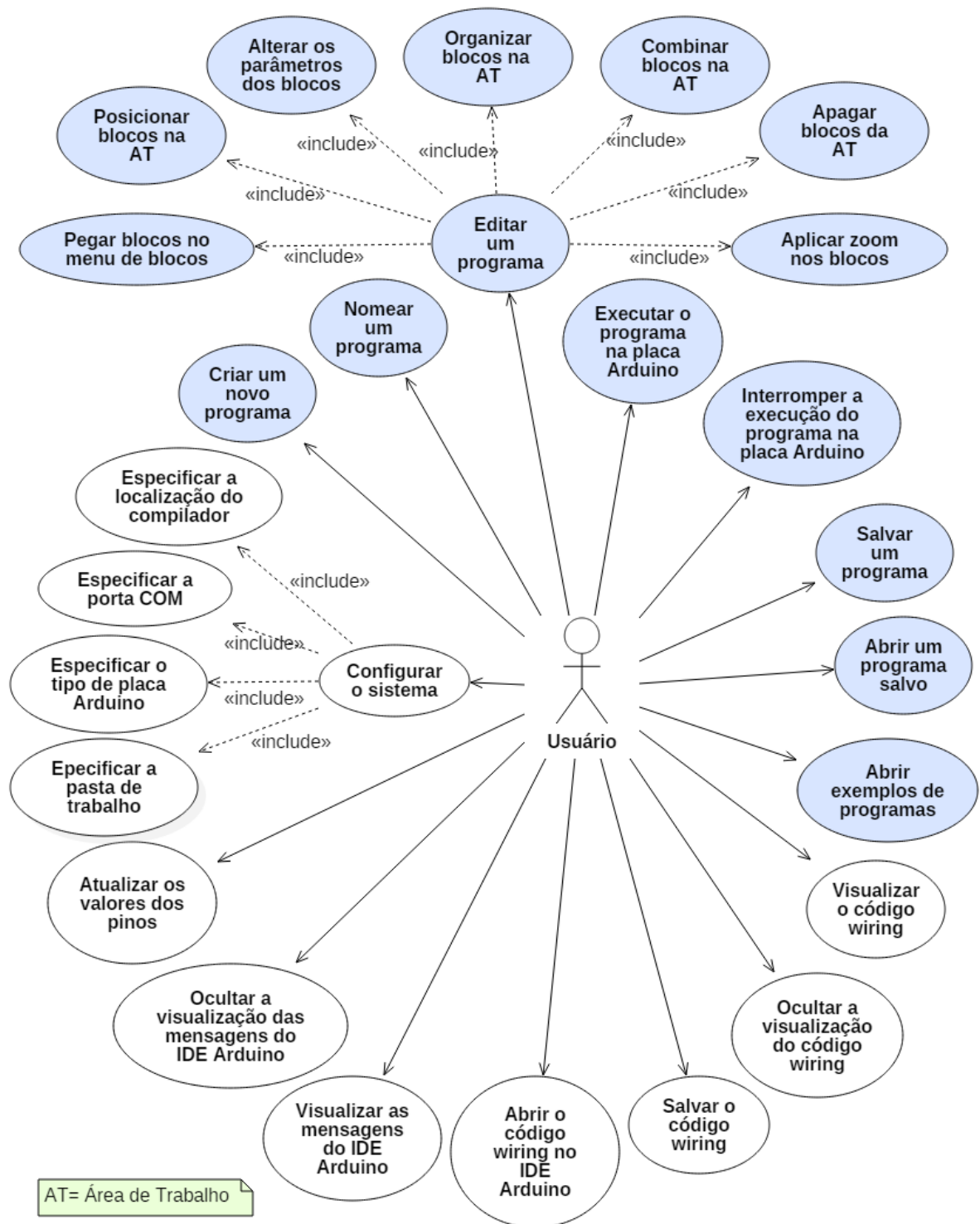


Figura 24: Diagrama de casos de uso com as principais funcionalidades do DB4K

Dentre os casos de uso destacados em azul, encontram-se as funcionalidades utilizadas na principal sequência de ações efetuada pelas crianças quando da

realização dos exercícios de programação dos materiais de robótica. Esta sequência pode ser observada no diagrama presente na Figura 25.

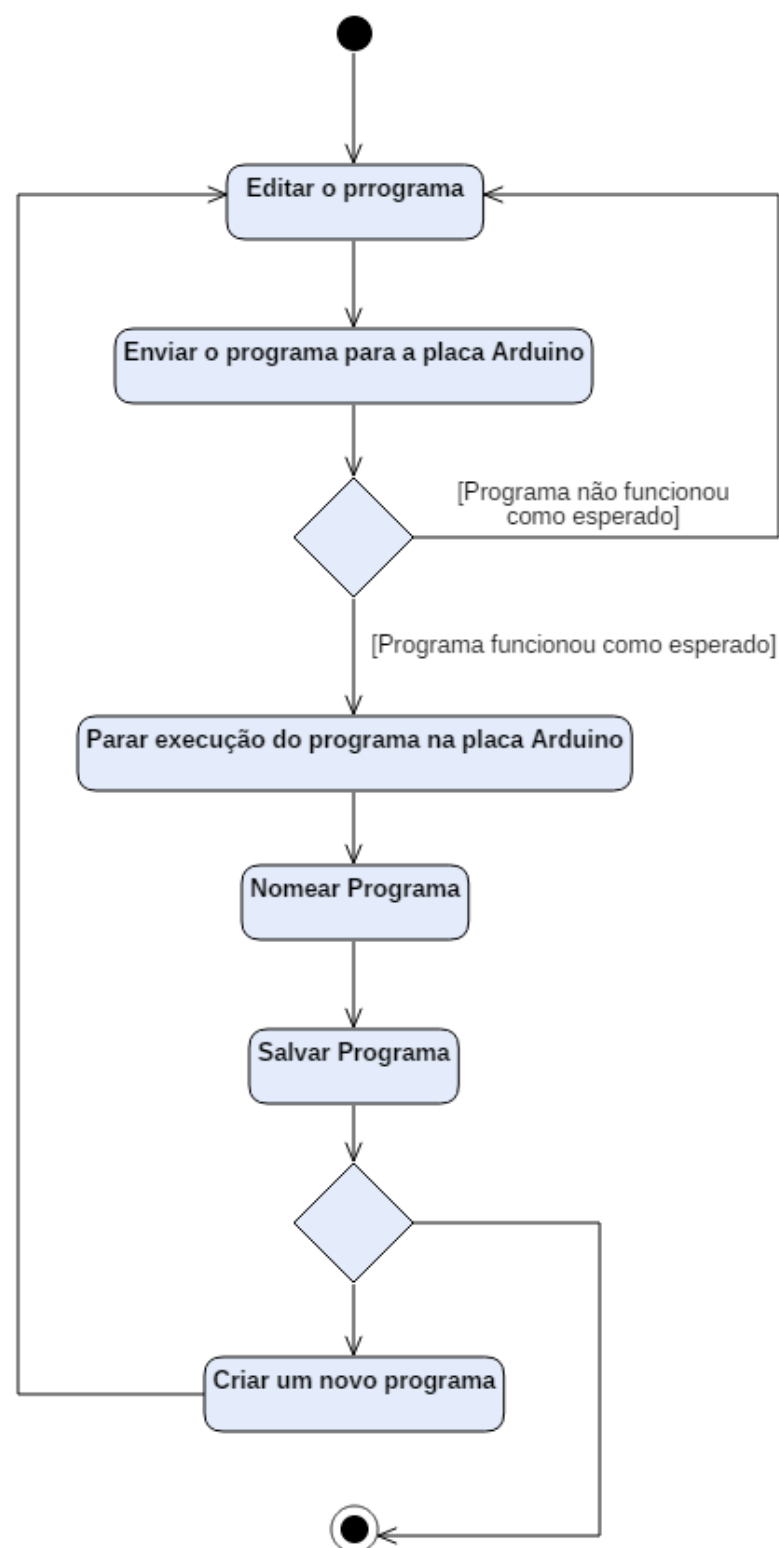


Figura 25: Diagrama da principal sequência de atividades realizada pelas crianças quando da programação dos materiais de robótica com o DB4K

4 A OFICINA

4.1 OBJETIVOS

- Habilitar os alunos a desenvolverem programas simples de computador com o emprego de um conjunto de estruturas básicas de lógica de programação.
- Possibilitar aos alunos o desenvolvimento e exercício de habilidades do Pensamento Computacional.
- Familiarizar os alunos com os dispositivos comumente utilizados na Robótica Educacional.
- Coletar, por meio das atividades propostas, dados que possibilitem a avaliação do aprendizado dos participantes e a usabilidade das ferramentas e materiais utilizados.

4.2 PROPOSTA METODOLÓGICA

O desenho da oficina foi fortemente influenciado pelos trabalhos de Bers e outros (2010) e Sullivan e Bers (2016) do Grupo de Pesquisa DevTech⁸³, do Departamento dos Estudos da Criança e do Desenvolvimento Humano da Tufts University, Massachusetts, EUA, no qual é apresentado um currículo para o ensino de programação e robótica a crianças da Educação Básica. Além do conteúdo a ser trabalhado e do modelo de avaliação de aprendizagem utilizado, foram também adotadas do curso da DevTech algumas estratégias pedagógicas, às quais foram adicionadas outras, como o uso de narrativas, configurando-se, ao final, a combinação dos seguintes recursos pedagógicos:

- Aulas expositivas dialogadas;
- Trabalhos práticos;
- Brincadeiras;
- Narrativas.

⁸³ <http://ase.tufts.edu/devtech/>

As atividades práticas referem-se ao auxílio na confecção de robôs de garrafa pet, montagem de pequenos circuitos eletrônicos e programação (com o Ambiente de Programação em Blocos DB4K) dos circuitos montados e dos materiais de robótica apresentados no capítulo 3.

As brincadeiras, inspiradas na técnica de computação desplugada (ver capítulo 2.8), visam apresentar e trabalhar, de maneira lúdica e divertida, as ideias de sequenciamento e repetições de ações, bem como de tomada de decisão (necessárias ao entendimento das estruturas de programação tratadas na oficina), conceitos esses que são posteriormente trabalhados no Ambiente de Programação em Blocos em conjunto com os materiais de robótica.

Um exemplo dessas brincadeiras é a atividade do robô humano (MICHALAKIS, 2013) na qual as crianças, em grupos, recebem cartões com desenhos de blocos onde se encontram escritos alguns comandos tais como: *dar um passo para frente*, *repetir <n> vezes*, *pegar objeto*, *soltar objeto*, etc. De posse dos cartões, os alunos constroem então um programa responsável por fazer com que o professor, ou algum dos colegas, realize uma tarefa pré-definida, como por exemplo, levar um objeto de um ponto a outro da sala seguindo um determinado caminho. Uma vez o programa montado, o robô humano executa os comandos conforme eles são lidos em voz alta pelos alunos, que observam se a tarefa planejada é executada corretamente. Caso ocorra alguma falha na realização da tarefa, as crianças buscam identificá-la, corrigindo e executando novamente o programa até que o robô humano finalize a tarefa com êxito.

As narrativas são utilizadas para se contextualizar o uso dos dispositivos de robótica e estruturas de programação sendo trabalhadas. Como exemplo podemos citar a narrativa utilizada para a atividade com o protótipo de peixe robô de garrafa pet (ver capítulo 3.1) onde são trabalhados o sensor de luz, o buzzer, LEDs e o motor DC, e possui a dinâmica descrita a seguir:

O peixe robô é apresentado aos alunos a partir de uma narrativa que, resumidamente, fala de um grupo de mergulhadores que estão em busca de um tesouro perdido no fundo de uma cratera submarina onde o oceano é absolutamente escuro. Nas imediações dessa cratera, existem peixes mutantes devoradores de outros peixes e também de seres humanos e que possuem uma luz na cabeça para atrair suas presas. Para se protegerem desses peixes, os mergulhadores tiveram a

ideia de utilizar um peixe robô que ficaria nadando em círculos na entrada da cratera de maneira que, quando o peixe mutante se aproximasse da caverna, e sua luz iluminasse o peixe robô, este disparasse um alarme alertando os mergulhadores, que então se esconderiam em gaiolas de ferro posicionadas dentro da cratera submarina.

Contada a história, o peixe robô é apresentado aos alunos, e estes são informados de que fazem parte da equipe de pesquisadores-programadores escolhida para desenvolver um programa que faça o peixe robô funcionar como o esperado. Uma vez programado o robô, com a sala às escuras, os alunos brincam, com o auxílio de uma lanterna, de peixe-mutante e mergulhadores simulando a situação descrita na história

4.3 CONTEÚDOS

- Robótica Educacional:
 - Sensores;
 - Atuadores;
 - Circuitos Eletrônicos;
 - Placas de Prototipagem Eletrônica.
- Conceitos Básicos de Programação:
 - Estrutura de Sequenciação;
 - Estrutura de Repetição Contada (repita <n> vezes);
 - Estrutura de Repetição Condicional (enquanto <condição> faça);
 - Operadores de Comparação (igual, diferente, maior que, menor que).

Os conteúdos de programação foram escolhidos com base no currículo da disciplina de introdução à programação de computadores de algumas Universidades Federais e no currículo da DevTech (BERS *et al*, 2010), anteriormente mencionado. O conceito de variável, um dos primeiros a serem tratados nos cursos básicos de programação, não foi abordado porque o mesmo só é visto por crianças do quarto ciclo (Ensino Fundamental II). Já os conceitos de sequenciamento, repetição, condição e comparação fazem parte do dia a dia das crianças.

No plano de curso original, havia-se planejado trabalhar, além das estruturas de programação acima mencionadas, as estruturas de decisão simples e composta. No entanto, as demandas apresentadas pelos alunos durante a realização da oficina

(como será visto no capítulo 5) tornaram necessário uma reformulação geral dos planos de aula e, como resultado desse processo, acabou não sendo possível trabalhar-se essas duas estruturas.

Cabe aqui mencionar que todos os conteúdos foram trabalhados em um nível bastante elementar.

4.4 DURAÇÃO

A oficina teve vinte e uma horas de duração, divididas em três encontros semanais de noventa minutos, totalizando quatorze encontros.

4.5 PLANOS DE AULA

O detalhamento dos planos de aula, com a especificação e descrição de todas as atividades realizadas, bem como dos conteúdos abordados, materiais adotados, desenvolvimento metodológico e itens de avaliação de aprendizado utilizados, encontra-se no apêndice A⁸⁴: Planos de Aula.

4.6 PARTICIPANTES

A oficina contou com a participação de nove crianças da vila Paraíso⁸⁵ (uma comunidade de baixa renda da cidade do Rio de Janeiro), sendo seis meninos e três meninas: quatro do quarto ano: Bruno, Daniel, Diana e José; três do terceiro ano: Hilton, Marcelo e Ivana⁸⁶; uma do segundo ano e uma do primeiro ano. Destas, apenas Diana, do quarto ano, não concluiu a oficina. As duas crianças menores foram admitidas na oficina por questões “diplomáticas” relacionadas à associação de moradores da comunidade de origem dos participantes. No entanto, os resultados por elas apresentados não foram considerados no âmbito deste trabalho porque elas não se encaixavam em todos os critérios de inclusão da pesquisa, sendo eles:

⁸⁴ Disponível também em CD anexo à dissertação impressa e, futuramente, também no site do LIVRE (<http://www.nce.ufrj.br/ginape/livre/>)

⁸⁵ Este é um nome fictício utilizado para se manter o sigilo ético.

⁸⁶ Estes são nomes fictícios utilizados para se manter o sigilo ético.

- Residir na vila Paraíso.
- Ser aluno do terceiro ou quarto ano do ensino fundamental de uma escola pública do município do Rio de Janeiro.
- Possuir familiaridade com o uso de computadores.
- Não possuir experiência anterior com programação de computadores.

A decisão por se trabalhar com um número reduzido de crianças foi tomada com base nos seguintes fatores:

- Viabilidade: para o desenvolvimento da oficina foi necessária a aquisição de um certo número de sensores, atuadores e placas Arduino, bem como a confecção das caixinhas mágicas, o que demanda verba e pessoal. Dentro das possibilidades do projeto, tanto financeiras quanto de pessoas disponíveis para a realização desse trabalho, o que se tornou possível efetivar foi material suficiente para ser utilizado por quatro grupos ou indivíduos. Além disso, a opção por coletar dados em vídeo das ações de cada uma das crianças durante a construção dos programas em seus computadores (ver 4.7), tornaria inviável uma avaliação pormenorizada e cuidadosa de todas as imagens coletadas caso a oficina contasse com um número maior de participantes, o que poderia acarretar em perdas importantes de dados.
- Atividades realizadas em duplas: em um "pré-teste" realizado com seis crianças divididas em dois trios, utilizando cada grupo um computador com o DB4K e uma caixinha mágica, foi possível observar que, em cada grupo, uma criança assumiu uma postura mais passiva enquanto as outras duas trabalhavam na construção dos programas solicitados. Essa constatação levou a conclusão de que seria mais indicado se trabalhar com duplas.
- Fundamentação Teórica: de acordo com Dumas e Redish (1999 *apud* KOSTARAS *et al*, 2010) um número apropriado de participantes para um teste de usabilidade ficaria entre cinco e doze. Nielsen e Landauer (1993 *apud* KOSTARAS *et al*, 2010), sugerem quantidades de participantes próximas a essas: cinco para testes de observação que busquem apenas descobrir problemas de usabilidade e dez para testes que

tenham por objetivo fazer análises estatísticas sobre a extensão dos possíveis problemas de usabilidade existentes. E Rubin (1994 *apud* KOSTARAS *et al*, 2010) propõe oito participantes.

4.7 COLETA DE DADOS E AVALIAÇÃO

Os métodos utilizados para a coleta de dados qualitativos durante a realização da oficina foram:

- Registro das ações dos usuários: utilização de uma filmadora para o registro das dinâmicas dos alunos no ambiente de trabalho e de um software de captura de tela para o registro em vídeo das ações realizadas pelos usuários em seus computadores.
- Observações de campo: anotação em um “diário de bordo” das observações feitas no decorrer de cada encontro.
- Aplicação de questionários.

(COHEN, MANION e MORRISON, 2007)

Esses dados foram então utilizados para a realização das seguintes avaliações, cujos resultados encontram-se disponíveis nos capítulos 5, 6 e 7, respectivamente:

- Avaliação do aprendizado das estruturas básicas de programação e sua relação com aspectos do Pensamento Computacional.
- Avaliação da usabilidade do Ambiente de Programação em Blocos DB4K.
- Avaliação dos recursos pedagógicos.

O Quadro 1 apresenta um mapa sucinto do desenho final da oficina, possibilitando uma visualização global e sintética das principais atividades desenvolvidas e dos conteúdos trabalhados durante as aulas. Este quadro tem por objetivo auxiliar o leitor a “localizar-se” mais facilmente durante a apreciação dos capítulos dedicados à avaliação dos dados coletados. O detalhamento de todas as atividades realizadas durante a oficina, bem como dos conteúdos trabalhados e dos materiais e estratégias pedagógicas utilizados, como já mencionado, encontra-se disponível nos planos de aula presentes no apêndice A.

ESTRUTURA SEQUENCIAL	
Pensamento Computacional: capacidade de abstração / noções algorítmicas de fluxo de controle / depuração e detecção sistemática de erros	
Blocos de Controle: <i>esperar</i>	
Dispositivos: LED / motor DC	
Primeiro Encontro	Aula 1
O robô humano	
• Programação sequencial com o robô humano.	
Segundo Encontro	Aula 2
A caixinha mágica	
• Programas de um único comando para o controle dos LEDs e do motor DC na caixinha mágica.	
Terceiro Encontro	Aula 3
O interior da caixinha mágica, motor DC e LED: montando meu primeiro circuito eletrônico	
• Construção de pequenos programas sequenciais a partir de enunciados escritos, fazendo uso do bloco de controle <i>esperar</i> para controlar LEDs e o motor DC.	
ESTRUTURA DE REPETIÇÃO CONTADA	
Pensamento Computacional: capacidade de abstração / noções algorítmicas de fluxo de controle / depuração e detecção sistemática de erros.	
Blocos de Controle: <i>esperar / repetir</i>	
Dispositivos: LED / motor DC / display de 7 segmentos / LCD	
Primeiro Encontro	Aula 4
LEDs e os blocos <i>repetir</i> e <i>esperar</i> : montando meu segundo circuito eletrônico	
• Brincadeira baseada na técnica de computação desplugada para apresentação da <i>estrutura de repetição contada</i> .	
• Montagem de um pequeno circuito com LEDs na <i>protoboard</i> .	
• Construção de programas para controle do circuito montado, fazendo uso dos blocos de controle <i>repetir</i> e <i>esperar</i> .	
• Brincadeira da palavra código, fazendo uso dos blocos de controle <i>repetir</i> e <i>esperar</i> para o controle do conjunto LEDs presentes na caixinha mágica.	

Segundo Encontro	Aula 5
LED e os blocos <i>repetir</i> e <i>esperar</i> (Parte 2)	
• Robô-humano: atraso (bloco <i>esperar</i>)	
• Construção de programas para o controle de LEDs fazendo uso do bloco <i>esperar</i> .	
• Robô-humano: <i>estrutura de repetição contada</i> (bloco <i>repetir</i>)	
• Construção de programas para o controle de LEDs, com e sem o uso do bloco de controle <i>repetir</i> .	
Terceiro Encontro	Aula 6
LED e os blocos <i>repetir</i> e <i>esperar</i> (Parte 3)	
• Construção de programas para controle dos LEDs com variações no posicionamento e no tempo de atraso do bloco <i>esperar</i> .	
• Robô-humano: <i>estrutura de repetição contada</i> (bloco <i>repetir</i>) e entendimento de que programas que utilizam os mesmos comandos, mas organizados de diferentes formas, apresentam resultados diferentes.	
• Construção de programas para o controle de LEDs com o uso do bloco <i>repetir</i>	
Quarto Encontro	Aula 7
O LCD e o display de 7 segmentos / Tesouro submarino: conhecendo o peixe robô	
• Narrativa do peixe robô: apresentação do peixe robô a ser programado apenas em aulas futuras (atividade motivacional).	
• Revisão, por meio de um debate, do uso dos blocos de controle <i>esperar</i> e <i>repetir</i> em conjunto com os blocos <i>acender LED</i> e <i>apagar LED</i> .	
• Brincadeira para apresentação do display de 7 segmentos.	
• Construção de programas para o controle do display de 7 segmentos com o uso do bloco de controle <i>esperar</i>	
• Construção de programas para o controle conjunto dos LEDs, LCD e diplay de 7 segmentos, com o uso dos blocos de controle <i>repetir</i> e <i>esperar</i>	
Quinto Encontro	Aula 8
Construindo programas a partir da observação	
• Narrativa do peixe robô (apresentação do peixe robô às crianças que haviam faltado a aula anterior).	
• Programação do display de 7 segmentos com uso do bloco de controle <i>esperar</i> .	

• Construção de programas a partir da observação de vídeos onde a caixinha mágica é exibida mostrando um determinado “comportamento” (entre os quais, a repetição de determinadas ações).	
ESTRUTURA DE REPETIÇÃO CONDICIONAL	
Pensamento Computacional: capacidade de abstração / noções algorítmicas de fluxo de controle / depuração e detecção sistemática de erros / lógica condicional / decomposição estruturada de problemas	
Blocos de Controle: <i>esperar / repetir / enquanto</i>	
Dispositivos: LED / motor DC / display de 7 segmentos / LCD / buzzer / LED RGB / servo motor / sensor de distância / sensor de luz / sensor de temperatura	
Primeiro Encontro	Aula 9
LCD, os sensores de luz e temperatura e o bloco <i>enquanto</i>	
• Programação do LCD com uso do bloco de controle <i>esperar</i> .	
• Robô-humano: <i>estrutura de repetição condicional</i> (bloco <i>enquanto</i>).	
• Debate: bloco <i>repetir</i> x bloco <i>enquanto</i> .	
• Montagem (toda turma em conjunto) de um programa com uso do bloco <i>enquanto</i> (em conjunto com o sensor de distância, motor DC e LEDs).	
• Alternância do uso dos sensores de luz e temperatura no mesmo programa.	
Segundo Encontro	Aula 10
Programando o peixe robô	
• Programação do LCD com uso do bloco de controle <i>esperar</i> .	
• Construção de programas a partir da observação de vídeos onde a caixinha mágica era exibida mostrando um determinado “comportamento” (primeiro um vídeo que “sugeriria” o uso do bloco <i>repetir</i> e posteriormente um que “sugeriria” o uso do bloco <i>enquanto</i>).	
• Construção do programa para o controle do peixe robô, com base na observação do seu funcionamento (blocos de controle <i>enquanto</i> e <i>esperar</i> , LEDs, motor DC, buzzer, LED RGB).	

Terceiro Encontro	Aula 11
A legião de vampiros: O morcego robô	
• Construção de pequenos morcegos robô de garrafa pet.	
• Programação dos morcegos robô para funcionarem do mesmo modo que o morcego do professor (uso de LEDs e do motor DC em conjunto como o sensor de distância e blocos de controle <i>enquanto</i> e <i>esperar</i>).	
Quarto Encontro	Aula 12
A legião de vampiros: o carrinho robô vampiro	
• Revisão do uso da <i>estrutura de repetição condicional</i> .	
• Programação dos morcegos robô para funcionarem da maneira desejada.	
• Programação do carrinho robô para funcionar como observado (uso dos sensores de distância e luz para mudar o “comportamento” do carrinho).	
• Programação do carrinho robô para funcionar da maneira planejada pelos alunos.	
Quinto Encontro	Aula 13
O cachorro robô e o braço robótico: a missão espacial	
• Apresentação do cachorro robô de garrafa pet.	
• Programação do servo motor na caixinha mágica.	
• Programação do cachorro robô de garrafa pet (servo motor, sensores de distância e luz, dispositivos de locomoção do carrinho, blocos de controle <i>enquanto</i> e <i>esperar</i>)	
• Programação do braço robótico.	
Aula de Encerramento	Aula 14
Finalizando a oficina: projetando um robô	
• Observação de um vídeo com a caixinha mágica exibindo um determinado comportamento e descrição deste comportamento de forma textual.	
• Reconhecimento dos nomes dos atuadores e sensores trabalhados na oficina.	
• Verificação da correção de um código apresentado pelo professor em relação a um vídeo mostrando a caixinha exibindo um determinado comportamento.	
• Desenho (projeto) livre de robôs que fizessem uso de alguns sensores e atuadores trabalhados em aula.	

Quadro 1: Mapa sintético das atividades e conteúdos trabalhados na oficina

5 AVALIAÇÃO DO APRENDIZADO DAS ESTRUTURAS BÁSICAS DE PROGRAMAÇÃO E SUA RELAÇÃO COM ASPECTOS DO PENSAMENTO COMPUTACIONAL

A avaliação do aprendizado das estruturas básicas de programação seguiu o modelo utilizado pelo grupo DevTech da TUFFS University⁸⁷ (BERS *et al*, 2010). Esse modelo faz uso de uma escala *Likert* com os seguintes níveis:

5	4	3	2	1	0
Aquisição ou realização completa do objetivo, tarefa ou conteúdo	Aquisição ou realização quase completa do objetivo, tarefa ou conteúdo	Aquisição ou realização parcial do objetivo, tarefa ou conteúdo	Aquisição ou realização bastante incompleta do objetivo, tarefa ou conteúdo	Não aquisição ou realização do objetivo, tarefa ou conteúdo	Nem sequer tentou

Quadro 2: Escala *Likert* utilizada para a avaliação do aprendizado

Essa escala é aplicada a itens referentes à aquisição ou exercício das habilidades pretendidas em cada aula, como por exemplo:

1	Entende o funcionamento do sensor de distância	5	4	3	2	1	0
2	Monta corretamente o circuito proposto	5	4	3	2	1	0
3	Compreende a função do bloco <i>enquanto &ltcondição> faça</i>	5	4	3	2	1	0
4	Utiliza adequadamente o bloco <i>enquanto &ltcondição> faça</i> , inserindo o bloco com o sensor correto como fator condicionante	5	4	3	2	1	0
5	Programa corretamente o motor DC e o LED em conjunto com o sensor de distância	5	4	3	2	1	0
Obs.							

Quadro 3: Exemplo de itens de referência para avaliação de habilidades específicas

Os itens de avaliação dos resultados para cada uma das aulas encontram-se no apêndice A⁸⁸: Planos de Aula.

⁸⁷ <http://ase.tufts.edu/devtech>

⁸⁸ Disponível também em CD anexo à dissertação impressa e, futuramente, também no site do LIVRE (<http://www.nce.ufrj.br/ginape/livre/>)

Em associação à avaliação das habilidades adquiridas ou exercitadas, é feita também uma avaliação acerca da habilidade de depuração e correção dos programas desenvolvidos, conforme o seguinte modelo:

1	Percebe que alguma coisa não está funcionando como esperado	5	4	3	2	1	0	NA
2	Conserva a meta original	5	4	3	2	1	0	NA
3	Tem uma hipótese para a causa do problema	5	4	3	2	1	0	NA
4	Tenta solucionar o problema	5	4	3	2	1	0	NA
Obs.								

Quadro 4: itens de referência para avaliação da habilidade de depuração e correção dos programas desenvolvidos

A avaliação do desenvolvimento ou exercício das habilidades do Pensamento Computacional, bem como de outros aspectos relacionados à maturidade cognitiva das crianças, foi realizada a partir das competências que se entedia como sendo necessárias para a construção adequada dos programas propostos (ver capítulo 2.4).

Como visto no capítulo 2.3, a programação de computadores é considerada uma competência fundamental da Ciência da Computação e uma ferramenta chave de apoio às habilidades cognitivas relacionadas ao Pensamento Computacional (GROVER e PEA, 2013), sendo que, para o desenvolvimento de muitas dessas habilidades é necessário que se avance bastante nas práticas envolvidas na codificação de sistemas computacionais. Desse modo, em se tratando de uma breve oficina de introdução à programação de computadores para crianças do Ensino Fundamental I, não seria possível o desenvolvimento ou exercício de uma gama muito ampla de habilidades do Pensamento Computacional. Assim, tendo-se como base os conteúdos tratados durante a oficina, entendeu-se como sendo passíveis de serem trabalhadas e avaliadas as seguintes habilidades:

- a) **Capacidade de abstração:** A capacidade de abstração é considerada como sendo a "pedra fundamental" do Pensamento Computacional (GROVER e PEA, 2013). Jeannette Wing, pesquisadora que cunhou o termo *Computational Thinking*, a define como sendo “o processo mais importante e de nível mais elevado no Pensamento Computacional, [...] usado na definição de padrões, na generalização de instâncias

específicas e parametrização [...]” (WING, 2011, p.20, tradução nossa). Como um dos exemplos de abstração ligados à Ciência da Computação, Wing cita o algoritmo, definido por ela como sendo “a abstração de um processo que pega entradas, executa uma sequência de passos e produz saídas para satisfazer a um objetivo desejado” (WING, 2011, p.20, tradução nossa). Tendo em vista essas definições, no escopo desse trabalho, a capacidade de abstração foi avaliada com base na habilidade das crianças de observarem a execução ou descrição de um processo do mundo real e abstrair esse processo na forma de um programa de computador.

- b) **Compreensão de Fluxos de Controle:** “Fluxo de controle é o termo utilizado para decidir quais comandos do programa são executados em qual ordem” (ARNOLD, JAMES e DAVID, 2009, p.36). No caso de um programa construído com o Duinoblocks4Kids, cada bloco representa um comando, sendo esses comandos executados pelo computador, um a um, em um *loop* infinito, do primeiro bloco (posicionado no topo da “pilha” de comandos), até o último bloco (posicionado na base pilha). Alguns blocos, como o *Acender LED* e *Girar motor DC*, representam “comandos simples”. Já os blocos de controle: *repetir*, *enquanto*, *se e se/senão*, representam estruturas que podem ser vistas, sintaticamente, como um único comando composto por um conjunto de comandos que podem ou não ser realizados uma ou n vezes. Tanto a construção de programas puramente sequenciais, ou seja, que utilizam apenas “comandos simples”, quanto a criação de programas mais elaborados, que façam uso de estruturas de repetição contada ou condicional, por exemplo, só se fazem possíveis se as crianças forem capazes de assimilar algumas noções básicas de fluxo de controle.
- c) **Uso da Lógica Condicional:** A lógica condicional é utilizada em programação para que um conjunto de instruções seja executado somente quando, enquanto ou até que determinadas condições sejam satisfeitas. A satisfação ou não dessas condições acarreta na tomada de diferentes decisões que levam a lógica do programa a seguir diferentes fluxos (BHATTACHARYA, 2016). A *estrutura de repetição condicional enquanto*, trabalhada durante a oficina, é um exemplo de estrutura de

programação utilizada para este fim, sendo deste modo necessário que as crianças consigam desenvolver um entendimento, ainda que rudimentar, acerca da lógica condicional, para que possam construir programas que utilizem esse tipo de estrutura.

- d) **Decomposição de problemas:** Decomposição de problemas é a atividade de "desmembrar um problema em partes menores que possam ser mais facilmente resolvidas" (BARR e STEPHENSON, 2011, p.52, tradução nossa). Mesmo programas simples, que contenham, por exemplo, uma sequência de poucos comandos simples e dois laços de repetição, cada um destes contendo também uma sequência de poucos comandos, podem ser divididos em, digamos, 3 partes menores (ver exemplo na Figura 26), de modo que cada uma dessas partes possa ser desenvolvida e testada separadamente, diminuindo assim a complexidade do que precisa ser analisado, por exemplo, na busca por erros. Esse gênero de exercício pode auxiliar as crianças a perceberem a possibilidade do uso desse tipo de estratégia na solução de problemas de outra natureza.

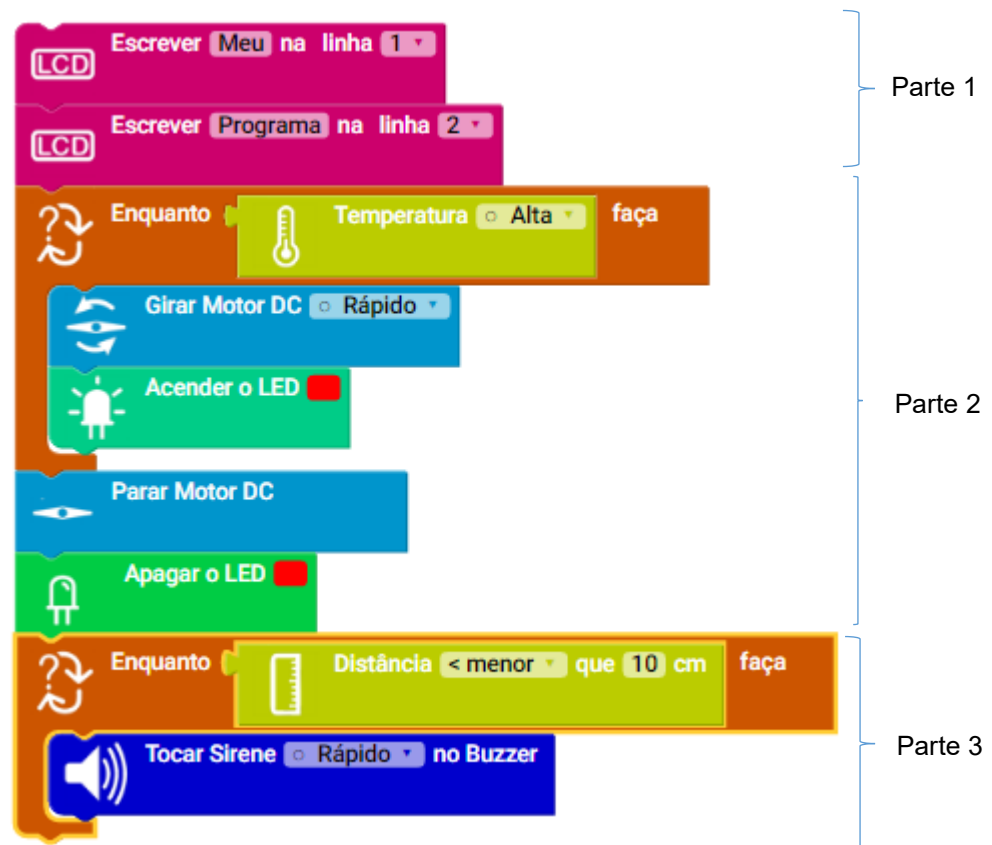


Figura 26: Exemplo de decomposição de problemas

e) **Depuração:** “Depuração é a atividade [...] de encontrar e eliminar os erros (*bugs*) de um programa” (AGUILAR, 2008, p.672). O Ambiente de Programação em Blocos DB4K não possibilita a ocorrência de erros de compilação⁸⁹. Desse modo, a depuração fica focada na detecção e correção de possíveis erros na lógica do programa construído. O trabalho conjunto do Ambiente de Programação em Blocos com os materiais de robótica utilizados na oficina possibilita um exercício inicial acerca da depuração sistemática de erros a partir da realização da sequência de atividades presente na Figura 27.

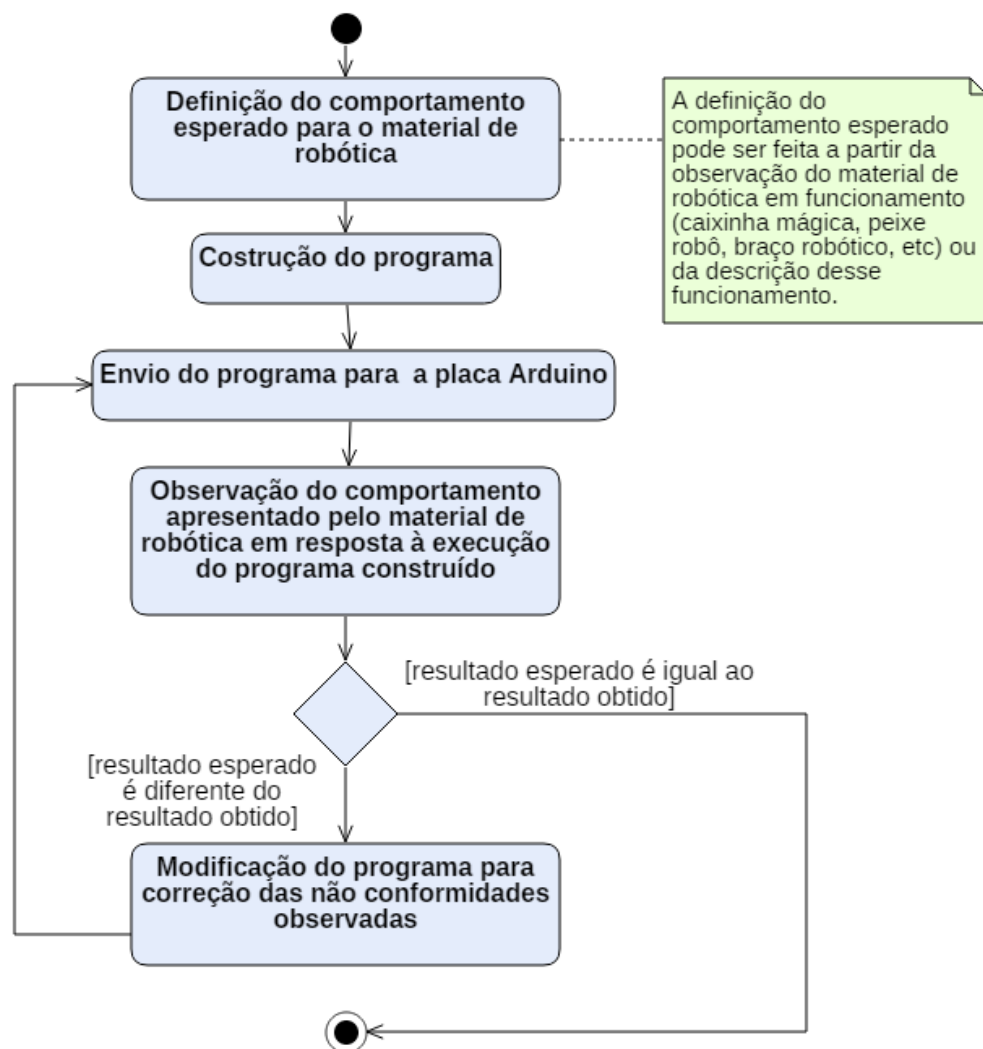


Figura 27: Diagrama de Atividades da tarefa de depuração de erros

⁸⁹ Um compilador, grosso modo, é um software que varre o código do programa escrito pelo programador à procura de erros e, posteriormente, gera uma versão deste programa em uma “linguagem” que o computador possa entender. Erros de compilação são erros relacionados à escrita do código do programa, como por exemplo, erros de sintaxe.

Este exercício traz para as crianças, dentre outros benefícios, a possibilidade de uma maior autonomia em relação ao seu aprendizado, bem como o entendimento de que uma falha pode apontar o caminho para um resultado positivo (BARR e STEPHENSON, 2011).

Para apresentação do aprendizado das estruturas de programação e habilidades do Pensamento Computacional acima listadas, optou-se por organizar este capítulo conforme descrito no capítulo 1.5 (Metodologia) e seguindo a mesma estrutura presente no Quadro I: Mapa sintético das atividades e conteúdos trabalhados na oficina (apresentado no capítulo 4), qual seja: Tomando as estruturas de programação como elemento norteador, apresentando-se então, para cada uma delas, encontro a encontro, as atividades desenvolvidas e os resultados observados. Durante a apresentação dos resultados, as considerações acerca das respostas dadas pelas crianças aos desafios propostos são feitas ora focando o indivíduo, ora focando o grupo, e ora mesclando esses dois olhares.

Cabe mencionar que, no tocante às atividades, serão apresentadas, dentre todas as atividades realizadas durante a oficina, apenas uma descrição breve daquelas que estão diretamente relacionadas com o aprendizado de programação. O mesmo aplica-se aos recursos pedagógicos adotados. Como já mencionado, o detalhamento de cada uma das atividades, bem como dos recursos pedagógicos utilizados para sua realização, encontra-se no apêndice A⁹⁰.

Uma vez que a avaliação do aprendizado das crianças deu-se de forma processual, torna-se necessária uma descrição detalhada dos processos observados de modo a permitir a ratificação das inferências realizadas. Assim sendo, para possibilitar ao leitor o acesso tanto aos pormenores envolvidos na avaliação do aprendizado das crianças (apresentados nas seções 5.1 a 5.4), quanto a uma visão mais concisa dos resultados observados, optou-se por incluir, na última seção deste capítulo, uma apresentação sintetizada desses resultados.

⁹⁰ Disponível também em CD anexo à dissertação impressa e, futuramente, também no site do LIVRE (<http://www.nce.ufrj.br/ginape/livre/>)

5.1 PROGRAMAÇÃO SEQUENCIAL

5.1.1 Primeiro Encontro (Aula 1)

5.1.1.1 Atividades Desenvolvidas

O aprendizado da programação sequencial partiu inicialmente do estabelecimento do entendimento de que um programa de computador é, basicamente, uma sequência de comandos. A primeira atividade realizada nesse sentido foi uma brincadeira baseada na técnica de programação desplugada, chamada “programando o robô humano” (MICHALAKIS, 2013). Para a realização desta brincadeira, as crianças, em grupos, receberam cartões com desenhos de blocos onde se encontravam escritos alguns comandos tais como: dar um passo para frente e pegar objeto. De posse desses cartões, as crianças precisavam construir programas que fizessem com que o professor, ou algum dos colegas que estivesse fazendo o papel de robô, se movimentasse pela sala pegando objetos e levando-os a locais previamente definidos. Uma vez os programas montados, o robô humano executava os comandos conforme eles eram lidos em voz alta pelas crianças, o que permitia a elas observarem se, a partir da execução dos comandos na ordem por elas estabelecida, o robô realizava corretamente a tarefa planejada.

5.1.1.2 Resultados

Como os alunos montavam os programas para o computador humano em grupo, não foi possível, por meio dessa atividade, avaliar o nível de entendimento de cada uma das crianças acerca da algoritmização de um processo. O que pôde ser observado é que, nos exercícios em que os objetos a serem colhidos pelo robô humano eram colocados em posições onde o robô precisava fazer mais de uma movimentação à direita ou à esquerda, ou seja, girar 180 graus, as crianças não conseguiam montar o algoritmo correto, pois, aparentemente, tinham dificuldade em visualizar mentalmente o resultado da soma dos dois movimentos. Já a construção de sequências simples, como dar passos para frente, virar uma vez à direita, virar uma vez à esquerda e pegar objetos, ao final da atividade, estavam sendo algoritmizadas

pelas crianças sem maiores dificuldades, exceto pelo fato de que faziam alguma confusão entre direita e esquerda, solicitando algumas vezes que o robô humano girasse para um lado quando queriam que ele realizasse o giro para o outro.

5.1.2 Segundo Encontro (Aula 2)

5.1.2.1 Atividades Desenvolvidas

No segundo encontro foram desenvolvidas atividades que consistiam na construção de programas de um único comando (Figura 28) no ambiente de programação, como por exemplo, fazer um motor DC girar ou fazer um LED acender. Para este tipo de tarefa era necessário apenas que o aluno fizesse:

- A escolha do bloco adequado no menu de acesso aos blocos;
- A colocação do bloco na área de trabalho;
- A seleção do valor do parâmetro (cor do LED ou velocidade do Motor);
- O Envio do programa para a caixinha mágica.



Figura 28: Exemplos de programas de um único comando

5.1.2.2 Resultados

Pôde-se observar em todas as crianças, na tarefa onde era solicitado que montassem programas de um único comando, a expectativa de que a simples disposição de um bloco na área de trabalho, ou a mudança de um parâmetro de algum bloco, resultasse em uma modificação no estado dos dispositivos presentes na caixinha mágica. Ou seja, se colocavam o bloco *acender LED* na área de trabalho, ficavam esperando o LED acender na caixinha sem que "mandassem" o programa executar (apertando o botão *executar programa*). Esse comportamento se repetiu algumas vezes, mas, aos poucos, as crianças desenvolveram o entendimento de que

era primeiro necessário montar a sequência de comandos e escolher os parâmetros adequados para só então enviar o programa para a caixinha executar.

Pode-se, com base nessa observação, especular que esse comportamento resulte do fato de usuários finais de sistemas de computador, via de regra, observarem algum evento em resposta a cada ação executada, como por exemplo, ao apertar uma tecla em um editor de texto, vê-se uma letra aparecer na tela do computador; ao "derramar-se" uma cor de tinta em uma forma dentro de um editor de imagens, vê-se essa imagem ser preenchida com essa cor, e assim por diante. O mesmo se dá no caso dos jogos digitais, muito comumente utilizados pelas crianças, nos quais, de modo geral, a cada ação realizada corresponde uma resposta visual imediata.

No caso da programação de computadores, as coisas não se dão dessa forma. É preciso que se consiga construir mentalmente o resultado esperado e depois, por meio de um programa, organizar o conjunto de ações que, acredita-se, levará a esse resultado, para só então "solicitar" ao computador que realize a tarefa desejada. Esta pode ser uma atividade não muito trivial para algumas crianças entre oito e nove anos, uma vez que, por estarem no início do período operacional-concreto, a habilidade de realizar operações mentais (ainda que apoiadas em experiências concretas) e de conseguir distinguir o todo e suas partes (importantes para a realização deste tipo de tarefa) podem não estar suficientemente desenvolvidas. Especialmente em se levando em consideração que, como já mencionado, a maturidade cognitiva é dependente “das características biológicas do indivíduo e de fatores educacionais, sociais” (FURTADO, BOCK e TEIXEIRA, 2001, p.102).

5.1.3 Terceiro Encontro (Aula 3)

5.1.3.1 Atividades Desenvolvidas

No terceiro encontro em que se trabalhou a *estrutura sequencial de programação*, foram desenvolvidas atividades nas quais os alunos precisavam construir programas compostos por uma sequência simples de comandos a partir de um resultado esperado, como por exemplo:

- Fazer o motor DC girar rápido por 10 segundos com o LED vermelho aceso e depois parar o motor, apagar o LED e parar o programa (Figura 29 (a)).
- Acender o LED azul, esperar 1 segundo, depois apagar o LED azul e esperar 1 segundo ((Figura 29 (b))).

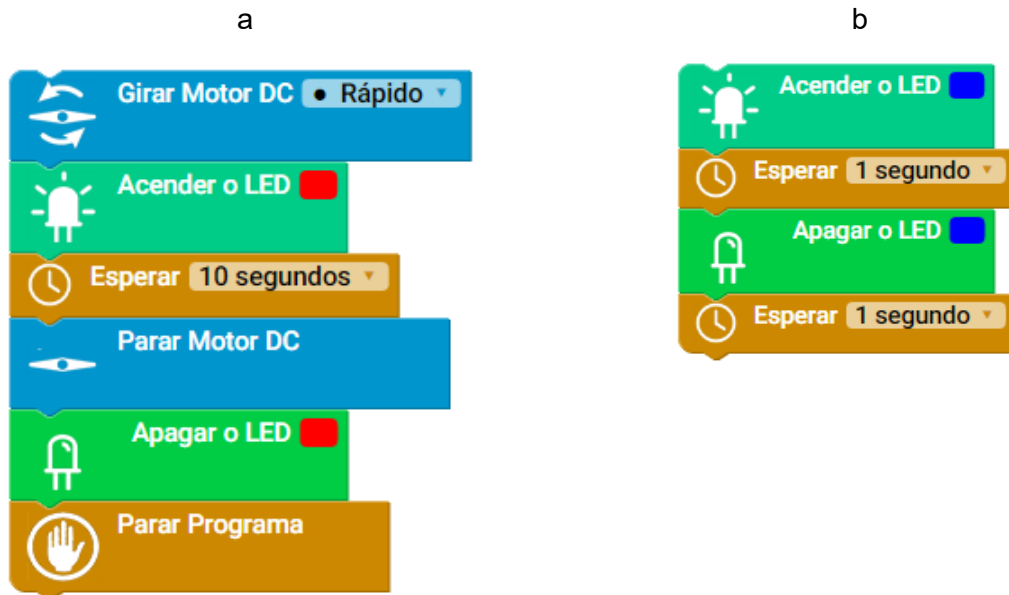


Figura 29: Exemplos de programas utilizando apenas a estrutura sequencial

Esses desafios eram apresentados às crianças por meio de textos exibidos pelo projetor.

5.1.3.2 Resultados

Por meio dos programas construídos pelas crianças em resposta aos desafios propostos, pôde-se observar que, das sete crianças do terceiro e quarto anos presentes inicialmente na oficina, quatro demonstraram um entendimento claro da relação entre a sequência de comandos utilizados e o comportamento observado nos dispositivos presentes na caixinha mágica, sendo elas: Daniel, Diana e José, do quarto ano, e Hilton, do terceiro ano. Esse fato pôde ser constatado não apenas por meio dos programas construídos por essas crianças em resposta imediata ao enunciado das tarefas, mas também pelo reconhecimento e correção de erros de programação

realizados quando o resultado observado na caixinha mágica não correspondia ao comportamento por elas esperado.

No que tange ao desenvolvimento e prática de habilidades relacionadas ao Pensamento Computacional, essa construção mental da sequência das ações necessárias para a obtenção de um determinado resultado, e sua posterior materialização na forma de um programa de computador, reflete tanto a capacidade de abstração quanto noções algorítmicas de fluxo de controle por parte das crianças que conseguiram realizar essas tarefas.

Outra observação interessante foi a de que as crianças tendiam a construir seus programas a partir da leitura fragmentada do texto que apresentava o desafio, montando o programa com base na sequência das palavras dispostas no texto e não da ideia completa que o texto continha. De acordo com O'Connell (2007, p.50), “A interpretação requer muito mais que a simples decodificação dos sinais gráficos, mais que a simples compreensão ou reconhecimento do conteúdo de um texto”. É a fase mais refinada da leitura e está relacionada intimamente com o repertório de experiências, tanto acadêmicas quanto da vida cotidiana do leitor (O'CONNELL, 2007). Ou seja, essa aparente dificuldade no entendimento da informação global contida no texto dos enunciados pode ter uma série de explicações, tanto relacionadas à maturidade cognitiva quanto ao acervo de experiências prévias das crianças, de maneira que qualquer possível inferência nesse sentido necessitaria de um estudo aprofundado do tema, o que foge ao escopo desta pesquisa.

Este tipo de dificuldade pôde ser observado, dentre outras ocasiões, quando as quatro crianças que haviam entendido de forma mais clara que a construção do programa se dava pelo sequenciamento dos comandos, ao tentarem executar a tarefa descrita como: Girar o motor DC rápido por 10 segundos com o LED vermelho aceso, parar o motor, apagar o LED e parar o programa; traduziram esse enunciado para o programa como a sequência *de* comandos apresentadas na Figura 30(a), quando o correto seria a sequência apresentada na Figura 30(b).

No entanto, tal falha na construção da ideia do que havia sido pedido foi reconhecida por Daniel quando ele observou o resultado da execução do programa na caixinha mágica e reavaliou o enunciado da tarefa, e por Hilton, quando o professor leu novamente o enunciado dando ênfase à preposição “com”. Sendo que ambos, ao

perceberem o equívoco, corrigiram a ordem dos comandos, colocando o bloco para acender o LED antes do bloco para esperar 10 segundos.⁹¹

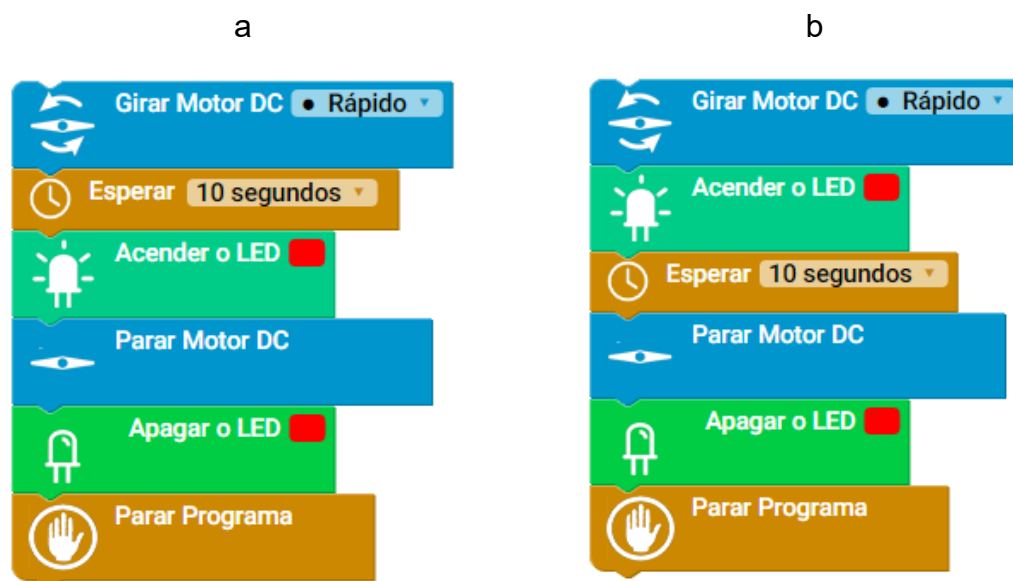


Figura 30: Interpretação equivocada do enunciado do programa proposto

Esse exercício de detecção de erros a partir da comparação do programa construído com o resultado observado na sua execução, como já mencionado, possibilita às crianças trabalharem a depuração e detecção sistemática de erros, uma habilidade do Pensamento Computacional. Além disso, a percepção das crianças da existência de um erro no programa a partir da comparação do resultado obtido com o texto do enunciado da tarefa proposta, aponta também o potencial desse tipo de atividade como uma prática capaz de trabalhar habilidades relacionadas à alfabetização funcional⁹².

É importante também relatar que, durante as seis primeiras aulas da oficina, as crianças compartilhavam o uso dos computadores em duplas ou trios, buscando-se com isso estimular um trabalho cooperativo no desenvolvimento dos programas. No entanto, essa expectativa não se concretizou, pois as crianças com mais "atitude", ou que conseguiam desenvolver um entendimento dos conteúdos trabalhados mais

⁹¹ Devido a um problema no registro das atividades, não foi possível observar, nesse primeiro momento, as possíveis percepções e correções de erros das demais crianças em relação aos seus programas. O problema ocorrido refere-se ao encerramento dos programas de captura de tela por parte de algumas crianças. Entre abrir e fechar outras janelas de programa elas acabavam fechando também, sem intenção, a janela do programa de captura.

⁹² Alfabetismo funcional refere-se à "capacidade de utilizar a leitura e a escrita para fins pragmáticos, em contextos cotidianos, domésticos ou de trabalho, [...]" (RIBEIRO, 1997, p.145).

facilmente, passavam a construir sozinhas os programas propostos ou direcionar a ação dos outros membros do grupo quando eles tentavam construí-los, não permitindo desse modo que estes pudessem desenvolver, a seu tempo e modo, um entendimento próprio sobre o que estava sendo trabalhado.

Essa dinâmica acabou resultando na impossibilidade de se observar se esses alunos haviam desenvolvido ou não o entendimento da *estrutura sequencial de programação*, bem como a observação do exercício de aspectos relacionados ao desenvolvimento do Pensamento Computacional por parte dessas crianças. Por outro lado, observou-se o evidente contentamento de todas as crianças em verem, nos dispositivos de robótica presentes na caixinha mágica, a materialização dos programas por elas construídos.

5.2 ESTRUTURA DE REPETIÇÃO CONTADA

Havia uma expectativa inicial de que, em um encontro de 90 minutos, seria possível estabelecer-se um entendimento, ao menos parcial, da *estrutura de repetição contada*. Este prognóstico foi feito com base no programa da oficina proposta no trabalho da Tufts University, utilizado como base para o desenho da oficina realizada nesta pesquisa (ver capítulo 4.2). No entanto, as crianças precisaram de cinco encontros para compreenderem satisfatoriamente o uso dessa estrutura e também do bloco *esperar* (correspondente ao comando *delay* da linguagem *Wiring*)⁹³, o que acarretou o replanejamento de uma série de atividades.

5.2.1 Primeiro Encontro (Aula 4)

5.2.1.1 Atividades Desenvolvidas

No primeiro encontro no qual a *estrutura de repetição contada* foi trabalhada, foram desenvolvidas três atividades. Para a apresentação da estrutura foi realizada

⁹³ O computador executa as operações a ele solicitadas em uma velocidade muito alta, muitas vezes na casa dos nanossegundos. Desse modo, quando se deseja que o resultado da execução de um determinado comando, como acender LED, Apagar LED, escrever no LCD, etc, seja observável para o olho humano, torna-se necessário que se estipule um “atraso” no início da execução do próximo comando (ou seja, um tempo de espera). Isso faz-se necessário, por exemplo, para que se possa ver a luz do LED acesa antes que ela apague ou para se ver um texto escrito no LCD antes que ele seja limpo.

uma brincadeira na qual as crianças precisavam levantar e abaixar cartões em formato de estrela e maçã, um certo número de vezes por um determinado tempo, seguindo um programa em blocos apresentado pelo projetor, como no exemplo da Figura 31.

Em seguida, as crianças montaram um pequeno circuito com uma *protoboard* e uma placa Arduino Uno, composto por 1 LED azul e 1 LED verde. Uma vez o circuito montado, foi solicitado às crianças que fizessem um programa para acender o LED azul 4 vezes por 1 segundo e depois o LED verde 2 vezes por 3 segundos, usando o bloco *repetir*.

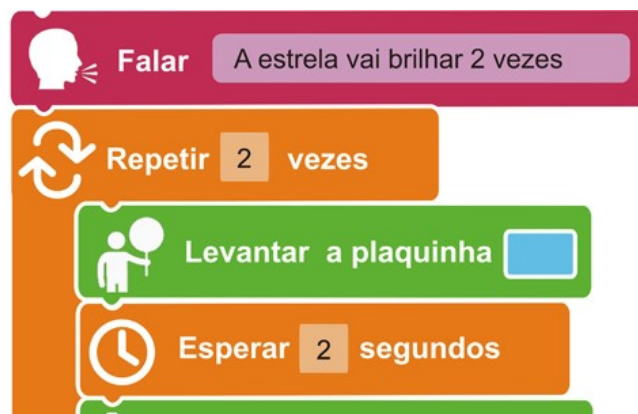


Figura 31: Exemplo de programa a ser executado pelo robô humano

Para as crianças que finalizaram a atividade acima mais rapidamente, foi solicitado também que fizessem um programa para escrever uma palavra através de um código no qual cada letra da palavra correspondia a um número de piscadas do LED de uma determinada cor, como no exemplo da Figura 32.

R	■ ■ ■ ■ ■
O	■ ■ ■ ■
B	■ ■
Ô	■ ■

Figura 32: Gabarito para construção do programa da palavra-código

5.2.1.2 Resultados

As crianças tiveram bastante dificuldade para realizar a atividade de montagem do circuito com os LEDs, ainda que fosse um circuito simples e existissem marcações coloridas indicando onde os dispositivos e fios deveriam ser encaixados. Era complicado para elas, por questões aparentemente relacionadas à maturidade motora, encaixarem as “pernas” dos LEDs ou dos resistores e os terminais dos *jumpers* nos “furos” da *proto-board* e da placa Arduino. Ficou evidente que, para a realização de mais atividades como esta, inclusive buscando o entendimento, por parte das crianças, da lógica eletrônica presente nos circuitos montados, seria necessária a realização de um curso relativamente mais longo. No entanto, apesar das dificuldades encontradas, foi uma tarefa que agradou bastante aos alunos.

Os exercícios nos quais foi solicitado que os LEDs acendessem por um determinado tempo um número específico de vezes, mostraram, como já observado nos exercícios de programação sequencial, que as crianças apresentam dificuldade em interpretar os textos dos enunciados. Isso ficou claro quando, por exemplo, ao ser apresentado o seguinte enunciado: Acender o LED azul 4 vezes por 1 segundo e depois o LED verde 2 vezes por 3 segundos; o programa construído foi o presente na Figura 33(a) quando o correto seria o programa presente na Figura 33(b). Ou seja, uma tradução para o programa não daquilo que o texto do enunciado representa, mas, da sua leitura sequencial não interpretada. Uma interpretação correta do texto do enunciado levaria as crianças ao entendimento de que acender o LED azul 4 vezes é o mesmo que fazê-lo piscar 4 vezes, o que acarretaria em acendê-lo e apagá-lo 4 vezes, mas, esse processo de construção do significado do texto não aconteceu. Há também que se considerar o fato de que o enunciado poderia estar melhor formulado e que o texto como está escrito necessita que se faça sobre ele uma inferência, uma dedução do que está se querendo de fato como resultado, o que provavelmente dificultou o entendimento do mesmo por parte das crianças.

Uma vez as crianças tendo sido alertadas de que os comandos a serem repetidos deveriam ficar dentro do bloco *repetir* e que o que estava sendo solicitado no enunciado era que os LEDs piscassem o número de vezes indicado, a dupla formada por Daniel e Diana (que vinham apresentando um bom entendimento dos



Figura 33: Programa construído a partir de uma interpretação equivocada do enunciado

conteúdos), embora tenha montado a sequência correta de comandos para fazer os LEDs piscarem, apresentou dificuldade quanto ao entendimento do uso do bloco *repetir*. Essa dificuldade persistiu mesmo após ter sido lembrado a eles que os comandos cuja repetição fosse desejada deveriam ficar dentro do bloco de repetição. A Figura 34(a) mostra o programa montado pela dupla depois desses primeiros esclarecimentos e a Figura 34(b) o programa esperado.

Problemas semelhantes aos reportados acima foram observados na construção dos programas feitos pela dupla formada por Hilton (que também vinha apresentando uma boa evolução no entendimento dos conteúdos sendo abordados) e Ivana.

A última dupla, que nessa atividade ficou composta por Breno e Marcelo, dois alunos que vinham apresentando dificuldade no entendimento dos conteúdos, só conseguiu montar a sequência de comandos para piscar o LED azul e depois colocá-los dentro do bloco *repetir*, com o auxílio do professor. Uma vez isso feito, foi solicitado



Figura 34: Exemplo de problema de entendimento do uso do bloco *repetir*

a essa dupla que fizesse, ao lado do pedaço do programa construído para o LED azul piscar, a parte do programa para fazer o LED verde piscar, o que foi realizado sem dificuldade. Finalizada essa parte da tarefa, a dupla encaixou o bloco *repetir*, que continha os comandos para fazer o LED verde piscar, embaixo do bloco *repetir* que continha os comandos para fazer o LED azul piscar, o que demonstra um entendimento da estrutura sequencial do programa solicitado.

Apenas José conseguiu terminar as tarefas anteriores a tempo de tentar realizar a tarefa da palavra-código. Durante a realização desta atividade, mais uma vez, quando da intenção de programar um determinado LED para piscar um certo número de vezes, José usou apenas o comando para fazer o LED acender, ao invés de montar a sequência de comandos necessária para fazê-lo piscar (Figura 35(b)), como pode ser visto na Figura 35(a).



Figura 35: Não entendimento da sequência de comandos necessárias para fazer um LED piscar

Infelizmente não houve tempo para que José pudesse enviar o programa para a caixinha e observar o resultado na busca por erros de programação. Essa inconstância no entendimento das crianças, tanto do uso do bloco *repetir* quanto do efeito do uso do bloco *esperar*, tornou nítida a necessidade de um trabalho mais prolongado acerca desses conteúdos.

Cabe comentar aqui que, de modo geral, cada ação esperada de um atuador pode ser traduzida em um único comando, como por exemplo, girar o motor DC, parar o motor DC, mover o servo motor para tantos graus, acender o LED, apagar o LED. No entanto, a ação *piscar* é resultante de um conjunto de ações, ou seja, não é resultado imediato de um comando. Após alguns encontros, como será visto posteriormente, Daniel, Hilton e José passaram a demonstrar a habilidade de decompor a ação *piscar* em um conjunto de ações. No entanto, Breno, Ivana e Marcelo aparentemente não foram capazes de desenvolver essa habilidade até o término da oficina. Assim como no caso da construção mental de uma sequência de comandos que resultem em um fim esperado (ver capítulo 5.1.2.2), esta dificuldade pode estar

relacionada com a habilidade de se relacionar o todo com suas partes, bem como de realizar operações mentalmente, habilidades estas que amadurecem no período operatório concreto e deste modo podem não estar ainda suficientemente desenvolvidas em algumas crianças do terceiro e quarto anos.

Por fim, cabe também mencionar que, assim como acontecia nos encontros anteriores, em algumas duplas ou trios, uma das crianças assumia para si a tarefa da construção do programa, enquanto as outras acabavam adotando uma postura mais passiva. Essa dinâmica pode ter contribuído para a evidente divisão da turma em dois grupos, cada um deles com três crianças: um formado por Daniel, Hilton e José, que apresentavam uma clara evolução em relação ao entendimento dos temas sendo trabalhados nas aulas, e outro, formado por Breno, Ivana e Marcelo, que não conseguia acompanhar a evolução apresentada pelo primeiro grupo.

5.2.2 Segundo Encontro (Aula 5)

5.2.2.1 Atividades Realizadas

Neste encontro, diante das dificuldades apresentadas pelas crianças em relação ao uso do bloco *repetir* e do bloco *esperar*, decidiu-se, antes de aplicar novas tarefas de programação, trabalhar um pouco mais os conceitos de *repetição contada* e de *espera (delay)* por meio da brincadeira do robô humano.

Para o entendimento do uso do bloco *esperar* em conjunto com os blocos *acender LED* e *apagar LED*, o robô humano acendia e apagava uma lanterna (que representava o LED) no intervalo de tempo estipulado pelo programa. Esse exercício era executado lentamente com o objetivo de se fazer as crianças refletirem sobre o comportamento que cada comando produzia no robô humano e qual o resultado final observado: a lanterna piscar. Uma vez finalizada essa atividade, foi solicitado às crianças que construíssem uma série de programas que fizessem com que a caixinha piscasse os LEDs, como por exemplo: piscar o LED azul 1 vez, depois piscar o LED amarelo 1 vez e, por fim, piscar o LED vermelho 2 vezes, sem o uso do bloco *repetir*.

Para o entendimento do funcionamento da *estrutura de repetição contada*, ou seja, o uso do bloco *repetir*, foi realizada novamente a brincadeira do robô humano, que, desta vez, precisava dar um determinado número de passos (a cada passo o

robô humano movia-se uma “lajota” do assoalho da sala) para chegar até um objeto, pegá-lo e, depois, dar um determinado número de passos para levar o objeto até um novo local. As crianças liam em voz alta os comandos e o professor os executava. Esses programas, preparados pelo professor, eram apresentados pelo projetor no formato de blocos, como no exemplo da Figura 36, onde vê-se dois programas cuja execução possui o mesmo resultado, sendo que um utiliza o bloco *repetir* e o outro não.



Figura 36: Exemplos de programas executados pelo robô humano

Durante a brincadeira, as crianças eram levadas a observar que a execução do programa com o bloco *repetir* gerava, com o uso de um número menor de blocos, o mesmo resultado da execução do programa que não fazia uso do bloco *repetir*. Após esse trabalho, foi solicitado às crianças que fizessem alguns exercícios de piscar os LEDs com e sem o uso do bloco *repetir*.

5.2.2.2 Resultados

Ficou evidente, durante a construção dos primeiros programas, nos quais era solicitado que as crianças fizessem determinados LEDs piscarem um certo número de vezes sem o uso do bloco *repetir*, que elas continuavam apresentando dificuldade em relação ao entendimento de que a ação piscar o LED era composta por um conjunto de quatro ações. O não entendimento dessa composição ficou nítido porque

todas as crianças só conseguiram finalizar os exercícios propostos com o auxílio do professor, ainda que em todos esses exercícios fosse solicitado o mesmo tipo de ação: fazer os LEDs piscarem; variando apenas a cor dos LEDs, o intervalo entre as piscadas, o número de piscadas e a ordem em que deveriam piscar. Um exemplo dos equívocos apresentados encontra-se na Figura 37, onde o programa solicitado era: Fazer o LED azul piscar 1 vez e depois o LED amarelo piscar uma vez.

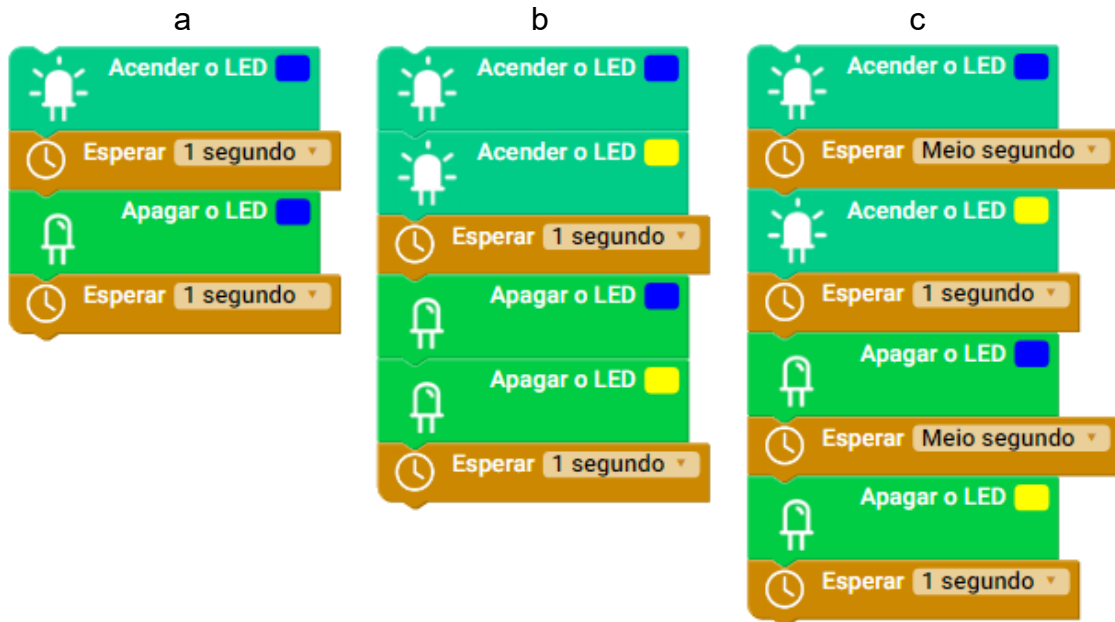


Figura 37: Dificuldade no entendimento dos comandos necessários para fazer um LED piscar

Partindo do exemplo do LED azul piscando, presente na Figura 37(a), ao ser solicitado que fizessem o LED amarelo piscar uma vez após o LED azul piscar, construíram o programa apresentado na Figura 37 (b). Após observarem o resultado da execução deste programa na caixinha e constatarem que não era o esperado, construíram o programa presente na Figura 37(c). O programa correto só foi construído com o auxílio do professor.

Apesar da dificuldade na realização dos exercícios propostos, pôde-se observar, durante a realização das tarefas, como já havia sido percebido nas atividades apresentados na seção sobre a estrutura sequencial, o potencial do aprendizado de programação com robótica no desenvolvimento da capacidade da depuração e correção de erros. Um exemplo disso pôde ser constatado quando Daniel, depois de montar o programa presente na Figura 38(a), ao enviá-lo para a caixinha mágica, observou que o LED amarelo permanecia aceso, e então exclamou: “Ih, o meu LED amarelo ficou aceso”. Ao ser então questionado sobre o porquê do

ocorrido, afirmou: “Porque eu não botei para ele apagar”. Diante da constatação do equívoco, ele corrigiu o programa (Figura 38(b)) e enviou novamente para a caixinha, observando agora o seu correto funcionamento.

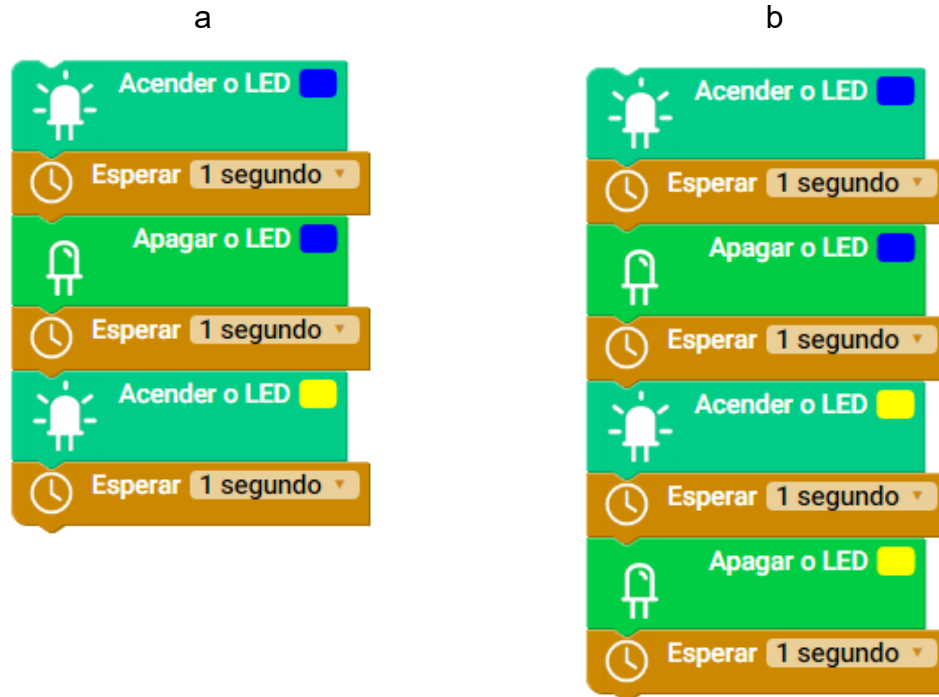


Figura 38: Programa incorreto corrigido com base na observação do resultado obtido a partir da sua execução

Durante a realização desses exercícios, os alunos eram levados a fazer uma série de reflexões acerca dos programas que estavam construindo, dos seus erros e acertos, e da relação entre os blocos que colocavam no programa e o resultado que observavam na caixinha. Com isso, no exercício seguinte, em que foi solicitado às crianças que fizessem o LED vermelho piscar duas vezes, Marcelo, Hilton e José montaram corretamente o programa, construindo duas sequências corretas de comandos para piscar o LED vermelho, uma abaixo da outra, como mostrado na Figura 39.

No exercício seguinte, realizado após a atividade do robô humano, através da qual foi novamente apresentado o funcionamento da *estrutura de repetição contada*, ao ser solicitado às crianças que fizessem o mesmo programa da atividade anterior (Figura 39), mas, com o uso do bloco *repetir*, dois dos três grupos concluíram a tarefa com sucesso, sem o auxílio do professor, colocando dentro do bloco *repetir* uma das sequências de comando para fazer o LED vermelho piscar e apagando a outra sequência, como mostrado na Figura 40.



Figura 39: Sequência de blocos para fazer o LED vermelho piscar 2 vezes sem o uso do bloco *repetir*.



Figura 40: Sequência de blocos para fazer o LED vermelho piscar 2 vezes com o uso do bloco *repetir*.

Importante relatar aqui, que foram as três crianças que vinham apresentando melhor aproveitamento (Daniel, Hilton e Davi) as responsáveis pela correta construção dos programas em seus grupos. Além disso, ficou evidente, na análise dos vídeos, o quanto elas conduziam os outros no desenvolvimento das tarefas, mesmo quando eram as outras crianças que estavam manipulando o ambiente de programação, a ponto de, muitas vezes, fazê-las desistirem de continuar no computador.

5.2.3 Terceiro Encontro (Aula 6)

5.2.3.1 Atividades Realizadas

Como muitas crianças permaneciam apresentando dificuldade em compreender a lógica utilizada para se fazer um LED piscar, desenvolveu-se nesse encontro uma atividade que objetivava permitir que as crianças observassem, com mais clareza, a diferença de resultado do uso dos comandos *acender LED* e *apagar LED* com e sem o uso do comando *esperar*. Para isso, foi solicitado que enviassem para a caixinha mágica programas utilizando os seguintes blocos em sequência:

- *acender LED* e *apagar LED*;
- *acender LED*, *esperar* e *apagar LED*;
- *acender LED*, *apagar LED* e *esperar*;
- *acender LED*, *esperar*, *apagar LED* e *esperar*.

Em seguida, foi solicitado que modificassem o parâmetro que indicava o tempo de espera nos comandos *esperar*, fazendo com que os LEDs ficassem mais ou menos tempo acesos ou apagados. A cada programa enviado era realizado um debate sobre os porquês dos resultados observados na caixinha mágica quando da execução de cada um dos programas.

Na sequência, para exercitar o entendimento do uso do bloco *repetir*, fez-se novamente a brincadeira do robô humano movimentando-se e pegando objetos na sala, mas, dessa vez, com a participação das crianças como robôs que executavam os comandos lidos do telão pelos demais colegas. Essa atividade foi também utilizada para trabalhar-se com os alunos o entendimento de que, com os mesmos comandos, mudando-se apenas a ordem em que eles são dispostos no programa e o número de vezes que serão executados, pode-se fazer com que um robô se mova e execute tarefas de diferentes maneiras. Terminada a atividade, foi solicitado às crianças que construíssem um programa com o uso do bloco *repetir* para piscar diferentes LEDs um determinado número de vezes.

5.2.3.2 Resultados

Após a atividade em que as crianças analisaram, via programação da caixinha, os comandos necessários para fazer o LED piscar, dos três grupos de crianças do terceiro e quarto anos, dois conseguiram montar sem problema o programa para fazer o LED azul e o LED amarelo piscarem em sequência, mostrando uma evolução em relação aos encontros anteriores. No entanto, uma das duplas, formada por Daniel e Diana, ainda apresentou dificuldade em entender a lógica utilizada para fazer o LED piscar, utilizando o comando *esperar* apenas após o comando *acender LED*.

Em relação aos exercícios realizados após a brincadeira do robô humano, nos quais foi solicitado que se usasse o bloco *repetir*, os grupos construíram corretamente as sequências para fazer os LEDs piscarem, e também não apresentaram dificuldade para usar o bloco *repetir*. O exemplo da Figura 41 mostra um desses programas. Uma vez construído pelos alunos o programa que fazia o LED azul piscar 2 vezes (com o uso do bloco *repetir*), e o LED amarelo piscar 1 vez (Figura 41(a)), foi solicitado que fizessem o LED amarelo também piscar 2 vezes. Diante dessa solicitação, José imediatamente pegou um novo bloco *repetir*, encaixou embaixo do bloco *repetir* já

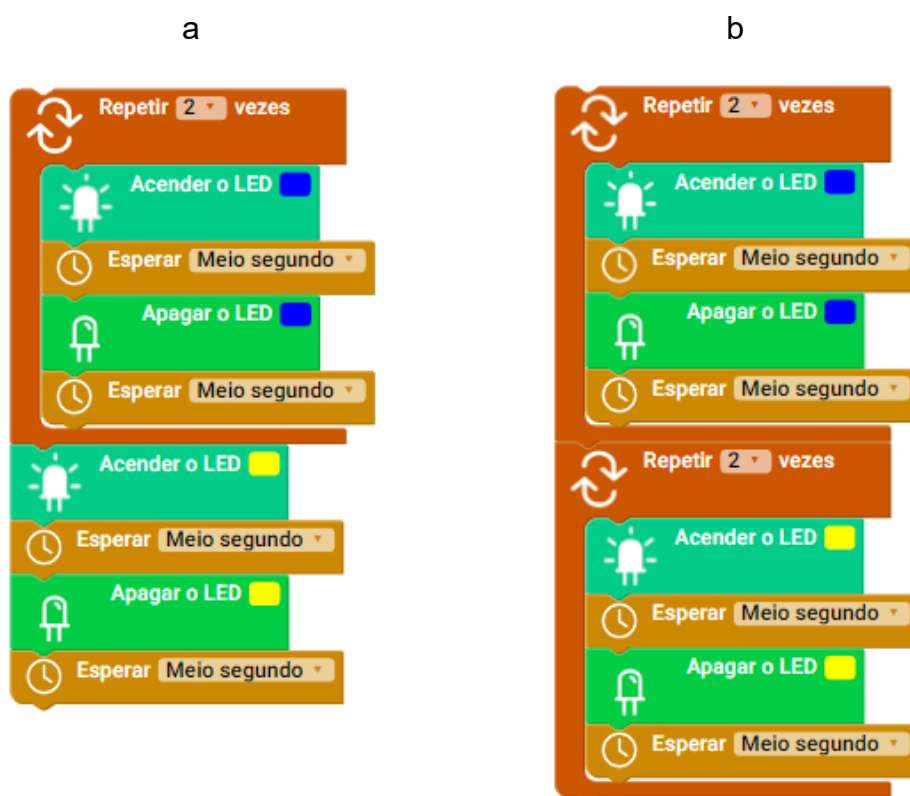


Figura 41: Uso do bloco *repetir*

presente no programa e movimentou, para dentro do novo bloco, os blocos utilizados para fazer o LED amarelo piscar (Figura 41(b)).

5.2.4 Quarto Encontro (Aula 7)

5.2.4.1 Atividades Realizadas

No início do quarto encontro, foi apresentado um protótipo de peixe robô de garrafa pet, que faz uso de sensores e cuja programação necessita da *estrutura de repetição condicional*. Essa atividade foi utilizada como motivação para as aulas futuras, sendo que, durante sua realização, as crianças foram informadas de que, continuando com o aprendizado, em breve estariam aptas a programá-lo. A descrição completa dessa atividade encontra-se na aula 7 do apêndice A⁹⁴.

Em seguida, através de um debate, foi feita uma revisão do uso dos blocos *acender LED*, *apagar LED*, *esperar* e *repetir*. Terminado o debate, as crianças, por meio de uma brincadeira, foram apresentadas ao display de 7 segmentos. Finalizada a brincadeira, foi solicitado aos alunos que construíssem um programa que fizesse os números de 0 a 9 aparecerem no display de 7 segmentos. O objetivo era observar se elas perceberiam a necessidade de se usar um bloco *esperar* entre cada comando para escrever no display de modo a permitir que os números fossem vistos.

Por fim, foi solicitado aos alunos que haviam finalizado a tarefa anterior, que fizessem um programa utilizando o LCD, o display de 7 segmentos e os LEDs. O programa deveria fazer algum LED piscar um certo número de vezes, sendo que no LCD deveria ser escrita a cor do LED e, no display de 7 segmentos, o número de vezes que esse LED iria piscar.

5.2.4.2 Resultados

Diante dos problemas que vinham sendo observados acerca do uso de apenas um computador por dupla ou trio, decidiu-se pela disponibilização, a partir desse encontro, de um computador por aluno.

⁹⁴ Disponível também em CD anexo à dissertação impressa e, futuramente, também no site do LIVRE (<http://www.nce.ufrj.br/ginape/livre/>)

Nesta aula estiveram presentes apenas quatro crianças: Hilton e Ivana, do terceiro ano, e Celso e Isadora, do segundo e primeiro anos. Hilton vinha desde o começo da oficina apresentando uma boa evolução, já Ivana é uma das alunas que mais vinha apresentando dificuldade, tendo ficado bastante alheia ao desenvolvimento das tarefas nos encontros anteriores.

Durante a revisão do uso dos blocos *acender LED*, *apagar LED*, *esperar* e *repetir*, Hilton explicou, com propriedade, como montar um programa para fazer o LED piscar, como fazer para algum dispositivo, por exemplo um motor, ficar funcionando por um determinado tempo, e que blocos utilizar para fazer um LED piscar várias vezes. Além disso, demonstrou completo entendimento de que apenas os comandos presentes dentro do bloco *repetir* eram repetidos. Ivana não se sentiu motivada a responder aos questionamentos, e talvez tenha faltado ao professor habilidade em promover essa motivação.

Na atividade na qual os alunos tinham que escrever os números de 1 a 9 no display de 7 segmentos, ao serem questionados sobre o que seria necessário fazer para que se pudesse ver os números sendo escritos no display, Hilton respondeu: “colocar o tempo”, o que demonstra o seu claro entendimento do efeito resultante do uso do comando *esperar*. Hilton montou o programa proposto com muita facilidade. Já Isabel não se sentiu inicialmente motivada a fazer o programa e só o fez com o auxílio do professor.

Hilton também montou com tranquilidade o programa seguinte, que fazia uso do LCD, do display de 7 segmentos e dos LEDs. Já Ivana, ao ser solicitada algum tempo depois que fizesse o mesmo o programa, exclamou: “- Tio, eu não entendi nada do que você falou”. Como a aula estava quase no fim, não havendo assim tempo hábil para que o professor pudesse auxiliá-la na tarefa, ela foi avisada de que o trabalho seria continuado em outra oportunidade.

A fala de Ivana torna bastante explícita a diferença de entendimento dos conteúdos existente entre ela e Hilton, diferença essa que pode ter alguma relação com as questões referentes ao compartilhamento dos computadores descritas anteriormente, dentre outros fatores não conhecidos (além daqueles relacionados à maturidade cognitiva).

5.2.5 Quinto Encontro (Aula 8)

5.2.5.1 Atividades Desenvolvidas

Como muitas crianças haviam faltado à aula anterior, a atividade para apresentação do peixe robô foi novamente aplicada. Pelo mesmo motivo, foi feita também, mais uma vez, a revisão do uso dos blocos *acender LED*, *apagar LED*, *esperar* e *repetir* e a apresentação do display de 7 segmentos.

Isto feito, foi solicitado às crianças que construíssem alguns programas para apresentar, no display de 7 segmentos, o número x, depois o número y e, por fim, o resultado da soma ou subtração desses 2 números, como por exemplo: 5,3,8.

Por último, foi desenvolvida uma atividade na qual as crianças observavam um vídeo onde era apresentada a caixinha mágica executando um determinado conjunto de ações e, a partir da observação do que a caixinha estava fazendo, os alunos deveriam construir um programa que fizesse a caixinha deles funcionar da maneira observada. Os vídeos mostravam a caixinha executando programas que faziam uso dos comandos e estruturas de programação previamente estudados. Um desses vídeos, por exemplo, mostrava o LED amarelo piscando 2 vezes, depois o LED vermelho piscando 5 vezes, logo em seguida o motor DC girando por 5 segundos e, por fim, o motor DC parando e o ciclo recomeçando.

5.2.5.2 Resultados

Durante a revisão do uso dos blocos *acender LED*, *apagar LED*, *esperar* e *repetir*, ao ser solicitado aos alunos que indicassem qual a sequência de comandos a ser utilizada para fazer um LED piscar, Marcelo, que de modo geral não vinha apresentando muito sucesso na solução dos exercícios propostos, e havia faltado à aula anterior, mostrou, com propriedade, qual a sequência de comandos necessária para fazer um LED piscar. O mesmo aluno, ao questionar-se a turma sobre o que precisaria ser feito para que o LED piscasse várias vezes, respondeu de imediato que bastava usar o bloco *repetir*. Os outros dois alunos com desempenho menos avançado, Bruno e Ivana, ficaram bastante desatentos durante a revisão, o que pode ser um indicativo de que os temas sendo abordados estavam além daquilo que haviam

conseguido absorver dos conteúdos trabalhados. Interessante também mencionar que, quando foi perguntado à turma se na construção de um programa fossem usados dois blocos para escrever no display de 7 segmentos, um em seguida do outro, seria possível ver o que estava sendo escrito, Isadora, do primeiro ano, respondeu: “Não. Tem que colocar o esperar”.

No exercício da apresentação de números no display de 7 segmentos, Bruno, Ivana e Marcelo, os 3 alunos que vinham apresentando mais dificuldade, só concluíram o programa com o auxílio do professor. Importante notar que Marcelo havia, durante a revisão, informado corretamente a sequência de comandos para fazer o LED piscar, mas, na hora de montar o programa com o display de 7 segmentos, não usou corretamente o bloco *esperar*, o que indica que possivelmente Marcelo tenha memorizado a sequência de comandos para fazer o LED piscar, mas, sem entender o porquê do uso desses comandos.

Hilton e José fizeram rapidamente o programa, no entanto, não colocaram o bloco *esperar* depois do bloco responsável por escrever o resultado da operação no display de 7 segmentos, corrigindo o problema após verem a execução do programa na caixinha mágica. Esse era um problema bastante recorrente. De modo geral, as crianças tinham o costume de não colocar o bloco *esperar* depois do último comando dos programas, percebendo e corrigindo o erro somente depois de ver o programa sendo executado na caixinha.

Neste caso, é possível que esse erro tenha ocorrido como resultado do seguinte raciocínio na hora da construção do programa para apresentar os números 5,3,8 (Figura 42): “escrevo o número 3, espero um segundo, escrevo o número 5 (que substitui o 3), espero mais um segundo e, por fim, escrevo o número 8 (que substitui o 5). Como não mandei escrever mais nenhum o número, o número 8 não será substituído e ficará escrito no display”. No entanto, como por padrão, todo programa no Arduino é construído dentro de uma estrutura de *loop infinito*, depois do número 8 ser escrito, o número 3 será novamente escrito, e assim por diante, o que invalida o raciocínio apresentado acima. O problema é que normalmente as crianças não lembravam que os comandos utilizados no programa seriam repetidos continuamente. Isso era facilmente observável pois, em muitas ocasiões, ao enviarem o programa para a caixinha e verem os comandos serem repetidos indefinidamente, modificavam

o código colocando o bloco “parar programa” ao final e o enviavam para a caixinha para observá-lo sendo executado uma única vez.



Figura 42: Programa para escrever a sequência de números 5,3,8 no display de 7 segmentos

A última atividade, na qual as crianças deveriam observar a caixinha mágica realizando um conjunto de ações e então construir um programa que fizesse a caixinha funcionar da maneira observada, trouxe resultados interessantes. Foi possível, por meio dela, constatar que algumas crianças já estavam conseguindo abstrair um programa de computador a partir da observação de eventos do mundo real, ou seja, construir mentalmente um algoritmo: uma sequência de ações que representasse o evento observado.

A primeira evidência nesse sentido pôde ser constatada quando, no primeiro exercício, após olharem a caixinha piscando o LED amarelo 3 vezes, o LED vermelho 5 vezes e, por fim, o Motor DC girando por 5 segundos e depois parando, ao ser solicitado que construíssem o programa que fizesse a caixinha funcionar daquela maneira, José perguntou: “Pode usar o bloco *repetir*?” Ou seja, o questionamento de José indica que ele traduziu as ações observadas em uma sequência de comandos e ainda visualizou duas formas de organizar esses comandos: por meio de uma *estrutura puramente sequencial* ou com o uso de uma *estrutura de repetição contada*.

Partindo para o desenvolvimento dos programas a partir da observação dos vídeos, José montou com facilidade o primeiro (cuja resposta esperada é apresentada na Figura 43), o que demonstra o entendimento do funcionamento dos comandos e estruturas que estava utilizando. O único erro cometido foi não colocar o bloco *parar motor DC* no final do programa. Mas, ao enviá-lo para a caixinha e observar sua

execução, vendo que o motor não parava, rapidamente se deu conta do equívoco e fez a correção necessária.

Na construção do segundo programa, no qual a caixinha, diferente do que acontecia no programa observado anteriormente, girava o motor DC entre as duas sequências de piscadas dos LEDs, José, habilmente, reorganizou os blocos do seu programa, posicionando os comandos responsáveis por fazer o motor girar por 5 segundos entre as duas estruturas de repetição responsáveis por fazer os LEDs piscarem.

Hilton também desenvolveu esses programas com muita habilidade. Já no exercício posterior, onde a caixinha rodava um programa que fazia uso também do display de 7 segmentos, Hilton cometeu alguns equívocos, mas, os corrigiu a partir de questionamentos feitos pelo professor. Daniel, um dos alunos que também vinha apresentando desempenho mais avançado, não estava presente nesse encontro.



Figura 43: Programa resultante da observação do funcionamento da caixinha mágica

Já Bruno, Marcelo e Ivana, só completaram as tarefas com a ajuda do professor, que teve que auxiliá-los na construção de todas as partes de cada um dos programas, sendo que, tanto Bruno quanto Ivana, além de constantemente solicitarem ajuda, “colavam” dos colegas ao lado (José e Hilton). No entanto, apesar das dificuldades, Ivana e Marcelo, com a disponibilização de um computador para cada criança, mantiveram-se bastante empenhados em concluir os exercícios propostos, o que não era comum nas aulas anteriores nas quais havia apenas um computador por grupo. Bruno manteve-se tão disperso quanto nas outras aulas.

A habilidade apresentada pelos dois alunos com melhor aproveitamento, Hilton e José, de construírem, com certa facilidade, um programa de computador a partir da observação do funcionamento da caixinha mágica, indica que os mesmos estão fazendo uso e, por conseguinte, exercitando a capacidade de abstração. Já a correção dos programas a partir da observação da sua execução na caixinha, como já verificado em atividades anteriores, demonstra o exercício da habilidade de detectar e corrigir erros. Por fim, a habilidade demonstrada por esses alunos, de reorganizar corretamente os comandos presentes no programa para que este refletisse as mudanças observadas na ordem das ações realizadas pela caixinha mágica, aponta o aprendizado de noções algorítmicas de fluxo de controle.

5.3 ESTRUTURA DE REPETIÇÃO CONDICIONAL

Como no caso da *estrutura de repetição contada*, o número necessário de aulas para o estabelecimento de um entendimento satisfatório do uso da *estrutura de repetição condicional* foi maior do que o inicialmente planejado. Foram assim utilizados 5 encontros para se trabalhar esta estrutura em conjunto com os conteúdos anteriormente trabalhados.

5.3.1 Primeiro Encontro (Aula 9)

5.3.1.1 Atividades Desenvolvidas

Como primeira atividade, após uma apresentação “formal” do LCD, foi solicitado às crianças que construíssem, no ambiente de programação, dois programas a partir de algoritmos bastante simples, como por exemplo:

Escrever o nome na linha 1 do LCD

Escrever o sobrenome na linha 2 do LCD

Limpar o LCD

O objetivo destes exercícios era observar se as crianças perceberiam a necessidade do uso do comando *esperar* depois dos comandos *escrever no LCD* e *limpar display* para que os textos pudessem ser visualizados.

Em seguida, para a apresentação da *estrutura de repetição condicional*, foi feita a brincadeira do robô humano, na qual as crianças deveriam seguir um programa em blocos exibido pelo projetor (Figura 44). Finalizada a brincadeira, foi feita uma reflexão acerca do funcionamento do bloco *enquanto* e estabelecido um paralelo entre o seu funcionamento e o funcionamento do bloco *repetir*.



Figura 44: Programa para o robô humano utilizando o bloco *enquanto*

Na sequência, após o bloco *enquanto* e o bloco *do sensor de luz* terem sido apresentados, o professor montou com as crianças o programa presente na Figura 45, inserindo os blocos, passo a passo, e explicando o efeito esperado de cada um dos comandos utilizados. Uma vez o programa finalizado, as crianças testaram o seu funcionamento, observando as mudanças de comportamento apresentadas pela

caixinha quando elas, com o auxílio de lanternas, aumentavam ou diminuíam a incidência de luminosidade sobre o sensor de luz.

Por fim, o professor pediu às crianças que substituíssem o bloco *do sensor de luz* pelo bloco *do sensor de temperatura* como parâmetro do bloco *enquanto*. Feita a substituição, as crianças enviaram o novo programa para a caixinha, sendo solicitado então que utilizassem a lanterna novamente e observassem o que aconteceria. Visto que o funcionamento da caixinha não se alterava, uma vez que a condição imposta havia sido modificada, foi solicitado às crianças que segurassem o sensor de temperatura com os dedos, explicando-se a elas que a temperatura dos seus corpos, que era mais alta que a do ambiente, seria percebida pelo sensor. O objetivo desse teste era ajudar as crianças a perceberem que eram os sensores os responsáveis por “disparar” a repetição dos comandos presentes dentro do bloco *enquanto*, e também, que diferentes sensores “respondem” a diferentes modificações no ambiente.

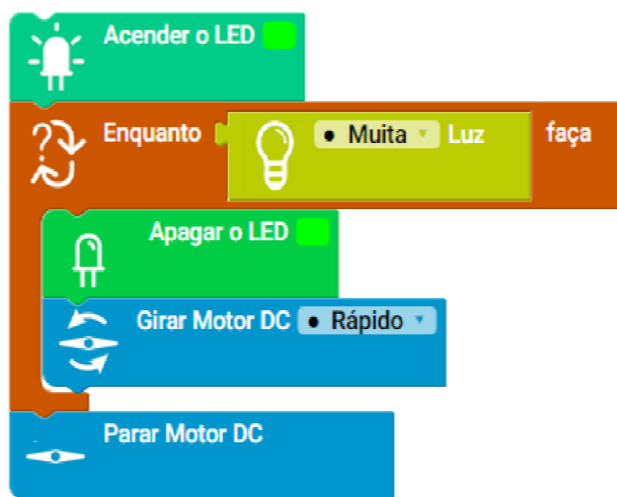


Figura 45: Programa exemplo utilizando o bloco *enquanto*

5.3.1.2 Resultados

Na tarefa na qual as crianças deveriam escrever textos nas duas linhas do LCD, José foi o único aluno que, por conta própria, lembrou de utilizar o bloco *esperar*, esquecendo apenas daquele que deveria ter sido colocado no final do programa (Ver Figura 46), um tipo de erro bastante recorrente entre os alunos, como já mencionado na seção 5.2.5.2.

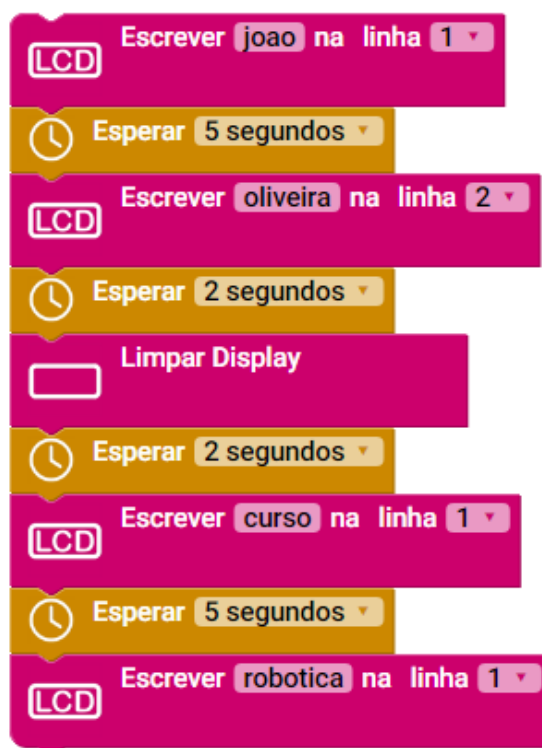


Figura 46: Programa com LCD faltando bloco *esperar*

Daniel construiu seu programa apenas com a sequência dos comandos apresentados no enunciado, usando o bloco *esperar* somente quando o professor provocou os alunos mencionando que algumas crianças haviam utilizado um outro bloco no programa além do bloco do LCD. E, depois do programa corrigido, assim como José, Daniel esqueceu de colocar o bloco *esperar* depois do último comando. Hilton, o outro aluno que vinha apresentando um rendimento mais expressivo, não estava presente nesse encontro.

Bruno fez seu programa olhando o de José. Marcelo e Ivana precisaram de indicações do professor, passo a passo, para construírem seus programas. Ficou nítido que Ivana não estava conseguindo construir um entendimento acerca da função dos blocos utilizados no programa e dos efeitos da sua execução na caixinha mágica, quando ao enviar para a caixinha o programa construído, que não possuía o bloco *esperar* após o bloco *limpar display*, não soube explicar ao professor o porquê de não se conseguir ver o LCD limpo.

Embora as crianças tenham demonstrado dificuldade em desenvolver o programa proposto, pôde-se constatar, novamente, que ter um computador à disposição de cada aluno fez com que Ivana e Marcelo ficassem mais empenhados

em realizar as tarefas apresentadas. Bruno continuou disperso, não apresentando interesse em desenvolver os programas propostos.

5.3.2 Segundo Encontro (Aula 10)

5.3.2.1 Atividades Desenvolvidas

Neste encontro foram novamente realizadas atividades nas quais as crianças construíam programas a partir da observação de vídeos que mostravam a caixinha mágica apresentando um determinado funcionamento. O primeiro vídeo apresentava a caixinha executando um programa que sugeria o uso de uma *estrutura de repetição contada*. O vídeo seguinte mostrava a caixinha mudando de comportamento quando uma lanterna iluminava o seu sensor de luz, o que sugeria o uso da *estrutura de repetição condicional*.

Finalizada a atividade com os vídeos, o peixe robô foi novamente apresentado às crianças, sendo solicitado a elas que observassem seu funcionamento e, assim como na atividade com os vídeos, criassem um programa que fizesse o peixe funcionar da maneira observada.

5.3.2.2 Resultados

No primeiro exercício com os vídeos, no qual via-se a caixinha realizando a repetição contínua da seguinte sequência de ações: a palavra “azul” aparecendo no LCD, o LCD limpo, o número 4 aparecendo no display de 7 segmentos, o display de 7 segmentos apagado, o LED azul piscando 4 vezes; Hilton, que havia faltado a aula anterior, construiu o programa muito rapidamente, utilizando corretamente o bloco repetir e posicionando o bloco esperar nos locais onde ele era necessário. No entanto, esqueceu de limpar o LCD e apagar o display de 7 segmentos (Figura 47(a)).

Esse era um detalhe relativamente sutil no vídeo, e passou despercebido para Hilton, que o corrigiu (Figura 47(b)) assim que o professor apontou a diferença entre o que a caixinha mágica estava fazendo (em resposta ao programa por ele construído) e o que estava sendo exibido no vídeo.

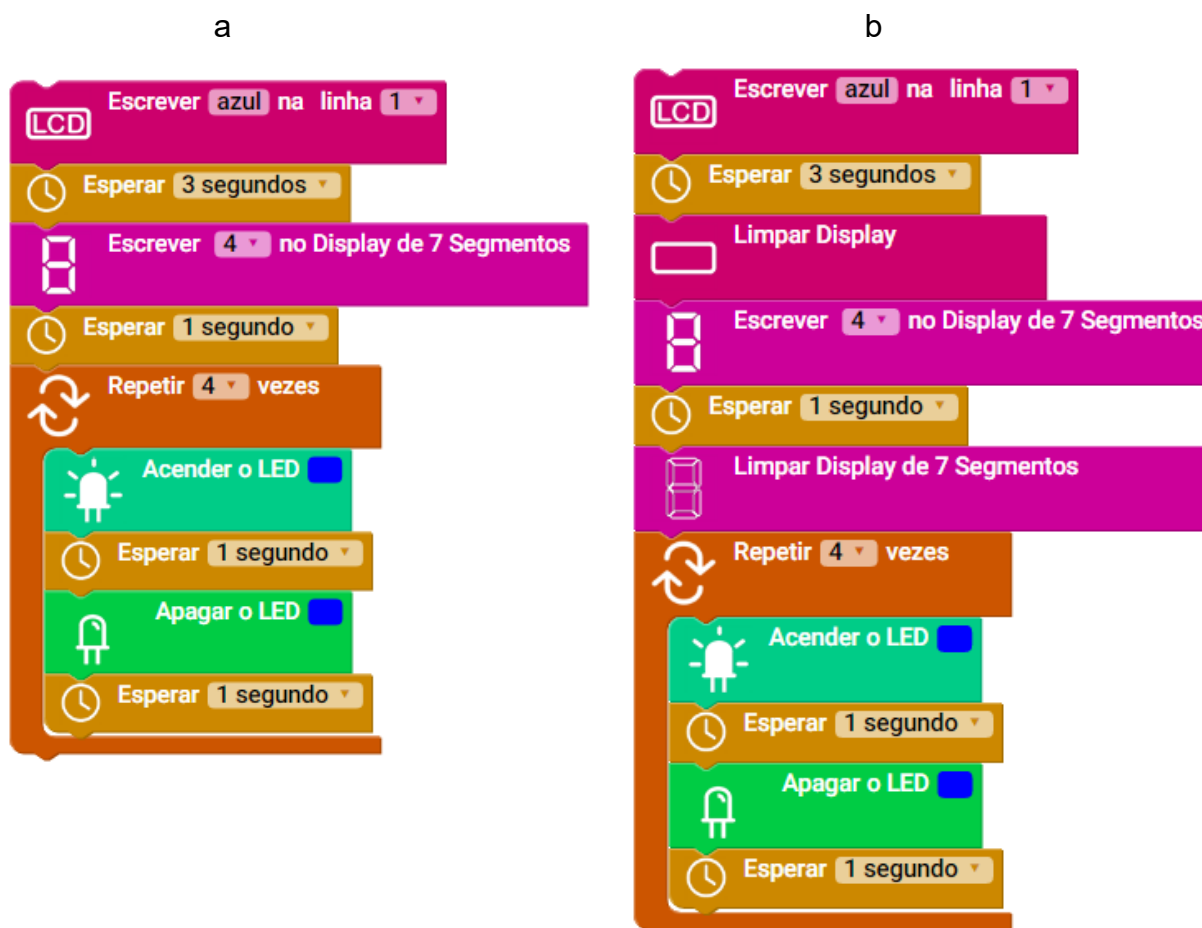


Figura 47: Programa montando por Hilton

Daniel montou o programa mais lentamente, assistindo atentamente ao vídeo e buscando reproduzir aquilo que observava, o que fez com que utilizasse corretamente os blocos para limpar os displays. No entanto, embora tenha utilizado corretamente o bloco *repetir* e os blocos *esperar*, não montou corretamente a sequência de comandos para fazer o LED azul piscar (Figura 48), e cometeu um erro quanto ao número de repetições.

José, que também vinha apresentando bons resultados, estava ausente nesse encontro.

Os resultados apresentados por Bruno mostram nitidamente que ele não havia desenvolvido um entendimento acerca dos conteúdos trabalhados. As Figuras 49 e 50 mostram uma sequência de programas por ele construídos na tentativa de fazer o LED piscar 4 vezes. A cada tentativa Bruno enviava o programa para caixinha mágica e, ao constatar que o resultado obtido não era o esperado, fazia as modificações que julgava necessárias, não conseguindo, porém, chegar a uma sequência correta de comandos e tampouco utilizar o bloco *repetir* conforme o esperado.



Figura 48: Programa de Daniel

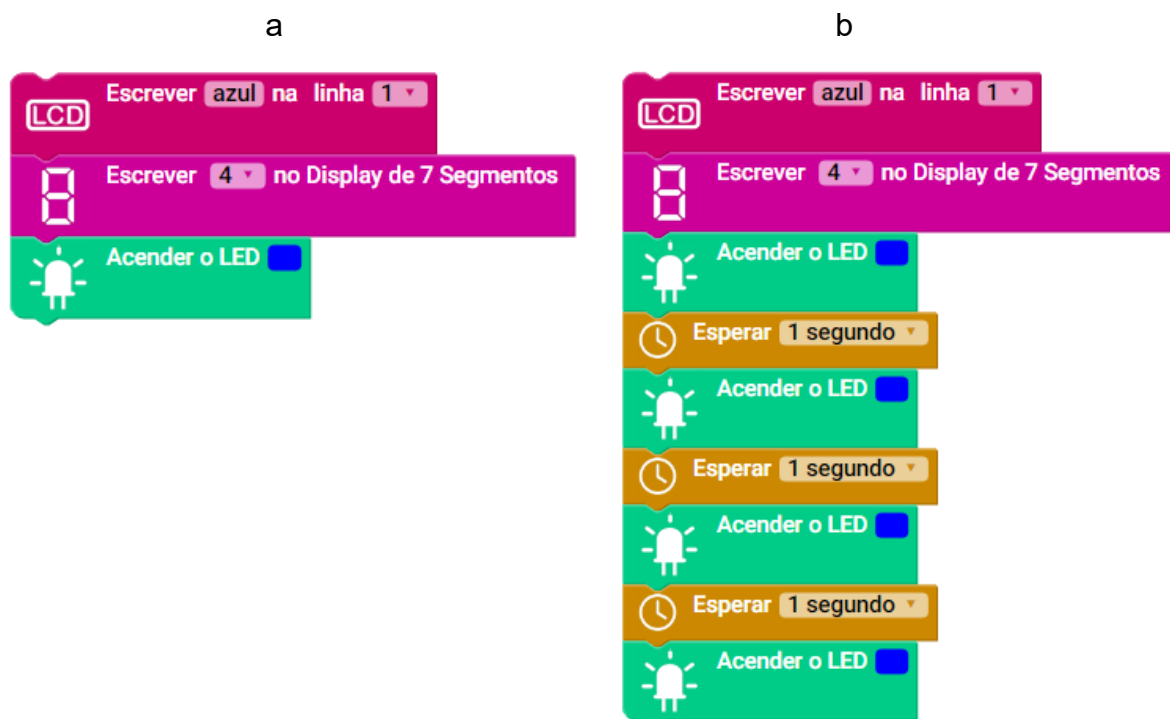


Figura 49: Primeira e segunda tentativas de Bruno para fazer o LED piscar

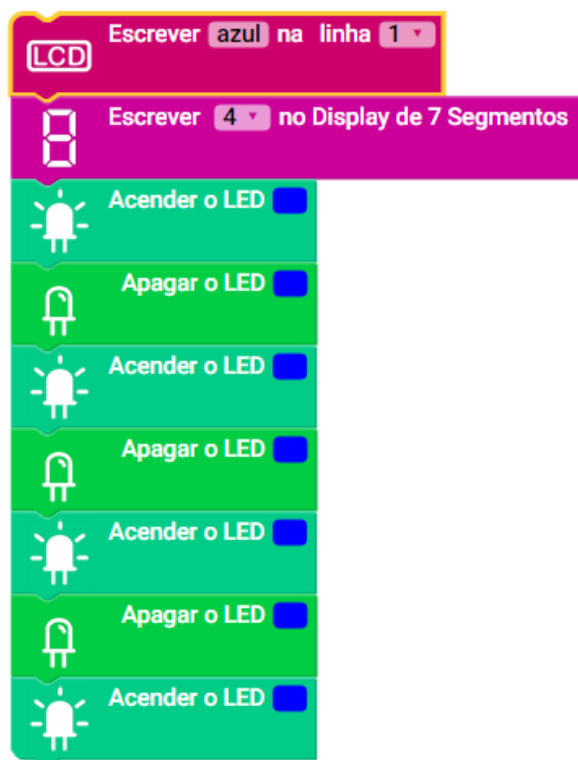


Figura 50: Última tentativa de Bruno para fazer o LED piscar

Ivana montou o programa apresentado na Figura 51(a), colocando todos os comandos dentro do bloco *repetir* e esquecendo de mudar o parâmetro que indica o número de repetições. No entanto, uma vez tendo sido explicado a Ivana que, do modo como o programa estava construído, todos os comandos seriam repetidos 2 vezes, e de ter sido solicitado a ela que modificasse a programação de modo que somente os comandos para piscar o LED fossem repetidos, Ivana retirou os comandos para escrita no display do bloco *repetir* e os posicionou no local adequado (Figura 51 (b)). Importante notar também que Ivana utilizou corretamente o bloco *esperar* depois dos displays e na sequência de comandos para piscar o LED (esquecendo apenas do último). No programa corrigido de Ivana ficou faltando também o bloco para limpar o display de 7 segmentos. A performance de Ivana nesta atividade indicaria, aparentemente, uma compreensão do uso dos blocos *esperar* e *repetir*, que ela não havia apresentado até então. No entanto, sua performance em atividades posteriores nos apresenta uma realidade diferente, o que pode ser um indicativo de que ela teria olhado o programa dos colegas nesta atividade (como veio a fazer em outras ocasiões). Entretanto, não é possível fazer, com base nas imagens disponíveis nas filmagens, qualquer afirmação nesse sentido.

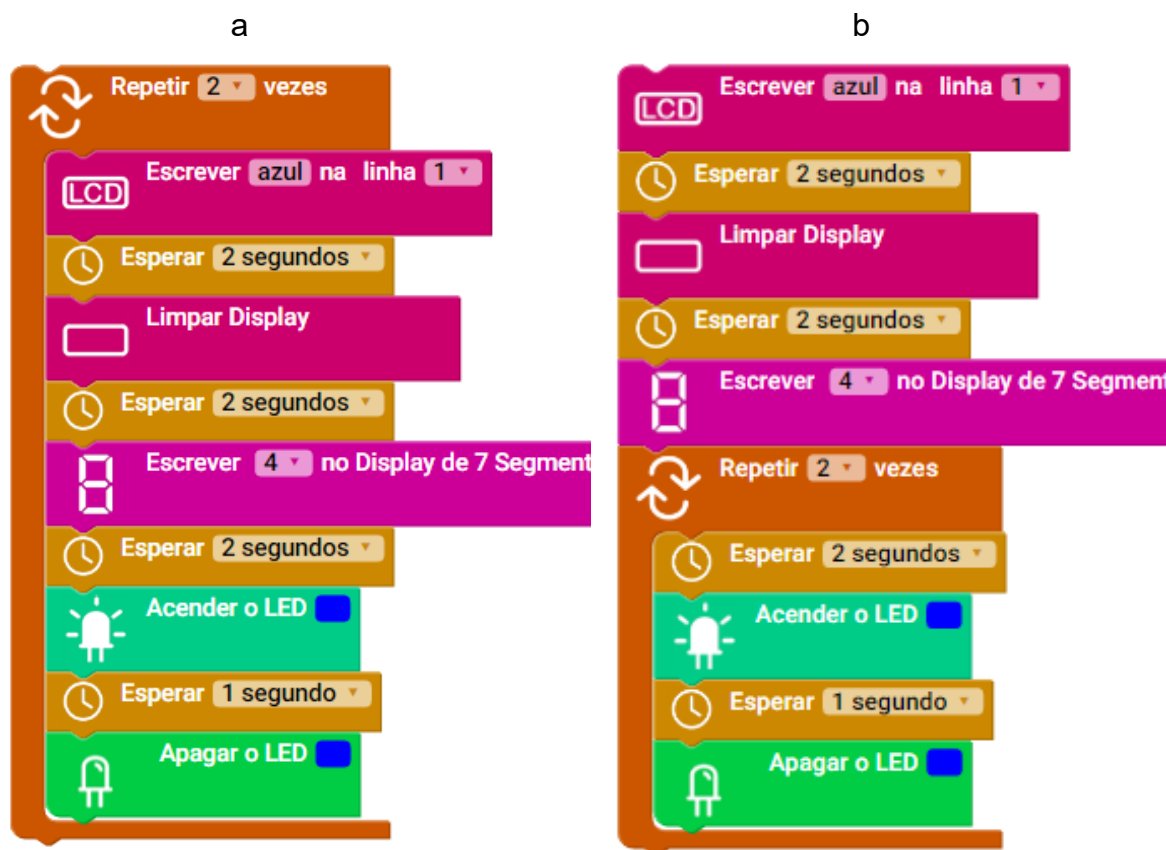


Figura 51: Programa de Ivana

Marcelo copiou o programa de Hilton, não sabendo, quando questionado, explicar a lógica do programa por ele apresentado.

No segundo exercício com os vídeos, onde foi utilizado um sensor de luz como condição para a mudança de “comportamento” da caixinha mágica, as crianças apresentaram dificuldade em determinar a lógica do funcionamento observado. Essa dificuldade aparentemente adivinha do fato de que, dependendo do tempo em que a lanterna permanecia iluminando o sensor de luz, a caixinha apresentava diferentes números de execuções das “ações” internas ao laço de repetição, o que diferia do resultado “constante” observado quando do uso da *estrutura de repetição contada*. Assim, para chegar ao programa esperado (Figura 52), ou a alguma variante dele, todas as crianças precisaram de uma permanente orientação realizada por meio de questionamentos lançados pelo professor.

Para facilitar o processo, foi feita uma sugestão inicial, de que construíssem o programa em duas partes. Primeiro a parte responsável pelo funcionamento observado quando o sensor de luz não estava sendo iluminado e, depois, a parte correspondente às ações realizadas pela caixinha enquanto a lanterna estava

iluminando o sensor. A dinâmica utilizada nesta atividade, ou seja, da divisão do problema em partes menores, mostra que é possível realizar-se, mesmo no desenvolvimento de programas muito simples, um trabalho inicial acerca da “decomposição estruturada de problemas”, uma habilidade também característica do Pensamento Computacional.



Figura 52: Programa a ser construído na segunda atividade

Daniel mostrou compreensão das decisões que estava tomando para o desenvolvimento do programa. Além disso, nos momentos em que era alertado pelo professor de que algo não estava em conformidade, rapidamente encontrava o problema e o solucionava. Por exemplo, ao ser questionado à turma onde deveriam ser colocados os comandos responsáveis pelas ações que eram executadas quando a lanterna iluminava o sensor de Luz, Daniel respondeu: “Enquanto!”.

Outro momento interessante de ser destacado refere-se a um dos envios do programa para caixinha, quando Daniel ainda não havia colocado o bloco *apagar LED* após o bloco *enquanto*. Ao deixar de iluminar o sensor de luz com a lanterna, Daniel

verificou que o LED vermelho permanecia aceso e então exclamou: “Eita, a luz vermelha ficou acesa”; o que deu início ao seguinte diálogo:

Professor: Por que ela ficou sempre acesa? Por que você acendeu ela. Você mandou apagar em algum momento?
 Daniel: Não.
 Professor: Então. Quando sair disso aqui (bloco *enquanto*), você quer que aconteça o que?
 Daniel: Apagar a luz.
 Professor: Então você vai botar o apagar a luz onde?
 Daniel: Tem que botar do lado de fora.

Hilton havia faltado a aula anterior, na qual a *estrutura de repetição condicional* foi apresentada. No entanto, conseguiu assimilar bem as explicações feitas pelo professor e desenvolveu o programa demonstrando bastante domínio em relação às suas escolhas. Assim como Daniel, ele também esqueceu de colocar o bloco *apagar LED* após o bloco *enquanto*. No entanto, diferente do que aconteceu com colega, ao perceber o problema e ser questionado sobre onde posicionar o bloco faltante, ele o colocou dentro do bloco *enquanto*, vindo a posicioná-lo no local correto apenas a partir de questionamentos feitos pelo professor. Tanto Hilton quanto Daniel não utilizaram o bloco *limpar display* em seus programas, o que é perfeitamente compreensível, uma vez que não havia como saberem, a priori, que o Arduino não limpa o LCD para escrever um novo texto, e sim, sobrescreve um texto ao outro.

Bruno tentou desenvolver o programa, mas, não conseguia produzir um resultado coerente sozinho, precisando da orientação do professor para inserir cada um dos comandos necessários. Marcelo construiu seu programa olhando o programa de Hilton, e Ivana, o programa de Marcelo.

Na última atividade, onde os alunos deveriam desenvolver um programa (Figura 53) que fizesse o peixe robô funcionar conforme o observado, Hilton montou o programa rapidamente, esquecendo apenas de apagar o LED vermelho e programando de forma equivocada o LED RGB. O erro na programação do LED RGB é bastante compreensível por tratar-se de um dispositivo com o qual os alunos ainda não haviam trabalhado e sobre o qual não havia sido feita uma apresentação detalhada (tanto do dispositivo quanto do bloco utilizado para controlá-lo). Em relação ao LED vermelho, Hilton corrigiu o programa assim que foi questionado sobre o porquê de o mesmo permanecer aceso, inserindo o bloco *apagar LED* no local adequado.

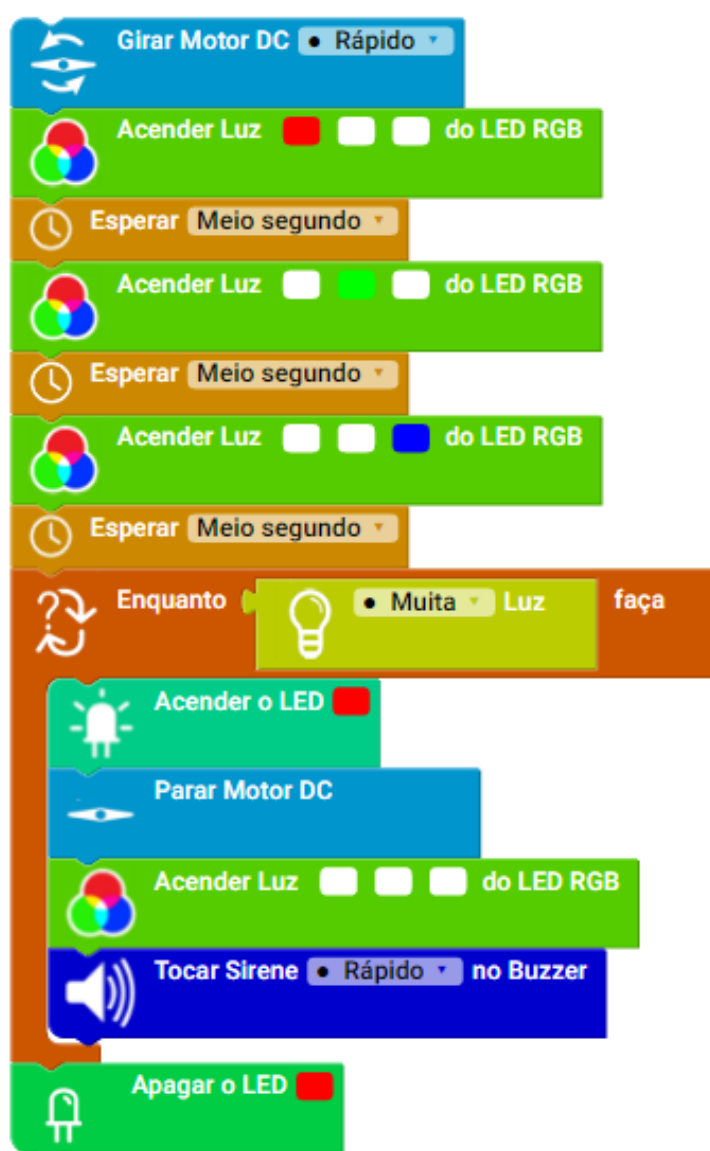


Figura 53: Programa do peixe robô

Daniel estava desatento, possivelmente devido ao cansaço, o que fez com que não se concentrasse na tentativa de solucionar o problema e ocupasse a maior parte do seu tempo acessando jogos na internet. Com isso, cometeu muitos equívocos na construção do programa, mas, corrigiu todos a partir da comparação do funcionamento do peixe robô com os resultados que estava obtendo ao enviar o programa para a caixinha mágica. Assim como Hilton, Daniel também não programou corretamente o LED RGB.

Ivana e Marcelo empenharam-se em resolver o problema, mas, só conseguiram fazê-lo seguindo exemplos mostrados pelo professor ou olhando os programas feitos pelos colegas. Ivana, assim como Hilton, não utilizou em seu programa o bloco apagar

LED. No entanto, diferente do que aconteceu com o colega, ela não foi capaz de determinar, ao observar a execução do programa na caixinha mágica, o porquê do LED permanecer aceso, o que evidencia o seu não entendimento acerca do código construído. Bruno demonstrou pouco interesse em realizar a tarefa, atuando somente quando o professor procurava saber o que ele havia desenvolvido.

Pode-se verificar, diante dos resultados apresentados, que Hilton e Daniel demonstram um bom entendimento dos conteúdos até então trabalhados. Quanto a Bruno, Ivana e Marcelo, os resultados confirmam uma defasagem significativa do entendimento deles em relação aos colegas, não tendo sido constatada em relação a eles qualquer evidência de compreensão do funcionamento das estruturas de repetição, tanto contada quanto condicional.

Interessante notar também, em relação à dinâmica adotada pelas crianças para observar a correção dos programas desenvolvidos, que elas decidiram, por conta própria, enviar para caixinha mágica o programa destinado ao peixe robô, o que indica a capacidade delas em generalizar o uso dos dispositivos. Ou seja, entender que o comando que faz o motor DC girar na caixinha mágica é o mesmo que faz o motor DC do rabo do peixe girar, que o sensor de luz presente no peixe funciona da mesma maneira que o sensor de luz presente na caixinha, e assim por diante, o que possibilita que se possa observar na caixinha o funcionamento do programa destinado ao peixe robô.

5.3.3 Terceiro Encontro (Aula 11)

5.3.3.1 Atividades Desenvolvidas

No terceiro encontro as crianças trabalharam com a programação de pequenos morcegos robô de garrafa-pet (ver capítulo 3.1). O morcego robô havia sido parcialmente construído pelo professor e coube às crianças a tarefa de finalizá-los. Os morcegos possuíam um sensor de distância no lugar dos olhos, uma hélice no topo da cabeça (movida por um motor DC) e um LED vermelho no "nariz".

Uma vez as crianças tendo finalizado a montagem dos robôs, o professor mostrou seu próprio morcego robô em funcionamento e pediu para que elas observassem o seu comportamento. As crianças puderam então, verificar que o

morcego ficava constantemente piscando o nariz vermelho e, quando a mão delas se aproximava dos olhos do robô (sensor de distância) e durante o tempo que lá se mantivesse, a hélice do topo da cabeça do robô girava rápido e o LED vermelho mantinha-se aceso, sem piscar. Afastando-se a mão, o motor parava de girar e o LED vermelho voltava a piscar⁹⁵.

Finalizada a etapa de observação, foi realizado um debate acerca do comportamento apresentado pelo morcego, sendo solicitado então às crianças que construíssem um programa que fizesse os morcegos delas funcionarem exatamente como o do professor.

5.3.3.2 Resultados

Daniel, Hilton e José cometeram pequenos erros na construção do programa para o morcego robô, exibido na Figura 54. José, por exemplo, não montou corretamente a sequência de comandos para fazer o LED vermelho piscar, mas, utilizou corretamente a *estrutura de repetição condicional*, posicionando todos os blocos nos locais adequados. Além disso, quando solicitado pelo professor que explicasse o programa que havia construído (já com a programação dos LEDs corrigida), José demonstrou perfeito entendimento do seu funcionamento.

Daniel apresentou, novamente, dificuldade em construir a lógica para fazer o LED piscar, utilizando para este fim, apenas o comando para acender o LED. A sequência correta de comandos para piscar o LED só foi construída a partir de questionamentos feitos pelo professor. Entretanto, assim como José, Daniel utilizou corretamente a *estrutura de repetição condicional*, posicionando adequadamente os blocos *acender LED* e *girar motor DC* dentro do bloco *enquanto* e o bloco *parar motor DC* fora da estrutura de repetição.

Hilton montou o programa muito habilmente e sem a necessidade da ajuda do professor. Esqueceu apenas de colocar o bloco *parar motor DC* após o bloco *enquanto*. No entanto, ao enviar o programa para o morcego robô e observar que ele não estava funcionando como esperado, imediatamente localizou o problema e fez a correção necessária. Além disso, foi o único aluno que, por conta própria, selecionou o operador de comparação adequado no bloco *do sensor de distância*. Cabe

⁹⁵ A narrativa que contextualiza o morcego robô encontra-se no apêndice A, aula 11.

menção que essa foi a primeira vez que os alunos utilizaram o bloco *do sensor de distância*, onde os operadores de comparação, *maior que* e *menor que* são utilizados, além do que, por um lapso do professor, esses conceitos não haviam sido trabalhados em aula.

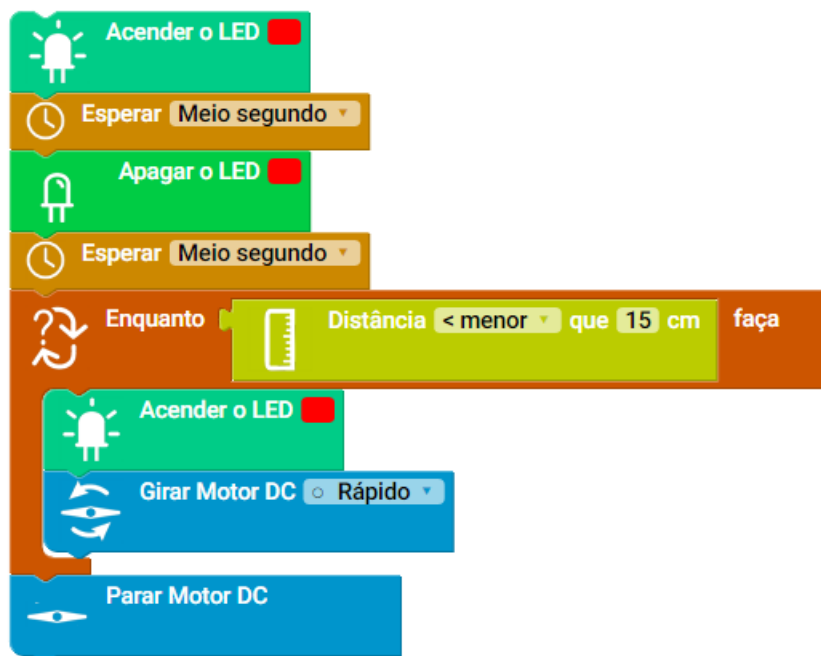


Figura 54: Programa de controle do morcego robô

Ivana demonstrou não fazer distinção entre os comandos que eram realizados dentro ou fora da *estrutura de repetição condicional*. Ela utilizou os blocos corretos, mas de forma equivocada. A única coerência apresentada no programa foi o fato de Ivana ter posicionado os blocos na ordem em que os dispositivos eram ativados quando da observação do funcionamento do morcego (Figura 55). Ivana só chegou ao resultado esperado com a indicação, passo a passo, do que deveria ser corrigido no programa.

Bruno e Marcelo, mais uma vez, construíram os programas a partir da observação dos programas dos colegas. O Professor buscou, por meio de questionamentos e explicações, motivá-los a entender o que precisava ser feito, no entanto, eles não demonstraram interesse em tentar montar o programa por conta própria.



Figura 55: Programa de Ivana para o morcego robô

Os dados observados continuam a evidenciar um entendimento satisfatório dos alunos Daniel, Hilton e José em relação aos conteúdos abordados, sendo que Hilton, como já vinha sendo notado, apresenta um maior domínio do que os colegas, cometendo menos erros na construção inicial dos programas. Fica bastante nítido também que Daniel e José, quando da intenção de fazer com que um LED pisque, continuam, em muitas ocasiões, não decompondo essa ação no conjunto de ações necessárias para este fim.

Bruno e Marcelo, por sua vez, não apresentaram evolução em relação aos conteúdos trabalhados, o que pode ter contribuído para o desinteresse de ambos em relação às atividades desenvolvidas. Por outro lado, Ivana, apesar de apresentar dificuldades semelhantes às dos colegas, manteve-se empenhada em resolver os problemas propostos.

Na busca por minimizar as dificuldades apresentadas por esses alunos, o professor procurou acompanhá-los de maneira bastante próxima, explicando constantemente a lógica dos programas construídos e a função de cada um dos blocos neles utilizados. Entretanto, até então, essa estratégia não havia apresentado resultados positivos. Uma outra alternativa supostamente viável seria propor a esses alunos desafios menos complexos, no entanto, essa estratégia não foi adotada pois as crianças poderiam sentir-se preteridas ao ser tirado delas a possibilidade de, assim como os colegas, programar os robôs utilizados em aula.

5.3.4 Quarto Encontro (Aula 12)

5.3.4.1 Atividades Desenvolvidas

No início do quarto encontro foi feita uma revisão do funcionamento da *estrutura de repetição condicional*. Para tanto, o professor apresentou, com o projetor, um programa exemplo com alguns comandos internos e externos a um bloco *enquanto* e, com base em questionamentos, buscou fazer com que as crianças explicassem qual seria o resultado observado quando da execução deste programa.

Finalizada a revisão, as crianças trabalharam novamente com o morcego robô, sendo solicitado a elas que criassem programas que fizessem os seus morcegos funcionarem do modo que desejassem. Foi feita uma sugestão inicial de que programassem a hélice presente na cabeça do morcego para girar em diferentes velocidades de acordo com a presença ou não de algum objeto próximo aos “olhos” do morcego.

O último trabalho desenvolvido fez uso do carrinho robô (Ver capítulo 3.1), que estava equipado com dois sensores, um de distância e outro de luz, e caracterizado como robô vampiro, com duas asas de morcego⁹⁶. Os alunos deveriam observar o comportamento do carrinho e programá-lo para se comportar conforme o observado. Finalizada esta primeira etapa, as crianças que desejassem poderiam então fazer um novo programa onde elas mesmas decidiriam que comportamentos seriam assumidos pelo carrinho a partir da leitura do estado dos sensores.

5.3.4.2 Resultados

Durante a revisão do uso do bloco *enquanto*, Daniel observou que no programa apresentado pelo professor estava-se utilizando o comando para girar o motor DC devagar tanto dentro quanto fora do laço de repetição, e comentou:

Daniel: Ô tio, ali está errado!

Professor: O que é que está errado?

Daniel: Por que ele está: girar o motor DC devagar. Aí, enquanto temperatura alta, girar o motor DC devagar de novo?

⁹⁶ A narrativa que contextualiza o carrinho robô vampiro encontra-se no apêndice A, aula 12.

Essa percepção de Daniel, de que não fazia sentido, no programa apresentado, dar a mesma ordem fora e dentro da *estrutura de repetição condicional*, indica o seu entendimento do funcionamento desta estrutura e de que a mesma é utilizada, dentro do universo dos conteúdos trabalhados na oficina, para que se faça uma modificação no comportamento do robô diante de uma determinada condição.

Também durante a revisão, quando o professor perguntou por quanto tempo o motor DC permaneceria girando rápido caso o sensor de temperatura (Figura 56) fosse trocado por um sensor de luz e uma lanterna o iluminasse por 10 segundos, José respondeu: “Dez segundos”.

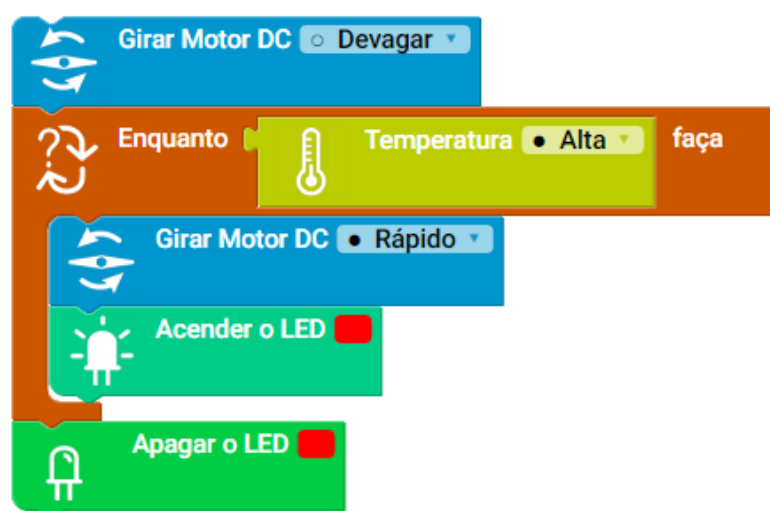


Figura 56: Programa utilizado na revisão sobre a *estrutura de repetição condicional*

O entendimento demonstrado por José é também bastante importante, qual seja, de que o tempo de execução de um determinado comando, ou conjunto de comandos, pode, com o uso da *estrutura de repetição condicional*, ser determinado por um fator externo e não apenas pelo uso do bloco *esperar*.

Na atividade seguinte, na qual os alunos deveriam criar um programa para fazer o morcego robô funcionar da maneira que eles desejassem, José montou um programa com uma lógica bastante coerente, como pode ser visto na Figura 57. No entanto, não definiu adequadamente o parâmetro de verificação da distância ($\text{Distância} < 15 \text{ cm}$) e também optou por utilizar o LED azul, ao invés do vermelho, equívocos estes que foram corrigidos a partir de questionamentos feitos pelo

professor. Daniel criou seu programa (Figura 58) sem a necessidade de auxílio e muito rapidamente.

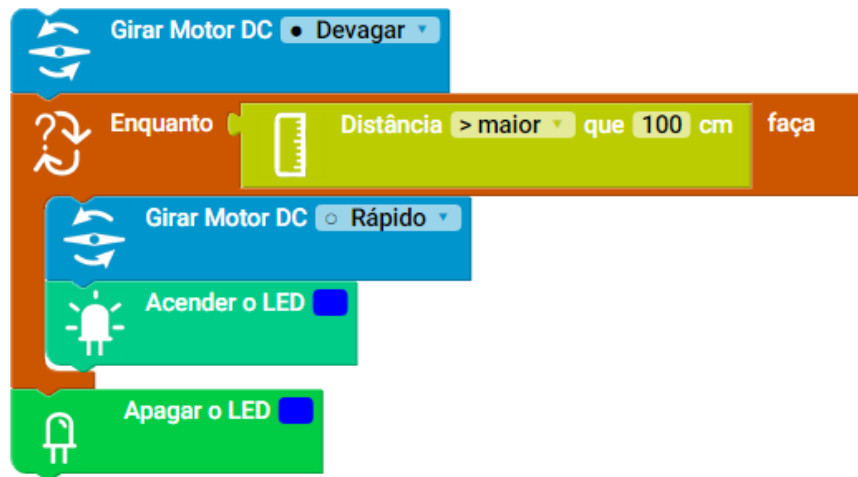


Figura 57: Programa de José para o morcego robô

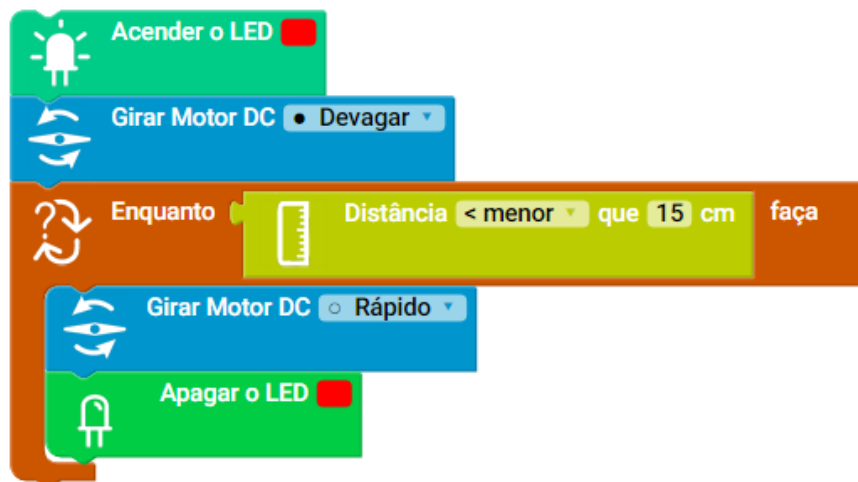


Figura 58: Programa de Daniel para o morcego robô

Marcelo, mesmo partindo de um programa parcialmente montado (Figura 59(a)), não conseguia definir onde colocar os blocos *acender LED* e *apagar LED* dependendo da condição apresentada no bloco *enquanto* ser ou não atendida. Por exemplo, ao ser solicitado que incluísse no programa um comando para manter o LED aceso enquanto a mão estivesse longe do robô, Marcelo posicionou o bloco *acender LED* dentro do laço de repetição (Figura 59(b)). Uma sequência de erros como este, e o mostrado na Figura 59(c), foram cometidos em resposta a uma série de solicitações do professor, o que mais uma vez evidencia o não entendimento de Marcelo em relação ao funcionamento das estruturas de repetição. Cabe ressaltar que, durante esse processo, o professor buscava explicar, paulatina e repetidamente,

qual o resultado esperado quando do uso de cada um dos blocos utilizados no programa.

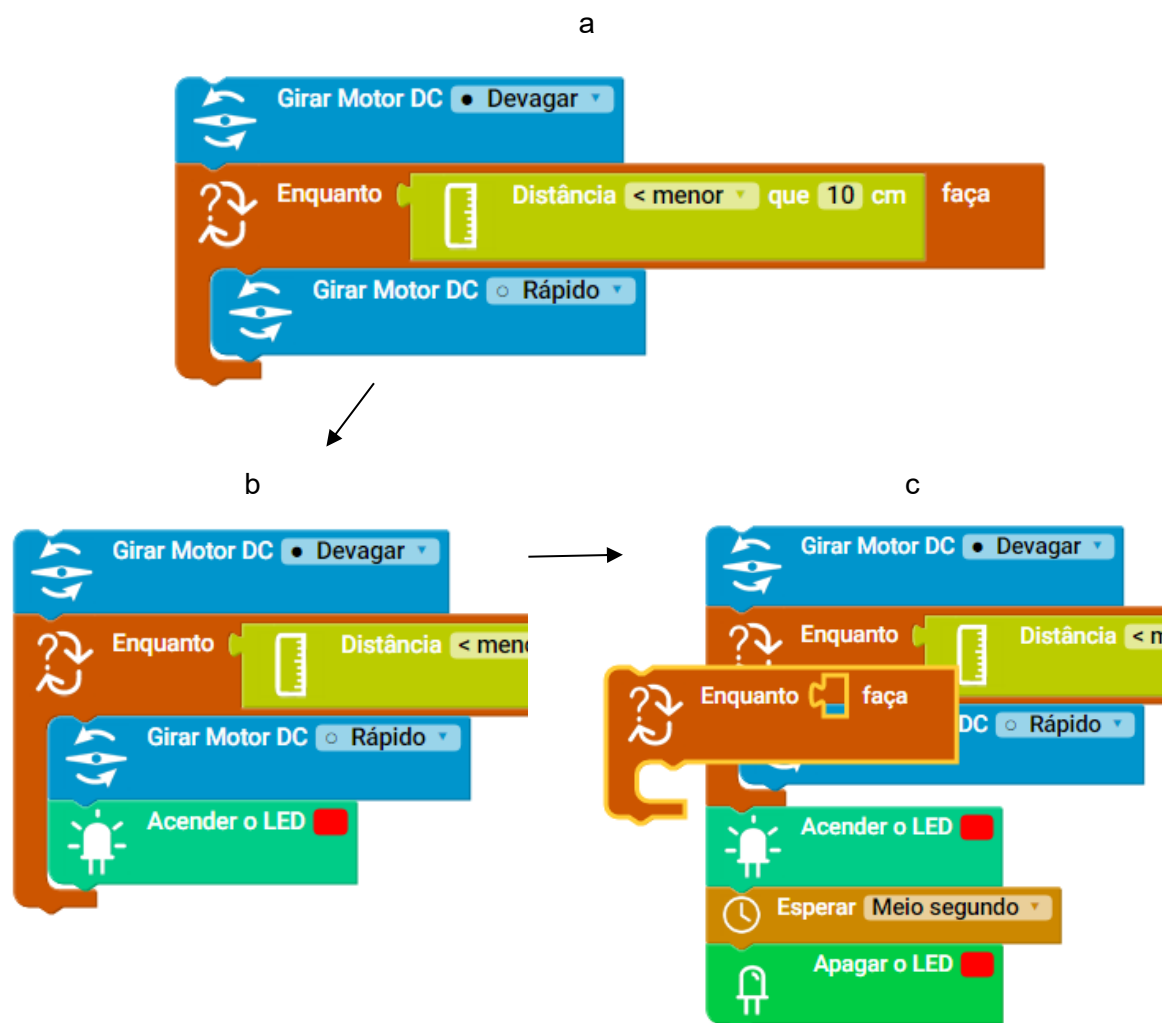


Figura 59: Código base utilizado por Marcelo para programar o morcego robô

Ivana só conseguiu montar o programa com o auxílio de uma assistente do professor que, passo a passo, corrigia e explicava a ela o porquê da utilização de cada um dos blocos. Ademais, assim como Marcelo, não conseguia distinguir quais comandos seriam realizados em função da condição presente no bloco *enquanto*.

Uma constatação bastante interessante pôde ser feita enquanto o professor buscava explicar para Bruno como construir o programa solicitado. Em um determinado momento, durante as orientações, Bruno interrompeu o professor e fez as seguintes perguntas:

Tudo que tá aqui [no programa] é que tá aqui [no morcego robô]? Tipo assim, eu vou apertar aqui [no elemento de interface que envia o programa para a placa] e aí tudo vai passar pra cá [morcego robô]?

Esses questionamentos sugerem que Bruno estava começando a desenvolver o entendimento de que os programas que eles construíam eram abstrações que se materializavam na caixinha mágica ou nos outros materiais de robótica trabalhados em aula. Esse mesmo entendimento já vinha, há algum tempo, sendo apresentado por José, Daniel e Hilton que, mesmo não tendo compreensão do que vem a ser uma abstração, demonstram total consciência da relação existente entre os programas que desenvolvem e os comportamentos apresentados pelos dispositivos eletrônicos presentes nos materiais de robótica. Já Ivana e Marcelo até então não apresentaram indícios claros de perceberem essa relação.

Esta dificuldade na construção de abstrações por parte de algumas crianças, como já mencionado, pode ser resultado de uma imaturidade cognitiva perfeitamente compreensível, uma vez que a capacidade de construir abstrações, ainda que apoiadas em materiais concretos ou em aspectos observáveis de uma determinada ação, como visto no capítulo 2.4, tem seu amadurecimento iniciado por volta dos sete/oito anos, e como os demais aspectos relacionados ao desenvolvimento cognitivo, possui forte influência de fatores sociais e educacionais.

A última atividade, na qual os alunos deveriam programar o carrinho robô para funcionar conforme o observado, exigia o uso de dois laços de repetição condicional, uma vez que o carrinho estava equipado com dois sensores, um de luz e um de distância, e apresentava comportamentos distintos quando a condição verificada por cada um desses sensores era satisfeita (Figura 60).

José construiu o programa corretamente, utilizando, por iniciativa própria e de forma adequada, os dois blocos *enquanto* e suas respectivas condições, além de posicionar os demais comandos nos locais apropriados. Seu único erro foi esquecer de selecionar o operador de comparação correto para o sensor de distância.

Além de fazer o carrinho funcionar conforme o observado, José havia dito que gostaria de fazer com que o LED verde, presente no carrinho robô, permanecesse aceso quando o carrinho estivesse próximo à algum objeto e ficasse apagado o restante do tempo. Assim, finalizado o programa básico, o professor solicitou que José modificasse o seu código de maneira que o carrinho passasse a funcionar conforme ele havia imaginado. Em resposta a essa solicitação, José posicionou, de imediato, os blocos *acender LED* e *apagar LED* nos locais adequados dentro do programa (Figura 61).

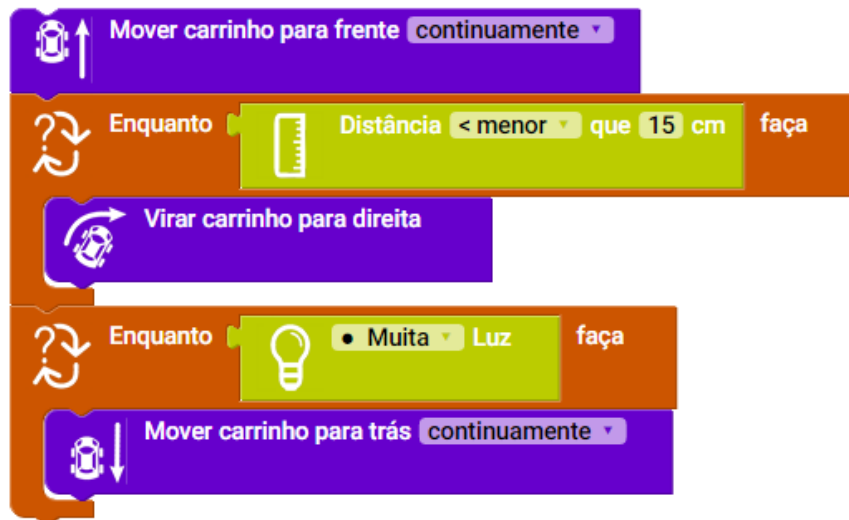


Figura 60: Programa básico do carrinho robô

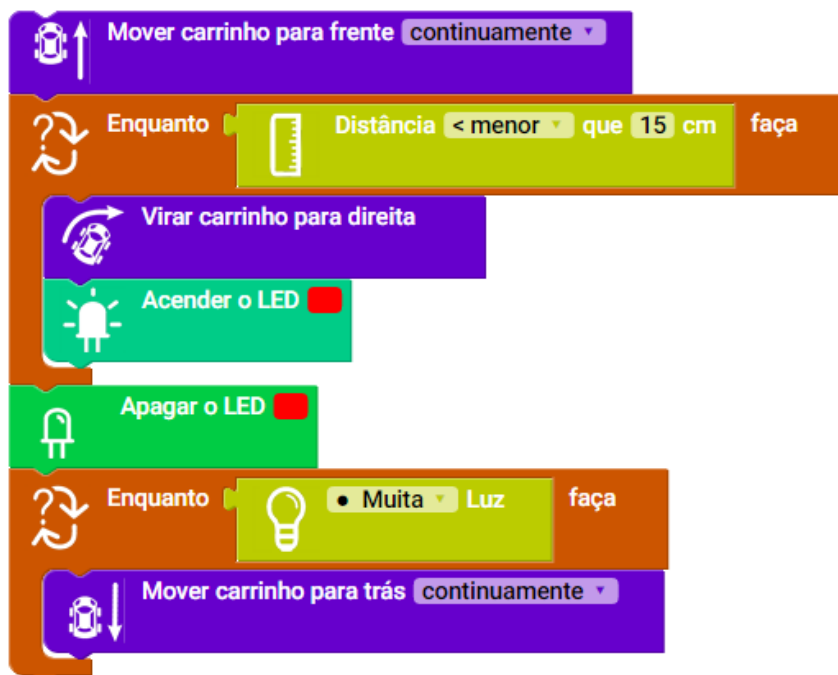


Figura 61: Programa do carrinho modificado por José

Daniel também percebeu, por conta própria, a necessidade do uso de dois blocos *enquanto*, um para cada sensor, mas, posicionou um dentro do outro (Figura 62). No entanto, ao ser questionado sobre como o carrinho funcionaria se seguisse os comandos do programa do modo como ele os havia organizado, Daniel imediatamente percebeu o equívoco e reposicionou os blocos nos locais adequados.

Hilton não estava presente neste encontro.

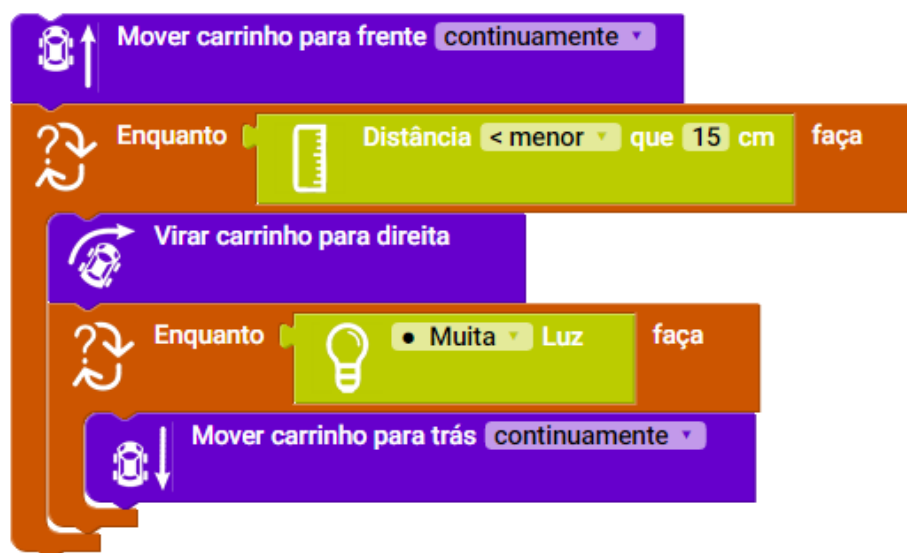


Figura 62: Programa inicial de Daniel para o carrinho robô

A percepção de José e Daniel acerca da necessidade do uso de dois blocos *enquanto* para que fosse possível reproduzir, na construção do programa, o comportamento apresentado pelo carrinho em resposta ao atendimento de duas condições distintas evidencia o crescente entendimento desses alunos acerca dos conteúdos trabalhados em aula.

Em relação às crianças que vinham apresentando maior dificuldade, para possibilitar que elas pudessem tentar desenvolver ao menos parte do programa, o professor solicitou que montassem somente o código necessário para fazer com que o carrinho não batesse nas coisas, ou seja: andar para frente e, quando estivesse próximo a algum objeto, virar para direita ou para a esquerda (à escolha deles). Ainda assim, Ivana só conseguiu construir o programa com o auxílio da assistente do professor, que corrigia, passo a passo, os erros cometidos. Do mesmo modo, Marcelo e Bruno, apesar das explicações detalhadas feitas pelo professor sobre a lógica de funcionamento do programa, permaneceram não demonstrando entendimento.

5.3.5 Quinto Encontro (Aula 13)

5.3.5.1 Atividades Desenvolvidas

O quinto encontro teve início com a apresentação do cachorro robô de garrafa pet (ver capítulo 3.1). O cachorro possuía rodinhas, como as do carrinho robô, um

sensor de distância na parte de cima da cabeça, um sensor de luz no focinho, e um rabinho controlado por um servo motor. Sempre que uma luz forte incidia sobre o focinho do cachorro (o fecho de uma lanterna, por exemplo) ele se movia para frente e, quando alguém tocava na sua cabeça, ou seja, quando recebia carinho, ele balançava o rabinho de um lado para o outro. As crianças puderam observar, durante a apresentação do cachorro, quais atuadores e sensores haviam sido utilizados na sua construção e que comportamentos ele assumia quando as condições verificadas por cada um dos seus sensores eram atendidas.

Na sequência, passou-se para a apresentação do servo motor, dispositivo com o qual as crianças ainda não haviam trabalhado. O funcionamento do servo foi explicado detalhadamente, sendo em seguida exibida uma animação no projetor com o servo motor movimentando-se do mesmo modo que o rabo do cachorro robô.

Finalizada a apresentação, foi solicitado às crianças que fizessem um programa para que o servo motor presente na caixinha mágica funcionasse de maneira semelhante ao rabinho do cachorro. Foram colados para isso miniaturas de rabinhos sobre os ponteiros dos servos presentes nas caixinhas.

Dando continuidade ao desenvolvimento do programa para o cachorro robô, as crianças receberam a tarefa de modificar o código construído na atividade anterior, fazendo agora com que o servo funcionasse apenas quando algo se aproximasse do sensor de distância presente na caixinha mágica. Concluída esta etapa, para terminar a programação do cachorro robô, foi solicitado às crianças que completassem o programa utilizando os blocos do carrinho robô para que o cachorro se movesse para frente sempre que a luz estivesse muito forte.

Finalizando as atividades do dia as crianças foram apresentadas ao braço robótico, um exemplo prático do uso de servo motores. O professor explicou às crianças a quais comandos o braço respondia e como utilizá-los para ordenar que o braço trocasse a posição de alguns “peões” presentes em um tabuleiro (Figura 63). Foi então solicitado às crianças que construíssem um programa que fizesse com que o braço robótico colocasse os peões nas posições desejadas. Uma vez concluídos, os programas eram carregados no braço robótico para que as crianças pudessem observá-lo realizando as ações determinadas⁹⁷.

⁹⁷ A narrativa que contextualiza o braço robótico encontra-se no apêndice A, aula 13.

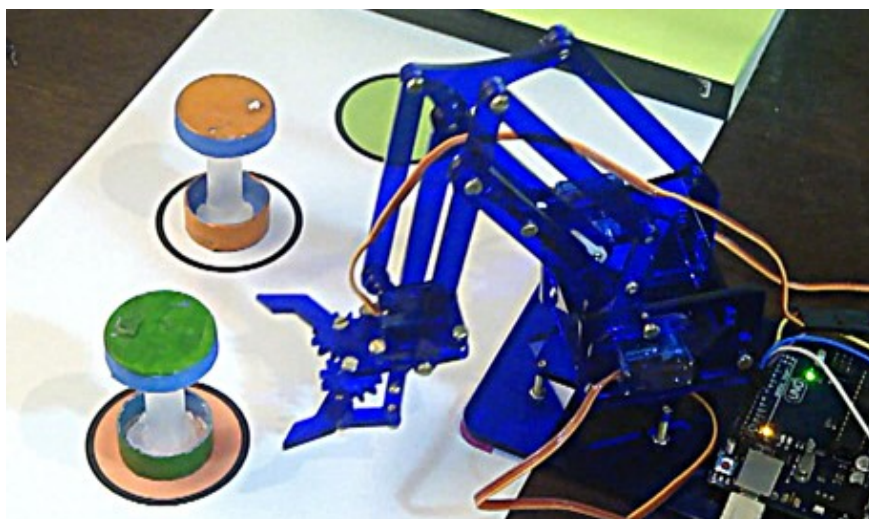


Figura 63: Tabuleiro e peões para atividade com o braço robótico

5.3.5.2 Resultados

Como mencionado na descrição das atividades, a construção do programa para o cachorro robô (Figura 64) foi dividida em 3 partes.

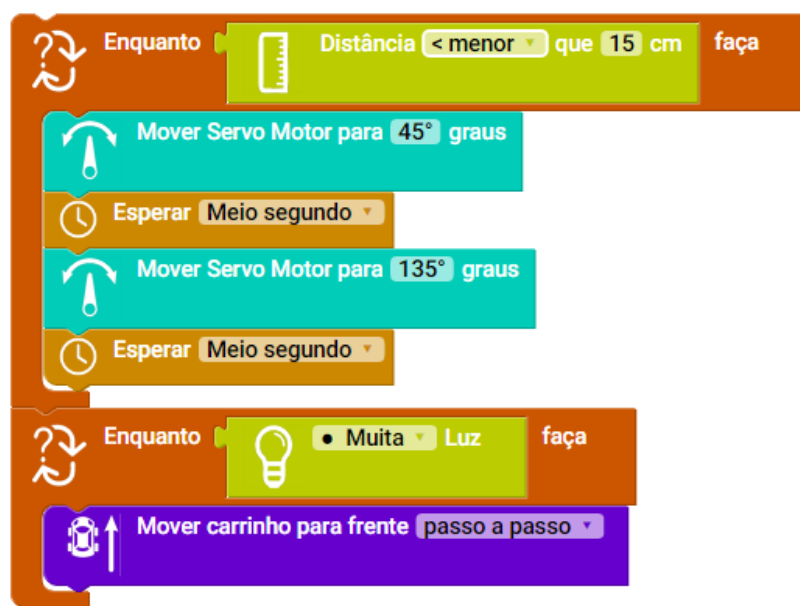


Figura 64: Programa de controle do cachorro robô

Daniel, Hilton e José fizeram sem maiores dificuldades a primeira parte da programação, que consistia em fazer o servo motor virar para a direita e para a esquerda. José esqueceu de colocar o bloco *esperar* na última “linha” do programa (Figura 65), mas, ao ser questionado se “não havia esquecido nada”, imediatamente

pegou um bloco *esperar* e colocou no local correto. Daniel cometeu o mesmo erro de José, no entanto, não conseguiu inferir, com base na observação da execução do programa na caixinha mágica, que o servo motor não estava se movendo conforme o esperado devido à ausência de um dos blocos *esperar*.

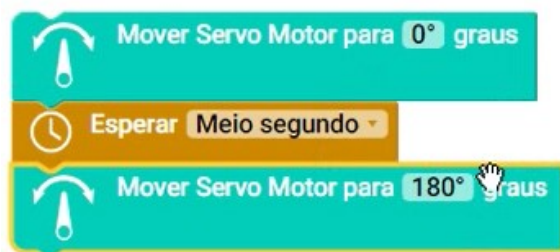


Figura 65: Erro cometido por José na parte do programa responsável por mover o rabinho do cachorro robô

Ivana e Marcelo montaram seus programas olhando os programas de Hilton e José, respectivamente. Devido ao posicionamento da filmadora, não foi possível verificar se Bruno montou o programa (muito semelhante ao de Daniel) ou copiou do colega. No entanto, considerando-se as dificuldades que Bruno vinha apresentando, a conclusão mais plausível é a de não ter sido ele o autor do programa. Em decorrência de Bruno ter encerrado o programa de captura, também não foi possível avaliar, por meio dos vídeos, a sua performance nas demais atividades do dia, contando-se apenas com as anotações feitas pelo professor ao término da aula.

Na segunda parte da programação do cachorro robô, onde foi solicitado que as crianças fizessem, na caixinha mágica, o servo se mover apenas quando algo se aproximasse do sensor de distância, Daniel, Hilton e José, corretamente, colocaram um bloco *enquanto* no programa, selecionaram a condição adequada e posicionaram os blocos já presentes no programa dentro do bloco *enquanto* (Figura 66).

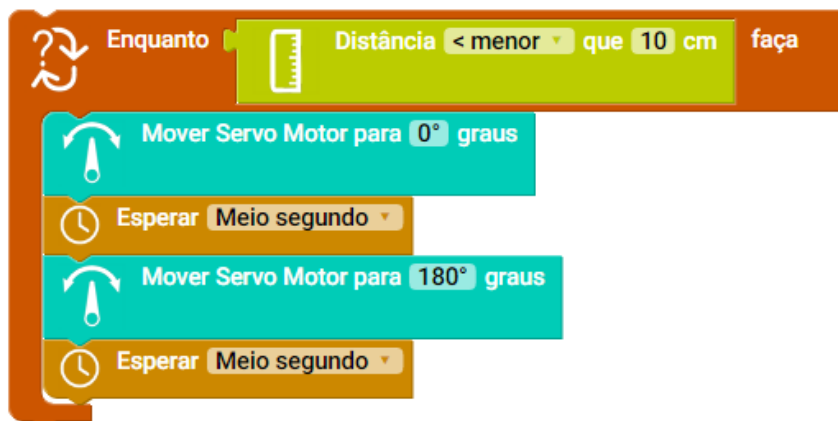


Figura 66: Código de José para a segunda etapa de construção do programa do cachorro robô

Ivana, novamente, montou seu programa olhando o programa de Hilton. Marcelo pegou um bloco *enquanto*, encaixou o bloco *do sensor de distância* no local certo, mas, posicionou o bloco *enquanto* embaixo da sequência de comandos para mover o servo motor (Figura 67).

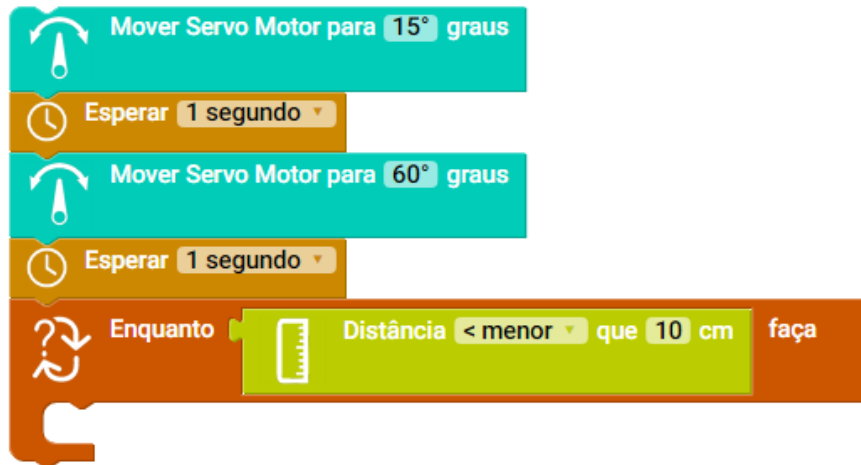


Figura 67: Código de Davi para segunda etapa do programa do cachorro robô

A partir deste ponto seguiu-se o seguinte diálogo entre o Professor e Marcelo:

Professor: Está ficando legal. Mas, enquanto for menor, ele vai fazer o que?

Marcelo: Acender o LED vermelho.

Professor: Não, não tem LED aqui. O que que o tio quer? Que enquanto for menor ele fique girando para lá e para cá (mostrando o servo motor movendo-se para direita e para a esquerda na caixinha). Como é que faz isso? Onde é que está mandando ele girar para lá e para cá?

(Marcelo pega então um bloco girar motor DC e posiciona dentro do bloco *enquanto* (Figura 68)).

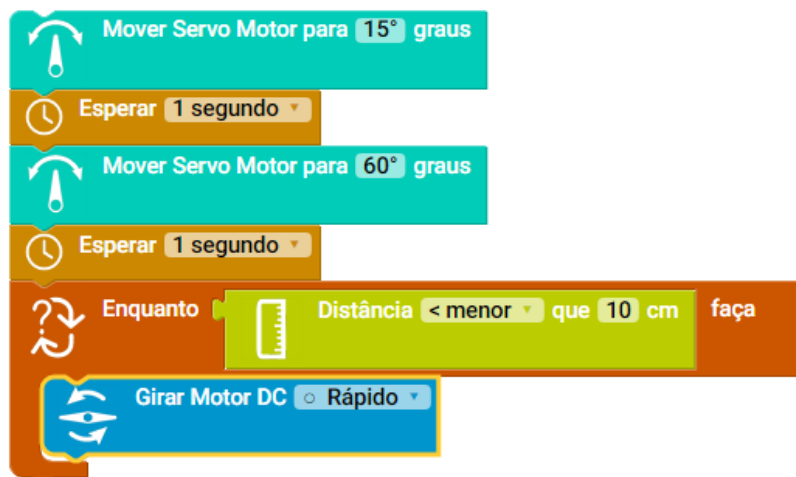


Figura 68: Uso equivocado do motor DC no programa do cachorro robô

Professor: Não. Calma aí, Marcelo. Não é esse motor. Olha só, esse programa que você fez aqui, ele mandava a caixinha fazer o quê? Você entende que você faz um programa aqui e aí esse programa vem para a caixinha e ela faz? Então, e ele mandava esse negócio (o servo motor) ir para lá e para cá, não mandava? Agora eu quero que só quando você aproximar a mão ele faça isso que já fazia. Ele já não estava fazendo? Agora você só quer que ele faça em outra situação.

Marcelo vai então no menu do braço robótico e diz: O braço.

Professor: Não. Olha só, Marcelo. Aqui em cima ele já não está fazendo isso? Só que eu quero que ele faça isso só quando estiver aqui dentro (apontando para o bloco *enquanto*). Se a gente colocar aqui para dentro, você acha que não funciona? Pensa assim... Pega esse bloco aqui de cima.

(Marcelo pega então, no menu de blocos, um novo bloco *mover servo motor*)

Professor: Mas você não precisa fazer de novo. Já está aqui (apontando para a sequência de blocos responsável por mover o servo motor). Para que você vai fazer de novo? É só fazer o quê? É só botar isso aqui...vê se você vai entender...o tio vai dizer o que vai acontecer aqui. Isso aqui está fazendo o quê? Mandando para um lado, esperando, e mandando para o outro, não é? Só que eu quero que ele faça isso aqui só enquanto a distância for menor. Vamos ver como a gente faz isso? Olha só (move os comandos para dentro do bloco *enquanto*). A gente coloca aqui para dentro. Viu? Agora ele só vai fazer enquanto estiver aqui dentro. Se não estiver aqui dentro ele não faz.

Marcelo: Vou mandar (Indicando que vai enviar o programa para a caixinha).

Professor: Pode mandar.

Esse diálogo indica que Marcelo não conseguiu, durante a oficina, entender que os programas que construía eram abstrações que se materializavam nos materiais de robótica com os quais ele estava trabalhando. Ou seja, Marcelo não demonstrou reconhecer a relação existente entre os comandos presentes nos programas e os comportamentos apresentados pelos materiais de robótica. O fato de Marcelo não conseguir reorganizar ou “reaproveitar” os blocos do programa, também pode ser o indicativo de uma dificuldade em relação à reversibilidade (ver capítulo 2.4).

A parte final do programa para o cachorro robô, que consistia em fazê-lo andar para frente quando uma luz forte incidisse sobre o sensor presente em seu focinho (Figura 69), foi concluída com exatidão por Daniel, Hilton e José. Ivana continuou copiando o programa de Hilton. Quanto à Marcelo, este não se motivou a finalizar o programa.

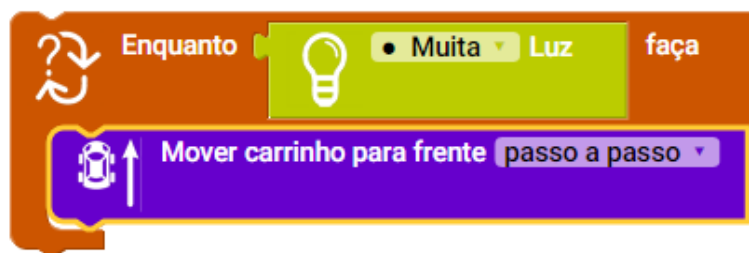


Figura 69: Terceira e última parte do código para o programa do cachorro robô

Para a última atividade deste encontro, em que as crianças deveriam programar o braço robótico, era necessário apenas definir uma sequência de movimentos (mover o braço para a direita, esquerda, centro, pegar objeto e soltar objeto), ou seja, não era necessário o uso das estruturas de repetição nem do bloco *esperar* (Figura 70). No início da atividade o professor mostrou o braço robótico executando os movimentos necessários à realização da troca da posição dos peões e depois solicitou às crianças que construíssem o programa responsável por fazer o braço operar da maneira observada.



Figura 70: Programa para o braço robótico

Importante relatar aqui, que um equívoco no planejamento dessa atividade dificultou a sua realização por parte de algumas crianças: havia apenas um tabuleiro com peões, assim, as crianças não tinham como verificar, continuamente, quais os movimentos necessários para o braço trocar os peões de lugar a partir da organização inicial do tabuleiro apresentada pelo professor.

Hilton fez o programa correto (Figura 70) sem a necessidade de auxílio, utilizando apenas a lembrança dos movimentos que havia observado o braço robótico realizar.

Daniel, em um primeiro momento, montou apenas metade do programa (Figura 71). No entanto, ao ser questionado pelo professor se não faltava colocar o outro peão no lugar adequado, ele percebeu que sim e finalizou o programa corretamente. Como Daniel também estava construindo o programa apenas com a lembrança dos movimentos que havia visto o robô executar, sem olhar novamente para o tabuleiro e os peões, e como mais uma vez, estava um tanto desatento envolvido com outras atividades no computador, a falha cometida por ele é bastante compreensível.



Figura 71: Primeira tentativa de Davi para o programa do braço robótico

José, que normalmente realizava as tarefas por conta própria, solicitou a ajuda do professor que, simulando com seu braço o braço robótico, executava, uma a uma,

as ações determinadas por José que, após observá-las, escolhia os blocos correspondentes e os posicionava em seu programa.

Já, Ivana, construiu seu código simulando com o próprio corpo os movimentos que havia observado. Para tanto, com o auxílio de uma cadeira giratória, girava o tronco para direita, esquerda e centro, simulando com um de seus braços os movimentos de elevação e descida do braço robótico e, com uma das mãos, os movimentos da garra. Antes de partir para esta estratégia, Ivana já havia tentado construir o programa por meio de outras abordagens, de modo que já existia um trecho do código presente no seu computador. Assim, a partir da adoção dessa nova “técnica”, a cada ação simulada, Ivana buscava uma correspondência entre o movimento por ela realizado e os blocos que já havia inserido no programa, mantendo, retirando ou inserindo novos blocos em resposta à essa “depuração”. Como resultado desse processo, Ivana errou apenas um comando na construção do seu código e, quando o professor fez, com seu braço, a simulação da execução do programa, ela rapidamente identificou o problema, pegou o bloco correspondente ao movimento correto e fez a substituição no ponto exato do código.

Marcelo pegou, um a um, os blocos presentes do menu do braço robótico e os posicionou em sequência na área de trabalho sem buscar estabelecer qualquer relação com os movimentos que havia observado. Isso feito, colocou um bloco *esperar* entre cada um dos blocos de movimentação do braço e gritou quatro vezes: “Tio acabei”, perguntando logo em seguida: “Cadê o robótico?” (Referindo-se ao braço, para poder carregar seu programa). Interessante perceber que Marcelo guardou a informação referente ao tempo necessário para poder-se observar a execução de determinados comandos (uso do bloco *esperar*). No entanto, no caso do braço robótico, não havia essa necessidade⁹⁸.

Pouco depois, o professor auxiliou Marcelo na programação, simulando com seu braço o braço robótico. A cada passo o professor perguntava: “o que você acha que o braço tem que fazer agora?”. Marcelo então respondia e colocava o bloco correspondente no programa. Com esse auxílio, Marcelo acertou todos os movimentos necessários para o braço robótico colocar os peões nas posições corretas do tabuleiro. Assim como Marcelo, Bruno também só fez a atividade com o

⁹⁸ Usar o bloco *esperar* após os blocos do braço robótico fara apenas com que o braço fique parado, durante o tempo especificado como parâmetro, antes de realizar o próximo movimento.

auxílio do professor, mas, apenas a partir de diversas solicitações deste, diferentemente do que aconteceu com Marcelo e Ivana que apresentaram absoluto interesse pela atividade.

Interessante perceber que, mesmo com muita dificuldade em acompanhar os trabalhos realizados em sala, de maneira geral, Marcelo e Ivana buscavam, a seu modo, finalizar as tarefas propostas, além de demonstrarem entusiasmo em observar o resultado da execução dos programas nos materiais de robótica.

Outro aspecto importante a ser considerado diz respeito à performance apresentada por Ivana na programação do braço robótico. Diferente do que acontece com outros blocos, como *esperar*, *enquanto* e *repita*, os blocos utilizados para programar o braço robótico são representações diretas de ações concretas. Assim, a habilidade demonstrada por Ivana em construir o programa para o braço robótico, em contraposição à sua dificuldade em desenvolver os programas que utilizavam noções abstratas de tempo e repetição, pode ser um indicativo de que ela, em relação à certos aspectos, possua uma maturidade cognitiva ainda pré-operatória. Neste período as crianças não conseguem, de modo geral, realizar operações mentalmente, razão pela qual, possivelmente, Ivana recorria à execução física de cada uma das ações para a construção, passo a passo, da sequência de comandos necessária para que o braço robótico realizasse a tarefa proposta.

5.4 AULA DE ENCERRAMENTO (AULA 14)

5.4.1 Atividades Desenvolvidas

No sexto e último encontro foram realizadas algumas atividades de avaliação. Na primeira atividade, as crianças deveriam observar, no projetor, um programa feito no DB4K e, através de um texto, descrever o seu funcionamento.

Para a segunda atividade, as crianças receberam uma folha A4 onde em uma coluna havia o nome de alguns sensores e atuadores trabalhados em aula e, em uma coluna ao lado, imagens desses sensores e atuadores. As crianças deveriam ligar os nomes dos dispositivos às imagens corretas.

Na terceira atividade era apresentado no projetor um programa feito no DB4K e, ao lado, um vídeo com a caixinha mágica executando um determinado programa.

As crianças deveriam verificar se o programa exibido correspondia ao que a caixinha estava executando, caso contrário, indicar quais modificações deveriam ser realizadas no código para que ele correspondesse ao programa sendo executado pela caixinha mágica.

Como última atividade as crianças receberam papéis cartão A4 e lápis de cor para desenhar um robô que deveria, necessariamente, usar alguns dos sensores e atuadores com os quais elas haviam trabalhado em aula. Por fim, ao terminarem o desenho, deveriam explicar ao professor o funcionamento dos robôs por elas projetados.

5.4.2 Resultados

Daniel e Hilton faltaram ao último encontro. Assim, as atividades da última aula foram realizadas com eles, em separado, em um encontro realizado na semana seguinte. No entanto, os resultados por eles apresentados serão descritos em conjunto com os das demais crianças.

A primeira atividade, na qual as crianças deveriam descrever textualmente o funcionamento de um programa exibido pelo projetor (Figura 72), não foi bem-sucedida, uma vez que nenhum dos alunos conseguiu concluir sequer uma linha inteira de texto. Por esse motivo esta atividade não foi aplicada posteriormente com Hilton e Daniel. Apenas José conseguiu verbalizar oralmente, a partir de perguntas feitas pelo professor, o que seria observado na caixinha caso o programa exibido fosse executado. No entanto, apesar de ter conseguido fazer uma descrição oral, ele teve bastante dificuldade em transformar em texto as descrições que fazia, desconcentrando-se da tarefa e, desse modo, não chegando a concluir uma frase sequer.

Em determinado momento desse exercício, em que Bruno e Marcelo estavam lendo, um a um, os comandos presentes no programa, José disse: “Piscar, cara. Ele não está perguntando o que tem ali, ele está perguntando o que faz: piscar”. Esta fala de José explicita um melhor entendimento dele, em relação aos seus colegas, acerca da proposta da atividade.

Interessante também destacar, em relação ao entendimento de José, o fato de ele em um dado momento ter perguntado se poderia separar a descrição em partes,

ou seja, primeiro escrever que o LED iria piscar 3 vezes, depois que o motor iria girar, e assim por diante. Percebe-se com isso que José conseguia identificar nitidamente a presença de grupos distintos de ações no programa, o que pode ser entendido como um exercício da decomposição estruturada de problemas, uma característica do Pensamento Computacional.



Figura 72: Programa a ser descrito textualmente pelas crianças

Diante da dificuldade apresentada pelos alunos em relação a essa tarefa, o Professor buscou, durante um bom tempo, por meio de diferentes explicações, fazer com que as crianças compreendessem o que estava sendo pedido. Mas, isso não aconteceu. Bruno, Ivana e Marcelo não conseguiram compreender como era possível

explicar o funcionamento do programa de uma maneira que não fosse ler, um a um, os comandos nele presentes. Ivana, inclusive, não conseguiu responder aos seguintes questionamentos feitos pelo professor (sempre em tom carinhoso e em separado das demais crianças):

Professor: Só diz para o tio. Quando a gente liga o LED meio segundo e apaga meio segundo, o que é que a gente vê acontecendo?
 (Silêncio de Ivana)
 Professor: A gente diz assim: esse LED está ... O quê?
 (Silêncio de Ivana)
 Professor: Você vai dizer assim: Ah, esse LED está acendendo, apagando, acendendo, apagando, ou você vai dizer: esse LED está O quê?
 (Silêncio de Ivana)
 Professor: Se o tio disser assim: faz o LED piscar. Como é que é pra fazer?
 (Silêncio de Ivana)
 Professor: Hein? Como faz o LED piscar?
 (Silêncio de Ivana)
 Professor: Você não sabe fazer o LED piscar? Olha.
 (O professor então pega uma lanterna e acende e apaga algumas vezes)
 Professor: O que é que o tio está fazendo? Acendendo, apagando, acendendo, apagando. O que é que ela faz? Pisca. Não é?
 (Ivana concorda com a cabeça)
 Professor: Então! É isso!

A segunda atividade, onde as crianças deveriam ligar os nomes dos atuadores e sensores às suas imagens (Figura 73), foi facilmente completada por Daniel, Hilton e José. Marcelo e Ivana precisaram de ajuda, pois não lembravam dos nomes de alguns atuadores e sensores. Bruno não quis realizar essa atividade.

Na atividade de comparação dos programas com os vídeos, foram mostrados dois programas. O primeiro programa (Figura 74) apresentava três erros e o segundo programa (Figura 75) não apresentava erros. José e Daniel encontraram facilmente dois dos três erros existentes no primeiro programa, sendo eles: a ausência dos blocos *esperar* para fazer o LED piscar e posicionamento incorreto do bloco *girar motor DC*. No entanto, não repararam que os números de repetições indicados nos blocos *repetir* estavam errados.

Hilton encontrou todos os erros, mas, demorou um pouco mais que os colegas para perceber a ausência dos blocos *esperar*. Importante mencionar que Hilton foi o aluno que mais rapidamente entendeu o uso e passou a aplicar corretamente o bloco *esperar* em seus programas. Assim, essa dificuldade em perceber o erro no programa

pode ter sido consequência da ansiedade, já que Hilton havia relatado estar tenso pois iria “ser avaliado”. Hilton era bastante ansioso, sempre querendo terminar os programas antes dos colegas, o que acabava levando-o a cometer muitos erros resultantes, nitidamente, da falta de atenção, uma vez que eram erros relacionados à temas sobre os quais ele já havia demonstrado bastante domínio.

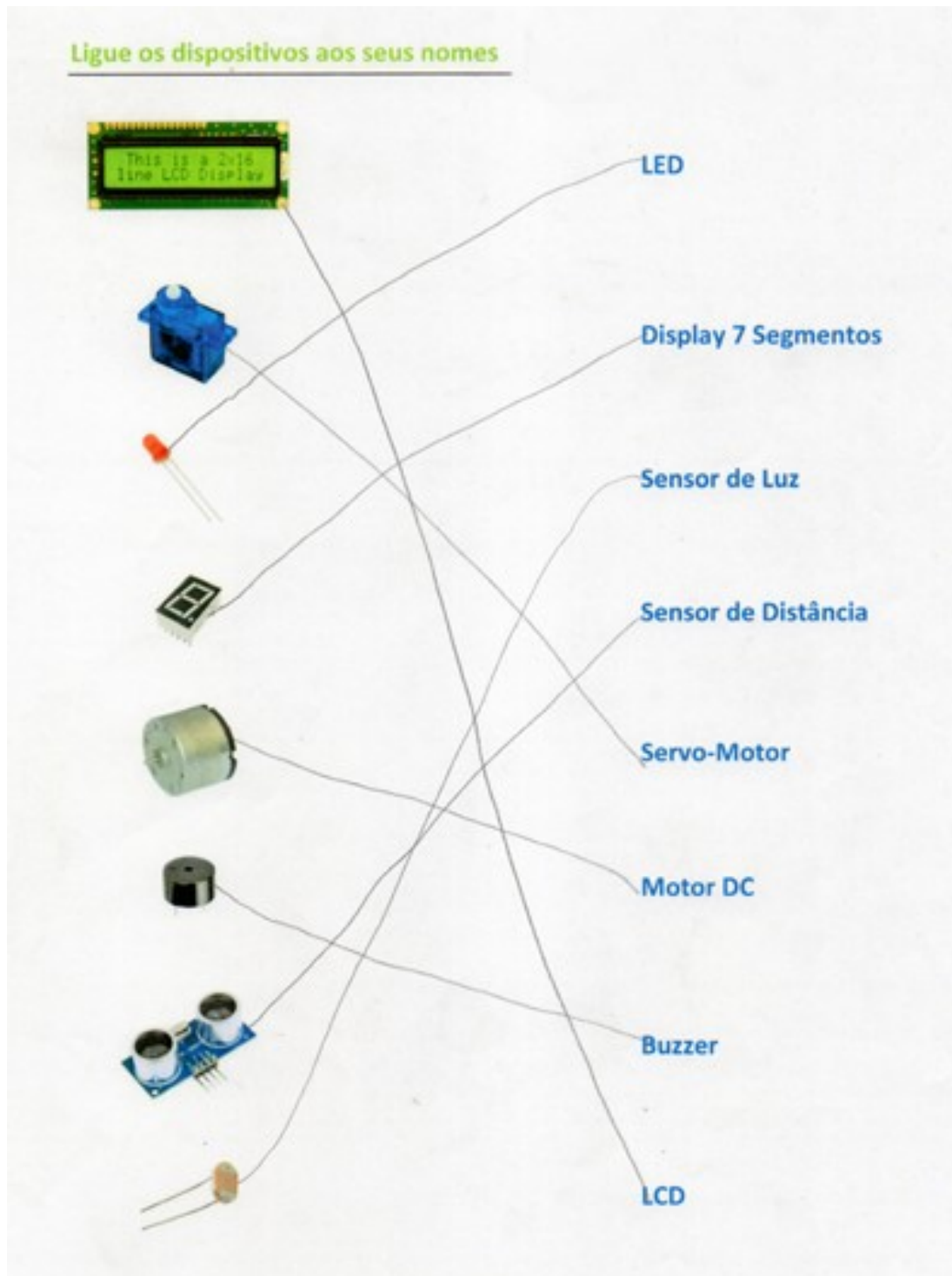


Figura 73: Exercício de reconhecimento dos dispositivos

Marcelo percebeu apenas o posicionamento incorreto do bloco *girar motor DC*, o que indica uma pequena melhora do seu entendimento em relação às estruturas de repetição. Bruno e Ivana não se interessaram pela atividade.

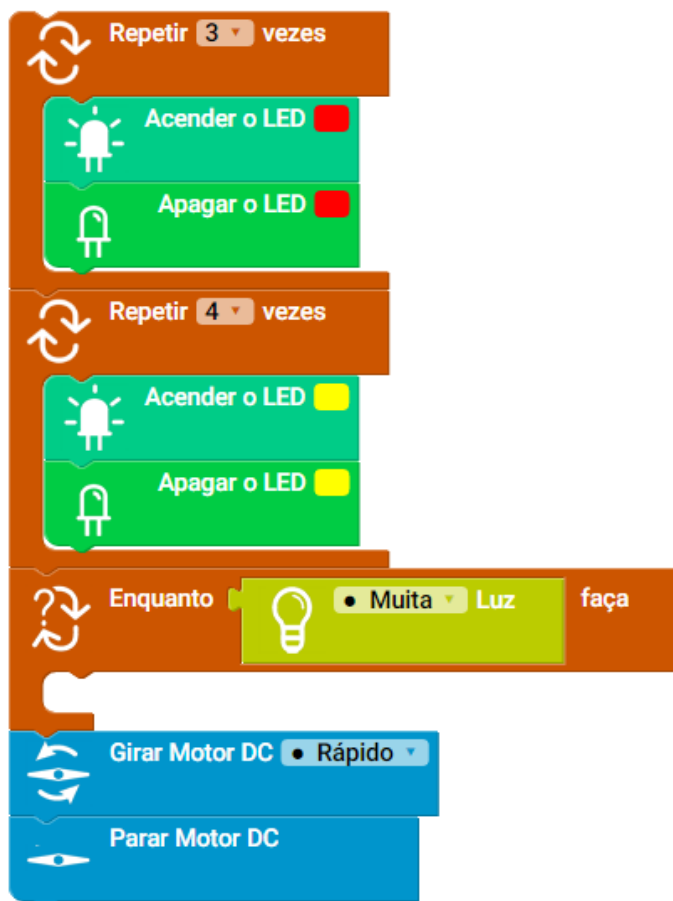


Figura 74: Programa com erros

Na análise do segundo programa (Figura 75), no qual o código apresentado correspondia exatamente ao que estava sendo executado na caixinha, Daniel, Hilton e José, após longa análise, indicaram todos o mesmo possível “erro”: o bloco *esperar* após o bloco *escrever no display de 7 segmentos*. O professor explicou então, a cada um, que aquele bloco realmente não era necessário, mas, que se eles observassem bem, veriam que havia um tempo entre a escrita no display e o LED começar a piscar. Este equívoco, cometido pelos três alunos, foi possivelmente induzido pela maneira como o desafio foi apresentado pelo professor, que insinuava a existência de erros, o que levava às crianças à necessidade de encontrá-los, já que partiam do pressuposto de que algum erro existia.

Marcelo acompanhou a atividade apenas descrevendo o que observava acontecendo na caixinha, ou seja, ele não buscava correlacionar os comandos do programa com as ações observadas. Bruno e Ivana não se interessaram por essa atividade.



Figura 75: Programa sem erros

5.4.2.1 Projetando um Robô

A última atividade, na qual foi solicitado que as crianças desenhassem um robô e explicassem seu funcionamento, deixou mais uma vez evidente a diferença de assimilação dos conteúdos entre as crianças. Daniel, Hilton e José, além de atuadores, utilizaram sensores em suas criações, sendo estes responsáveis por modificar o comportamento de seus robôs em resposta a determinadas alterações no ambiente. Já Bruno, Ivana e Marcelo desenharam robôs que apenas utilizavam atuadores e possuíam uma lógica de funcionamento bem pouco precisa.

José desenhou um carro robô de combate (Figura 76), inspirado em um vídeo de luta de robôs apresentado em aula. O robô possuía quatro rodas, uma serra circular na parte dianteira (movida por um motor DC), um sensor de distância acoplado à serra,

um canhão na parte superior e um sensor de luz na lateral. De acordo com a explicação dada por José, a serra iria girar sempre que um outro robô se aproximasse, o que era percebido pelo sensor de distância. Já o canhão, atirava sempre que uma luz forte estivesse incidindo sobre o robô.

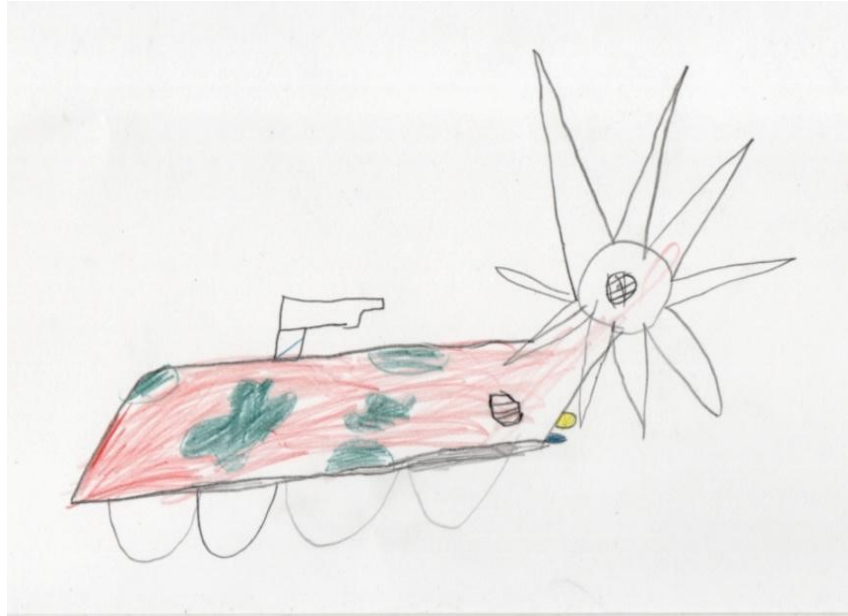


Figura 76: Robô lutador desenhado por José

O robô de Daniel (Figura 77) possuía um buzzer, um display LCD, um LED e um sensor de distância. O Robô ficava parado com o LED apagado e com o texto “Olá” escrito no LCD. Sempre que algo se aproximava (o que era percebido pelo

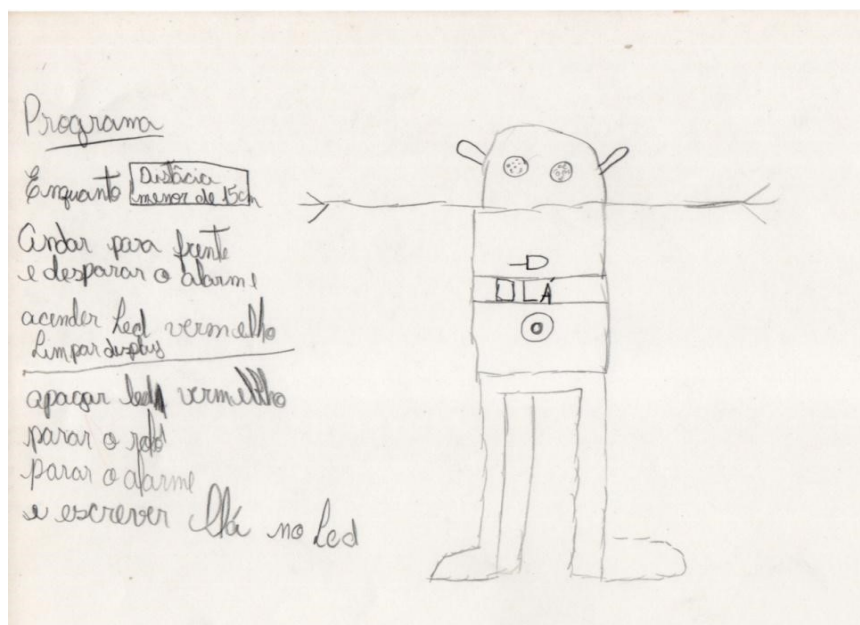


Figura 77: Robô desenhado por Daniel

sensor de distância), o robô disparava o buzzer (alarme), acendia o LED e limpava o Display. Caso o objeto se distanciasse, o robô voltava ao estado inicial. Embora não tenha sido solicitado, Daniel fez também o programa (Figura 78) para o funcionamento do robô, tanto no desenho quanto no ambiente de programação, utilizando o bloco do carrinho para fazer o robô andar para frente.

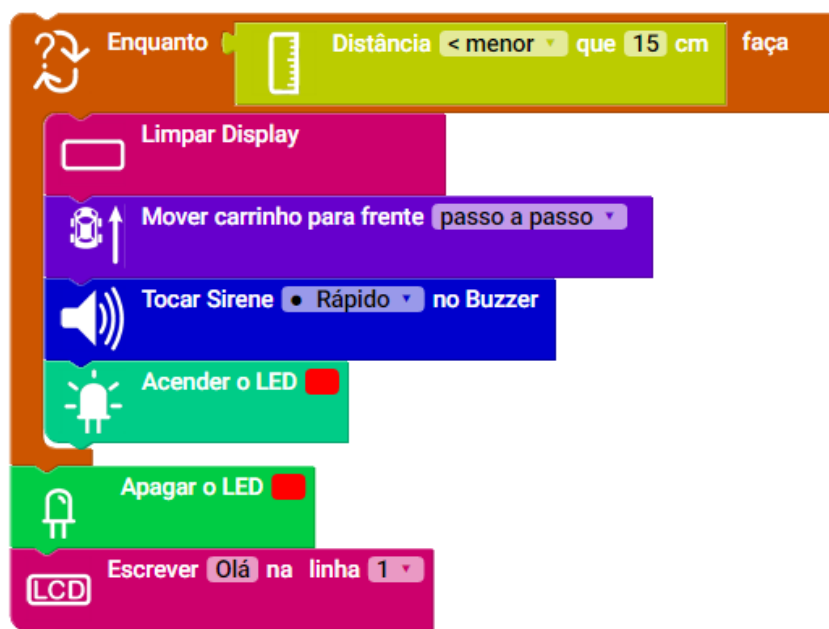


Figura 78: Programa criado por Daniel para o seu robô

Hilton criou uma espécie de “boné robô voador” bastante peculiar (Figura 79). Esse boné possuía uma hélice que o fazia pairar sobre a cabeça do usuário, mas, quando esquentava muito, o que era percebido por meio de um sensor de temperatura, ele descia, encaixando-se na cabeça do usuário até que a aba do boné (que era um longo nariz) ficasse na altura do nariz da pessoa. Uma vez que o boné ficava “enterrado” na cabeça do usuário, a sua parte dianteira possuía uma “janelinha” para não bloquear a visão. Essa janelinha era equipada com um limpador de para-brisa feito com um servo motor, que era acionado em caso de chuva. Além do sensor de temperatura, o boné possuía também um sensor de luz, que fazia com que ele voasse mais rápido quando a luz do sol estivesse muito forte. O boné possuía ainda LEDs para iluminar o ambiente, caso ficasse escuro, e um buzzer que, constantemente, tocava uma musiquinha. Interessante observar que, apesar de não se ter utilizado em aula os blocos do DB4K que possibilitam fazer o Buzzer tocar músicas, Hilton usou essa funcionalidade em seu projeto.

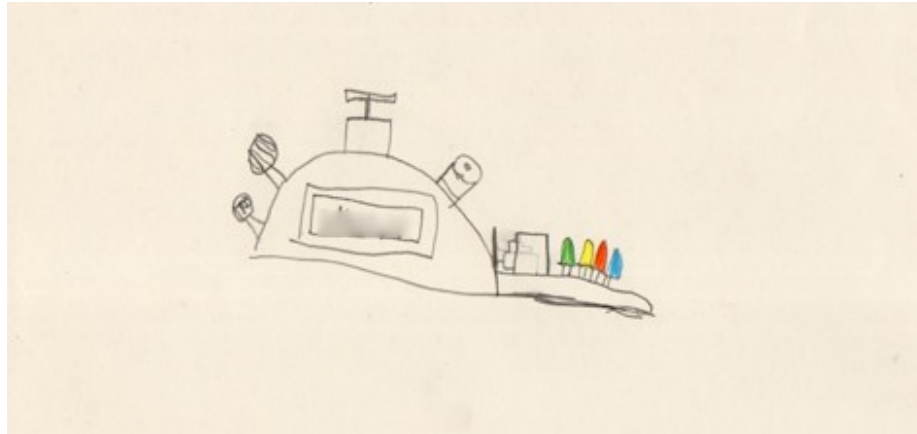


Figura 79: Boné robô desenhado por Hilton

Ivana desenhou um robô fazedor de pipoca e sorvete (Figura 80). De um lado colocava-se o creme para fazer sorvete e, do outro, o milho para fazer pipoca. O robô possuía três LEDs. O LED verde indicava que a pipoca estava pronta, o amarelo que a pipoca já tinha passado do ponto, e o vermelho que a pipoca havia queimado. Embora o robô de Ivana tivesse, necessariamente, que possuir sensores para funcionar conforme o planejado, ela não fez qualquer menção rem relação à essa necessidade.



Figura 80: Robô pipoqueiro-sorveteiro desenhado por Ivana

Marcelo criou um robô jogador de Tênis que possuía duas rodas, duas asas, muitos LEDs coloridos e segurava uma raquete (Figura 81). A não ser pelos LEDs, Marcelo não fez relação entre as partes do seu robô e os dispositivos trabalhados em aula, ou seja, ele descreveu que seu robô jogava tênis, voava, andava com as

rodinhas e acendia os LEDS, mas, não soube explicar de que forma as rodas giravam, as asas batiam, os LEDs acendiam e o braço movia a raquete.



Figura 81: Robô jogador de tênis desenhado por Marcelo

Bruno empenhou-se no desenho de seu robô (Figura 82), porém, assim como Marcelo, não estabeleceu relação entre os dispositivos presentes na caixinha mágica e o funcionamento do robô. Por exemplo, Bruno colocou um servo motor na frente do robô, com os ângulos e o ponteiro muito semelhantes aos do servo presente na caixinha, no entanto, esse dispositivo não possuía qualquer função prática no robô desenhado.



Figura 82: Robô desenhado por Bruno

5.5 SÍNTESE DA AVALIAÇÃO DO APRENDIZADO

Esta seção apresenta uma descrição sucinta dos resultados observados no que concerne ao aprendizado das estruturas básicas de programação e sua relação com aspectos do Pensamento Computacional. Maiores detalhes acerca dos dados que levaram às conclusões aqui apresentadas encontram-se nas seções anteriores deste capítulo.

5.5. Uma visão geral do aproveitamento dos alunos

Dos sete alunos participantes da oficina, três: Daniel e José (do quarto ano) e Hilton (do terceiro ano), apresentaram um aproveitamento bastante satisfatório de todo o conteúdo trabalhado, e três: Bruno (do quarto ano), Marcelo e Ivana (do terceiro ano), apresentaram dificuldade em absorver esses conteúdos, em especial no que se refere às estruturas de repetição. Diana (do quarto ano), que deixou a oficina antes do término, também apresentou um bom aproveitamento dos conteúdos trabalhados nas aulas das quais participou.

Bruno, Marcelo e Ivana, ao final da oficina, conseguiam desenvolver programas simples, com poucos comandos, sem o uso de laços de repetição e nos quais a noção de atraso (*delay*) não se fizesse necessária. Por outro lado, Daniel, José e Hilton finalizaram a oficina demonstrando claramente a capacidade de construir programas com múltiplos laços de repetição condicional a partir da observação do "comportamento" apresentado pelos materiais de robótica utilizados em aula.

5.5.2 Construção de programas puramente sequenciais

As respostas apresentadas pelas crianças às atividades desenvolvidas durante as três primeiras aulas, onde somente **programas puramente sequenciais** foram trabalhados, já permitiram que se observasse, de forma bastante clara, a "divisão" da turma nos dois grupos citados anteriormente. Essa diferença de absorção dos conteúdos acabou por fazer com que as crianças que apresentavam maior facilidade conduzissem, cada uma em seu grupo, a construção dos programas que, nos primeiros encontros, era feita em computadores compartilhados.

Essa dinâmica acabou por fazer com que as crianças que apresentavam maior dificuldade ficassem sem espaço para explorar, à sua maneira e tempo, os conteúdos trabalhados, o que, possivelmente, contribuiu para ampliar ainda mais a distância entre os resultados apresentados por elas em relação às demais crianças. Uma vez esse problema tendo sido identificado, e com o objetivo de buscar minimizá-lo, a partir da sétima aula (quando já estava sendo trabalhada a estrutura de repetição contada), foi disponibilizado um computador por aluno, mantendo-se, no entanto, os alunos trabalhando em duplas ou trios.

As maiores dificuldades apresentadas pelas crianças nesse primeiro momento foram:

- Entendimento do efeito resultante do uso do tempo de atraso (*delay*). Essa dificuldade manteve-se presente durante toda a oficina, apresentando-se maior em alguns alunos e menor em outros.
- Construção mental da ideia completa do texto dos enunciados dos desafios propostos (leitura funcional⁹⁹), o que levava ao desenvolvimento de programas equivocados oriundos da leitura fragmentada desses textos.

Em relação às habilidades relacionadas ao Pensamento Computacional, já foi possível observar nos três primeiros encontros o exercício das seguintes habilidades:

- **Depuração e detecção sistemática de erros:** correção dos programas a partir da observação do resultado da sua execução nos materiais de robótica.
- **Capacidade de abstração:** construção mental da sequência de ações necessária para a obtenção de um determinado resultado e sua posterior materialização na forma de um programa de computador
- **Noções algorítmicas de fluxo de controle:** Sequenciamento adequado dos blocos (comandos) responsáveis por fazerem os materiais de robótica realizarem a sequência de ações esperada.

⁹⁹ Aquela que é utilizada para fins pragmáticos, em contextos cotidianos (RIBEIRO, 1997).

5.5.3 Uso da Estrutura de Repetição Contada

Ao final dos cinco encontros dedicados ao aprendizado da **estrutura de repetição contada**, Daniel, Hilton e José apresentavam-se claramente capazes de identificar quando e como utilizar este tipo de estrutura. O mesmo não se deu com Bruno, Marcelo e Ivana que, durante o tempo em que os computadores foram compartilhados, não conseguiam "impor-se" na tarefa de desenvolver os programas e, uma vez tendo sido disponibilizado um computador por aluno, embora apresentassem um empenho inicial na busca por construir os programas por conta própria, acabavam por finalizá-los olhando os programas dos colegas.

As maiores dificuldades apresentadas pelas crianças nessa segunda etapa foram:

- Entendimento de que apenas os comandos posicionados dentro do bloco *repetir* seriam repetidos o número de vezes indicado como parâmetro. A tendência das crianças era acreditar que todos os comandos posicionados após o comando *repetir* (ou seja, a parte superior do bloco *repetir*) seriam repetidos, independente desses comandos estarem posicionados dentro ou abaixo do bloco.
- Manteve-se a dificuldade com o uso do bloco esperar (*delay*). Essa dificuldade ficava evidenciada pelo fato de as crianças, de modo geral, não conseguirem, sem auxílio do professor, montar a sequência de ações necessárias para se observar um bloco piscar, ou seja: acender o LED → esperar um tempo (*delay*) → apagar o LED → esperar um tempo. Algumas colocavam apenas o bloco *acender LED*, outras apenas os blocos *acender* e *apagar LED*, outras colocavam o bloco *esperar* apenas depois do bloco *acender LED*. No decorrer da oficina, os alunos com melhor aproveitamento passaram a entender qual a sequência de comandos necessária para fazer o bloco piscar. As demais crianças finalizaram a oficina sem demonstrar esse entendimento.

As habilidades relacionadas ao Pensamento Computacional que puderam ser observadas nessa segunda etapa foram:

- **Depuração e detecção sistemática de erros:** correção dos programas a partir da observação do resultado da sua execução nos materiais de robótica.
- **Capacidade de abstração:** abstrair um programa de computador a partir da observação de eventos do mundo real, ou seja, construir mentalmente um algoritmo (uma sequência de ações que representa um evento observado). Esse tipo de habilidade podia ser constatado nos exercícios nos quais os alunos assistiam a vídeos com a caixinha mágica apresentando um determinado "comportamento" e, apenas com base nessa observação, construíam um programa que fizesse a caixinha funcionar conforme o observado.
- **Noções algorítmicas de fluxo de controle:** reordenação correta dos comandos presentes no programa para que este viesse a refletir mudanças ocorridas na ordem das ações realizadas pelos materiais de robótica, o que podia ser constatado com base no seguinte tipo de exercício:
 - Assistir a caixinha mágica apresentando um determinado comportamento como, por exemplo: piscar o LED vermelho 2 vezes, girar o motor DC por 5 segundos, parar o motor DC e então piscar o LED amarelo 3 vezes.
 - Construir um programa que faça a caixinha funcionar conforme o observado.
 - Observar a caixinha apresentando um segundo comportamento, no qual a diferença para o comportamento anterior é a ordem na realização das ações, como por exemplo: piscar o LED vermelho 2 vezes, piscar o LED amarelo 3 vezes, girar o motor DC por 5 segundos e depois parar o motor DC.
 - Reordenar os blocos no programa para que ele reflita a mudança de comportamento apresentada pela caixinha mágica.

5.5.4 Uso da Estrutura de Repetição Condicional

Concluídos os cinco encontros durante os quais foi trabalhada a **estrutura de repetição condicional**, Daniel, Hilton e José, como já mencionado no início desta seção, demonstravam com clareza a capacidade de construir programas com múltiplos laços de repetição condicional. Já Bruno, Ivana e Marcelo conseguiam desenvolver apenas programas puramente sequenciais bastante simples e que não necessitassem do entendimento da noção de atraso (*delay*).

A principal dificuldade apresentada nesta última etapa foi:

- Compreender o funcionamento "não determinístico" dos programas que fazem uso de uma estrutura condicional. Ou seja, os programas *puramente sequenciais*, ou que utilizem apenas *estruturas de repetição contada*, geram como resultado da sua execução um comportamento sempre idêntico do material de robótica programado (lembrando aqui que o Duinoblocks4Kids não possibilita o uso de variáveis e que o código nele construído é, por *default*, executado dentro de um "loop infinito"). Por exemplo, se for definido neste tipo de programa que o LED amarelo vai piscar 3 vezes e depois o LED azul cinco vezes, esses LEDs sempre piscarão esse número de vezes nessa sequência enquanto o programa permanecer rodando. No caso da *estrutura de repetição condicional*, quando uma determinada condição é satisfeita (um aumento na intensidade da luz, a aproximação de um objeto, etc.) o comportamento apresentado pelos materiais de robótica se modifica em tempo de execução. Assim, quando do uso do bloco *enquanto*, o número de repetições das ações associadas à uma determinada condição é definido por um fator externo, sendo essas ações repetidas tantas vezes quantas forem possíveis durante o tempo em que a condição dada se mantiver satisfeita, e não de forma constante como no caso da *estrutura de repetição contada*.

No que se refere ao Pensamento Computacional, as seguintes habilidades puderam ser observadas:

- **Depuração e detecção sistemática de erros:** correção dos programas a partir da observação do resultado da sua execução nos materiais de robótica.
- **Capacidade de abstração:** em adição às demonstrações do uso da capacidade de abstração anteriormente mencionadas, e igualmente evidenciadas nesta última etapa, pôde-se também observar o exercício desta habilidade nas ocasiões em que algumas crianças testavam na caixinha mágica programas destinados a um outro material de robótica, como, por exemplo, o peixe robô. Ou seja, elas percebiam que, embora os sensores e atuadores presentes nos dois materiais estivessem organizados de diferentes maneiras e por vezes apresentassem aspectos distintos, por se tratarem dos mesmos dispositivos, exibiriam o mesmo comportamento quando controlados pelo mesmo programa.
- **Noções algorítmicas de fluxo de controle:** soma-se às demonstrações anteriormente mencionadas (e também observadas durante esta etapa), a percepção das crianças (ainda que não necessariamente consciente) de que, com o uso da *estrutura de repetição condicional*, o fluxo do programa passa a ser determinado por uma condição externa variável, o que pode tornar esse fluxo "inconstante".
- **Lógica Condicional:** o uso da lógica condicional tornou-se evidente quando as crianças passaram a construir programas com múltiplos laços de repetição condicional a partir da observação do comportamento de materiais de robótica que faziam uso de mais de um sensor. Por exemplo: ao observarem o carrinho robótico exibindo o seguinte comportamento: desviando de obstáculos ao aproximar-se deles cerca de 15 cm (usando para isso um sensor de distância), andando para trás enquanto uma luz forte estivesse incidindo sobre ele (fazendo uso de um sensor de luz) e andando para frente enquanto nenhuma das condições

anteriores era satisfeita; as três crianças com melhor aproveitamento eram capazes de, com base na observação desse comportamento, construir, no Ambiente de Programação em Blocos, um programa que fizesse o carrinho funcionar exatamente da maneira observada, para o que se fazia necessário o uso de duas estruturas de repetição condicional (uma para cada sensor).

- **Decomposição estruturada de problemas:** algumas das práticas realizadas na oficina indicaram ser perfeitamente possível, mesmo em programas muito simples, trabalhar essa habilidade (ainda que de maneira bastante rudimentar). Por exemplo, ao mostrar-se para as crianças um material de robótica apresentando uma sequência de três ações distintas, tal como:

- Piscar constantemente um LED;
- Girar um motor DC lentamente quando um objeto se aproxima;
- Girar o motor com muita velocidade quando a temperatura estiver alta;

elas podem dividir o comportamento observado em partes, e buscar programar cada uma dessas partes separadamente, acrescentando pouco a pouco cada uma delas no código e testando o programa a cada nova etapa. No caso do programa exemplo acima apresentado, as crianças podem primeiro tentar fazer o LED piscar, depois fazer o motor DC girar lentamente enquanto algum objeto se mantiver próximo e, por fim, adicionar ao programa o código responsável por fazer o motor girar rapidamente enquanto a temperatura está alta.

Importante mencionar aqui que em uma atividade desenvolvida nessa última etapa, com o uso do braço robótico, onde noções abstratas de tempo e laços de repetição não se faziam necessárias, Ivana apresentou grande desenvoltura na realização da tarefa, qual seja: a construção de um programa puramente sequencial onde cada um dos blocos correspondia a uma ação concreta facilmente observável e reproduzível.

5.5.5 Atividades de fechamento

Por fim, após os treze encontros em que as diferentes estruturas de programação foram trabalhadas, foi realizada uma última aula, na qual as crianças desenvolveram atividades nas quais era necessária a aplicação dos conhecimentos e habilidades trabalhados na oficina. Dentre as atividades desenvolvidas, destaca-se a tarefa de desenhar um robô que fizesse uso de alguns atuadores e sensores que haviam sido trabalhados durante a aula e, após o desenho feito, explicar o funcionamento do robô projetado. Como resultado desta atividade, Daniel, Hilton e José desenharam robôs que utilizavam sensores e atuadores, descrevendo com bastante propriedade a função de cada um dos dispositivos utilizados e o comportamento assumido pelo robô quando as condições estabelecidas para os sensores eram satisfeitas. Além disso, Davi, sem que se houvesse solicitado, construiu corretamente, no Ambiente de Programação em Blocos, o programa para controlar o seu robô. Já Bruno, Davi e Ivana desenharam robôs que utilizavam apenas atuadores, tendo por vezes dificuldade para especificar qual a função de alguns desses atuadores em seus robôs.

5.5.6 Sobre as dificuldades apresentadas

Dentre as causas possíveis para a diferença de aproveitamento apresentada pelas crianças, é presumível que níveis diferentes de maturidade cognitiva possam ter contribuído para este quadro. Essa suspeita advém do fato de que as dificuldades apresentadas por Bruno, Davi e Ivana tinham relação direta com a aplicação de conceitos abstratos como a noção de tempo e de laços de repetição, bem como da compreensão de que os programas por elas desenvolvidos eram representações do comportamento que se esperava observar nos materiais de robótica sendo trabalhados. Ou seja, a maior dificuldade encontrada refere-se a questões relacionadas à capacidade de abstração e, uma vez que as crianças participantes da oficina encontravam-se todas, de acordo com Piaget, no estágio operatório concreto (ver capítulo 2.4), onde a capacidade de abstração ainda está em desenvolvimento, é possível que em algumas crianças ela já se encontrasse mais madura e em outras menos.

6 AVALIAÇÃO DA USABILIDADE DO AMBIENTE DE PROGRAMAÇÃO EM BLOCOS

A norma ISO 9241-11 (1998) define usabilidade como sendo “A medida pela qual um produto pode ser utilizado por usuários específicos para alcançar objetivos específicos com eficácia, eficiência e satisfação em um contexto específico de uso” (RAZA, CAPRET e AHMED, 2012, p.1109, tradução nossa).

Com o objetivo de avaliar-se a Usabilidade e *fun* do Ambiente de Programação em Blocos DB4K, lançou-se mão da combinação de três métodos empíricos (já mencionados anteriormente):

- Registro em vídeo das ações dos usuários;
- Observações de campo;
- Questionários.

O registro de ações dos usuários e as observações de campo são considerados métodos experimentais, enquanto que os questionários são considerados métodos de inquérito. Essa combinação de métodos experimentais e de inquérito na avaliação de usabilidade é proposta por Kostaras e outros (2010).

Os itens observados no registro das ações dos usuários e nas observações de campo, bem como os questionários respondidos pelos alunos encontram-se no apêndice B¹⁰⁰: Avaliação de Usabilidade e *Fun*, e tomaram como base:

- O método SUMI (*Software Usability Measurement Inventory*) (KIRAKOWSKI e CORBETT, 1993) desenvolvido pela UCC (University College Cork)
- O plano de teste do Software Roboeduc, utilizado por Renata P. Barros em sua dissertação de Mestrado "Evolução, Avaliação e Validação do Software RoboEduc" (BARROS, 2011).
- O questionário utilizado por Schafer e outros (2013) para a avaliação do LIT KIT, um Sistema Robótico de tecnologia mista que visa auxiliar as crianças a aprimorarem sua leitura com livros ilustrados.

¹⁰⁰ Disponível também em CD anexo à dissertação impressa e, futuramente, também no site do LlvRE (<http://www.nce.ufrj.br/ginape/livre/>)

- O *Smileyometer*, uma ferramenta proposta pelo *Fun Toolkit* (READ e MACFARLANE, 2006).

O *Fun Toolkit* (READ e MACFARLANE, 2006) foi desenvolvido em resposta a uma demanda por ferramentas que permitissem trazer as crianças para dentro do processo de avaliação dos produtos direcionados a elas. Isso porque, de acordo com os criadores desse instrumento: “[...] Adultos e crianças vivem em mundos diferentes, e, por este motivo, adultos podem não entender o que as crianças querem” (READ e MACFARLANE, 2006, p.1). O *Fun Toolkit* é formado por um conjunto de quatro ferramentas especiais, dentre elas o *Smileyometer* (Sorrisômetro): um tipo especial de escala Likert na qual os “itens de classificação” são acompanhados de *smilies* (carinhas com sorriso) (Figura 83).

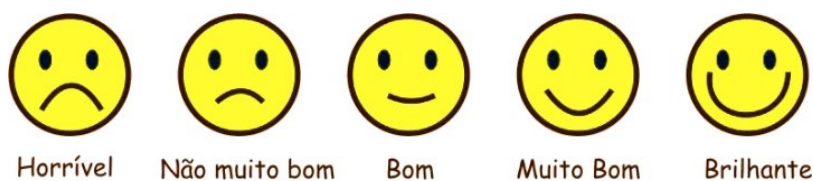


Figura 83: *Smileyometer*

Cabe aqui lembrar que o Ambiente de Programação em Blocos DB4K foi construído com base na biblioteca Blockly, que há muito vem sendo avaliada por sua equipe de desenvolvedores (FRASER, 2015). Desse modo, ele herdou uma série de características já devidamente testadas e validadas no que concerne à usabilidade de seus componentes.

6.1 RESULTADOS DAS OBSERVAÇÕES

Com base nas observações feitas em aula e nos vídeos gravados, pôde-se perceber, desde o primeiro contato com o ambiente, bastante facilidade no seu uso por parte dos alunos. As crianças conseguiram, com muita tranquilidade, localizar os blocos de programação no menu de acesso aos mesmos, selecionar esses blocos e dispô-los na área de trabalho. Do mesmo modo, as crianças não demonstraram maiores dificuldades em fazer a seleção de parâmetros nos blocos que apresentam

essa funcionalidade, bem como encaixar, desencaixar e excluir os blocos (fazendo para isso, uso da lixeira presente na área de trabalho do DB4K). No entanto, percebeu-se a necessidade da realização de modificações em alguns elementos de interface, bem como a inclusão de novas funcionalidades, conforme descrito a seguir.

6.1.1 Modificações realizadas nos elementos de interface e funcionalidades já existentes

Um problema observado em relação ao desenho dos blocos de programação deu-se com os blocos *acender LED* e *apagar LED* (Figura 84) que, devido à semelhança, acabavam por ser confundidos pelas crianças. Para solucionar esse problema modificou-se a cor do bloco *apagar LED*, aumentando assim sua distinção em relação ao bloco *acender LED*. Essa medida demonstrou-se adequada, uma vez que as crianças passaram a não confundir esses blocos a partir da sua realização.



Figura 84: Blocos *acender* e *apagar* LED.

Com base nas observações feitas, considerou-se útil também, em relação aos blocos, diferenciar os valores dos parâmetros que representam intensidade (rápido, médio ou de vagar; alto ou baixo; entre outros), por meio de elementos visuais, e não apenas na forma textual. Essa necessidade surgiu ao constatar-se que, uma vez conhecidos os blocos de programação e suas funcionalidades, as crianças passavam a identificá-los por seus ícones e cores, tendo essa possibilidade de reconhecimento visual apresentando-se especialmente útil às duas crianças do primeiro e segundo anos. No entanto, como a biblioteca Blockly não permite o uso de imagens nas *drop-down lists*, os parâmetros a serem utilizados nos blocos eram apresentados apenas na forma textual, o que dificultava sensivelmente às crianças menores a sua identificação. Para contornar esse problema, decidiu-se pela utilização de caracteres

unicode que possibilitassem também uma distinção imagética entre os parâmetros presentes nas *drop-down lists* (Figura 85).



Figura 85: Caracteres *unicode* utilizados para distinção dos parâmetros de intensidade

Outra modificação cuja necessidade se fez perceber está relacionada aos botões utilizados para compilação¹⁰¹ e execução dos programas (Figura 86), herdados do Ardublockly, ambiente sobre o qual foi desenvolvido o DB4K. Como as crianças não trabalham com o código *Wiring* e os programas construídos no ambiente não geram erros de compilação, os botões utilizados para compilar o código *Wiring* ou abri-lo no IDE Arduino tornavam-se dispensáveis. Além de não serem necessários, a presença desses botões causava confusão nos alunos que, algumas vezes, clicavam o botão errado e ficavam aguardando a execução do programa na caixinha. Por conta dessas questões, esses botões foram retirados da interface.

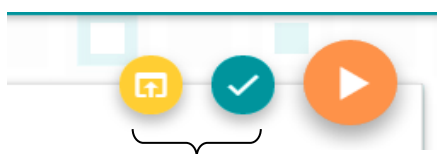


Figura 86: Botões flutuantes para abrir o código no IDE e compilar o código *Wiring*

6.1.2 Inclusão de novas funcionalidades

Sempre que terminavam um programa e davam início a outro, as crianças precisavam apagar todos os blocos presentes na área de trabalho, enviando-os para a lixeira, e depois apagar o nome do programa na área utilizada para este fim. No

¹⁰¹ Um compilador, grosso modo, é um software que varre o código do programa escrito pelo programador à procura de erros e, posteriormente, gera uma versão deste programa em uma “linguagem” que o computador possa entender. Erros de compilação são erros relacionados à escrita do código do programa.

entanto, muitas vezes, ao começarem um novo programa, as crianças esqueciam de apagar o nome do programa anterior, e acabavam desse modo, salvando o novo programa com mesmo nome do antigo (ao qual era adicionado automaticamente um número sequencial, o que acontece por padrão nos navegadores de internet). Isso fazia com que, além de causar confusão em relação aos nomes dos programas salvos, as crianças pudessem vir a criar o entendimento, ao iniciar uma nova tarefa, de que estavam continuando o trabalho anterior e não dando início a um novo programa.

Para evitar esses problemas, decidiu-se pela inclusão de um elemento de interface que possibilitasse às crianças “zerar” o ambiente de programação, ou seja, dar início a um novo programa, apagando todos os blocos presentes na área de trabalho e limpando o nome do projeto no local utilizado para este fim. Para o “disparo” dessa função optou-se pela inclusão de um item na barra de navegação (Figura 87).



Figura 87: Item de menu Novo Programa

Percebeu-se também a necessidade da inclusão de um elemento de interface que permitisse cessar um programa previamente enviado para a placa Arduino, de modo que as crianças pudessem fazer os sensores e atuadores da caixinha mágica “pararem” enquanto construíam um novo programa no ambiente. Essa funcionalidade apresentou-se como sendo importante pois, como alguns atuadores geram sons ou ruídos, após um programa ser enviado para execução na caixinha, esses sons e ruídos continuavam sendo emitidos até que as crianças concluíssem uma nova tarefa e enviassem um novo código para ser executado, o que por vezes levava um longo período de tempo. Além disso, alguns atuadores possuem partes mecânicas que sofrem desgaste com o uso, e deixá-los em funcionamento entre a execução de um programa e outro provocaria uma redução desnecessária na vida útil desses dispositivos. Para solucionar esse problema, foi criado um botão *finalizar execução do programa na placa* (Figura 88), posicionado ao lado do *botão carregar programa na placa*.

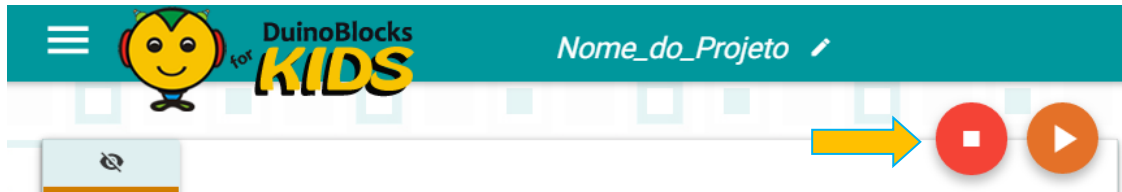


Figura 88: Botão Finalizar Execução do Programa

6.2 RESULTADOS DOS QUESTIONÁRIOS

Os questionários que, como visto, faziam uso de uma escala Likert adaptada (*smileyometer*), apresentaram resultados coerentes em relação às avaliações feitas a partir das observações em aula e dos vídeos capturados. No primeiro questionário, aplicado após o primeiro dia de uso do ambiente de programação (terceira aula da oficina), as crianças assinalaram o seu uso como sendo fácil ou muito fácil. Já em uma avaliação posterior, feita após a nona aula da oficina, duas das crianças assinalaram como sendo "mais ou menos fácil" o uso do ambiente (Figura 89 e Figura 90).

1	Você achou fácil ou difícil encontrar os blocos que precisava para fazer seu programa?	Muito Difícil	Difícil	Mais ou Menos	Fácil	Muito Fácil
2	Os ícones dos blocos ajudaram você a encontrar o bloco que você precisava?	Não	Mais ou Menos	Sim		
3	Você achou fácil ou difícil encaixar e desencaixar os Blocos?	Muito Difícil	Difícil	Mais ou Menos	Fácil	Muito Fácil
4	Você achou fácil ou difícil salvar e abrir um projeto?	Muito Difícil	Difícil	Mais ou Menos	Fácil	Muito Fácil

Figura 89: Questionário aplicado após a primeira aula na qual foi utilizado o ambiente de programação (terceira aula da oficina)

Deve-se levar em consideração, ao se analisar as respostas dadas, que, na segunda avaliação, as crianças estavam trabalhando na construção de programas

mais complexos, podendo ter havido, por parte dos alunos, uma confusão entre a dificuldade de usar o ambiente de programação (manipular os elementos de interface e fazer uso de determinadas funcionalidades) e a dificuldade de construir os programas solicitados (entendimento da lógica de programação). Essa suposição é resultante do fato de que as crianças que indicaram o ambiente como sendo "mais ou menos fácil de usar" são as mesmas que apresentaram uma maior dificuldade no aprendizado de programação. Além disso, nessas mesmas avaliações, embora o uso do ambiente tenha sido indicado como "mais ou menos fácil", as perguntas sobre a manipulação de determinados elementos de interface, e o uso de algumas funcionalidades específicas, foram assinaladas como "muito fácil" (Figura 90).

1	O que você está achando do DuinoBlocks4Kids?	 Ruim	 Mais ou menos	 Bom	 Legal	 Muito Legal
2	O DuinoBlocks4Kids é fácil ou difícil de usar?	 Difícil	 Um pouco Difícil	 Mais ou Menos	 Fácil	 Muito Fácil
3	Você está achando fácil ou difícil encontrar os blocos que precisa para fazer seu programa?	 Difícil	 Um pouco Difícil	 Mais ou Menos	 Fácil	 Muito Fácil
4	Os desenhos que têm nos blocos ajudam você a encontrar o bloco que você está procurando?	 Não	 Mais ou Menos	 Sim		
5	Você acha fácil ou difícil salvar um programa?	 Difícil	 Um pouco Difícil	 Mais ou Menos	 Fácil	 Muito Fácil
6	O que você está achando de programar a caixinha mágica?	 Ruim	 Mais ou menos	 Bom	 Legal	 Muito Legal
7	O que você achou da atividade de ver o vídeo e construir o programa para a caixinha funcionar igual ao vídeo?	 Ruim	 Mais ou menos	 Bom	 Legal	 Muito Legal

Figura 90: Questionário aplicado após a nona aula da oficina

7 AVALIAÇÃO DOS RECURSOS PEDAGÓGICOS

Para a avaliação dos recursos pedagógicos utilizados durante a oficina, quais sejam: as brincadeiras, as narrativas e os materiais de robótica; fez-se uso dos mesmos métodos utilizados para a avaliação da usabilidade do Ambiente de Programação em Blocos: registro em vídeo das ações dos usuários, observação de campo e questionários. A Figura 91 mostra um exemplo dos questionários aplicados durante a oficina.

O que você achou de programar a Caixinha Mágica?	 Ruim	 Mais ou menos	 Bom	 Legal	 Muito Legal
O que você achou da atividade de ver os círculos coloridos que o professor entregou mudando de cor na caixinha mágica?	 Ruim	 Mais ou menos	 Bom	 Legal	 Muito Legal
O que você achou de da atividade de pintar os círculos coloridos e depois colocá-los na caixinha pra mudar de cor?	 Ruim	 Mais ou menos	 Bom	 Legal	 Muito Legal
O que você achou de montar o circuito com os LEDs?	 Ruim	 Mais ou menos	 Bom	 Legal	 Muito Legal

Figura 91: Exemplo de questionário de avaliação dos recursos pedagógicos

7.1 AS BRINCADEIRAS

Como visto em capítulos anteriores, para a introdução de todos os temas referentes à programação de computadores trabalhados na oficina, lançou-se mão de algumas brincadeiras baseadas na técnica de computação desplugada (ver capítulo 2.8).

A brincadeira do robô humano, na qual as crianças programavam o professor ou os colegas para executarem determinadas tarefas, bem como seguiam um programa apresentado pelo professor, foi muito bem aceita todas as vezes em que foi aplicada, o que pôde ser percebido pelo engajamento das crianças durante a sua realização. Outras brincadeiras, como a da varinha mágica e do robô dançante (ver apêndice A¹⁰², aulas 3 e 9), também foram bastante bem-sucedidas. Apenas a brincadeira das estrelas e maçãs (ver apêndice A, aula 4) não apresentou um resultado muito bom em relação ao engajamento das crianças.

Não é possível afirmar se a realização dessas brincadeiras teve influência direta no aprendizado dos conteúdos trabalhados. No entanto, com base na resposta dos alunos à sua realização, elas apresentaram-se nitidamente válidas como veículo de motivação e de apoio à construção de um ambiente de aprendizagem divertido, afetuoso e acolhedor.

7.2 AS NARRATIVAS

Haviam sido planejados inicialmente quatro encontros nos quais seriam utilizadas narrativas como recurso pedagógico de apoio ao aprendizado dos temas trabalhados na oficina. No entanto, com a necessária reformulação das aulas, em decorrência do ritmo de aprendizado dos alunos, pôde-se trabalhar, de forma mais elaborada, apenas a narrativa referente ao peixe robô (ver capítulo 4.2).

As crianças ficaram bastante impressionadas com a história e com as imagens que eram apresentadas pelo projetor, como a caverna submarina e o peixe monstro mutante emissor de luz. No momento da “dramatização” da história, na qual, com a sala às escuras, uma criança munida de uma lanterna fazia o papel do peixe monstro (que ativava com sua luz o alarme do peixe robô), e as demais crianças faziam o papel dos mergulhadores, os alunos ficaram bastante engajados: corriam animadíssimos pela sala buscando um local para se esconder do peixe monstro sempre que o buzzer do peixe disparava e, logo que o alarme cessava, deixavam seus esconderijos e voltavam a se movimentar pela sala (imitando os movimentos de um mergulhador procurando pelo tesouro).

¹⁰² Disponível também em CD anexo à dissertação impressa e, futuramente, também no site do LIVRE (<http://www.nce.ufrj.br/ginape/livre/>)

Não era objetivo deste trabalho avaliar a eficácia do uso das narrativas no processo de ensino-aprendizagem, visto que, conforme apresentado no capítulo 2.9, muitos estudos já comprovaram o valor dessa ferramenta como facilitador nesse processo. Nos interessava sim observar aqui a adequação do uso deste recurso em atividades de aprendizado de programação por meio da Robótica Educacional, o que ficou bastante evidenciado durante a realização da atividade com o peixe robô: dentre todas as atividades desenvolvidas na oficina, essa foi a que teve o poder de manter as crianças mais atentas, focadas e curiosas.

7.3 OS MATERIAIS DE ROBÓTICA

7.3.1 A caixinha mágica

Como mencionado no capítulo 3, a caixinha mágica foi desenvolvida com o objetivo de viabilizar a construção de programas mais elaborados necessários para o controle de circuitos com um grau de complexidade que seria de difícil montagem para crianças com idade entre oito e dez anos.

Além disso, ter sempre à disposição um circuito previamente montado, composto por um conjunto de atuadores e sensores programáveis por meio do Ambiente de Programação em Blocos, permitia que se realizasse, com bastante facilidade, atividades de desenvolvimento de programas a partir da observação de um determinado “comportamento” apresentado por esse conjunto de dispositivos (ver capítulo 5.2.5). Esse tipo de atividade demonstrou ser bastante pertinente como ferramenta de apoio ao exercício da capacidade de abstração, uma vez que permitia às crianças trabalharem a habilidade de observar um “processo” do mundo real e, a partir desta observação, abstrair esse processo na forma de um programa de computador.

Outra indicação da validade da caixinha mágica como veículo para o exercício da capacidade de abstração pôde ser observada quando, por iniciativa própria, as crianças passaram a testar os programas feitos para controlar outros materiais de robótica, com fins mais específicos, a partir dos dispositivos de uso geral presentes na caixinha. Por exemplo, conforme visto nos capítulos 5.3.2.2 e 5.5.4, quando da programação do peixe robô, algumas crianças criavam seus programas objetivando

controlar o peixe, mas, o testavam inicialmente na caixinha mágica, fazendo, com base nesses testes, as alterações necessárias no código quando o resultado observado não era o esperado. Essa dinâmica nos permite, como já mencionado, perceber que essas crianças conseguiam visualizar na caixinha mágica o comportamento do peixe robô, e entender, ainda que de forma não totalmente consciente que, embora organizados de uma outra maneira, e por vezes apresentando aspectos distintos, por se tratarem dos mesmos dispositivos exibiriam o mesmo comportamento quando controlados pelo mesmo programa.

Em todas as atividades nas quais foi utilizada, a caixinha apresentou-se como sendo de fácil manipulação e bastante atrativa. Era nítida a alegria das crianças a cada dispositivo que conseguiam fazer funcionar a partir dos programas desenvolvidos no Ambiente de Programação em Blocos: elas batiam palmas, chamavam os colegas para ver o que haviam conseguido fazer, além de apresentarem uma série de outras demonstrações de contentamento.

Esse conjunto de fatores nos permite qualificar a caixinha mágica como sendo um “facilitador lúdico” bastante apropriado para o aprendizado de programação por meio da Robótica Educacional.

7.3.2 Os robôs de garrafa pet

O objetivo principal da utilização dos robôs de garrafa pet, como o peixe robô, o morcego robô e o cachorro robô, era o de criar contextos onde os dispositivos trabalhados inicialmente na caixinha mágica pudessem ser aplicados, uma vez que essa contextualização, como mencionado no capítulo 2.9, cria “[...] um arcabouço no qual os alunos podem acomodar os novos conhecimentos (e assim ampliar sua retenção e entendimento)” (SZURMAK e THUNA, 2013, p.550). Como as crianças, de modo geral, possuem grande atração pelos contos de fadas e histórias de ficção, decidiu-se por utilizar este tipo de temática para contextualizar os robôs utilizados na aula, o que se apresentou como sendo uma estratégia bastante acertada, uma vez que os alunos apresentaram grande interesse pelas atividades em que esses robôs foram trabalhados. Além disso, o fato de algumas crianças utilizarem, como mencionado na seção anterior, a caixinha mágica como base de teste de programação desses materiais, indica que elas reconheciam os dispositivos presentes nos robôs

como aplicações, em outro contexto, dos atuadores e sensores que haviam previamente aprendido a programar.

7.3.3 O carro robô e o braço robótico

Além dos robôs de garrafa pet, desenhados e produzidos especificamente para a oficina, e que eram programados com o uso dos mesmos blocos utilizados para programar os dispositivos presentes na caixinha mágica, foram também utilizados, durante as atividades, um carrinho robô e um braço robótico que possuíam blocos específicos para sua programação.

O carrinho robótico demonstrou ser bastante versátil, possibilitando ser facilmente equipado com diferentes sensores e atuadores, além de poder ser ornado com uma variedade de materiais, como no caso das asas do morcego vampiro (ver apêndice A¹⁰³, aula 12), o que torna possível a sua utilização em uma variedade de contextos a partir dos quais pode-se desenvolver uma diversidade de programas de controle.

Foi possível também observar, durante as atividades, que o conjunto de dispositivos utilizados para fazer o carrinho andar (rodas, motores e ponte H¹⁰⁴) poderiam ser utilizados em outros robôs, como no caso do cachorro robô de garrafa pet (Figura 92).

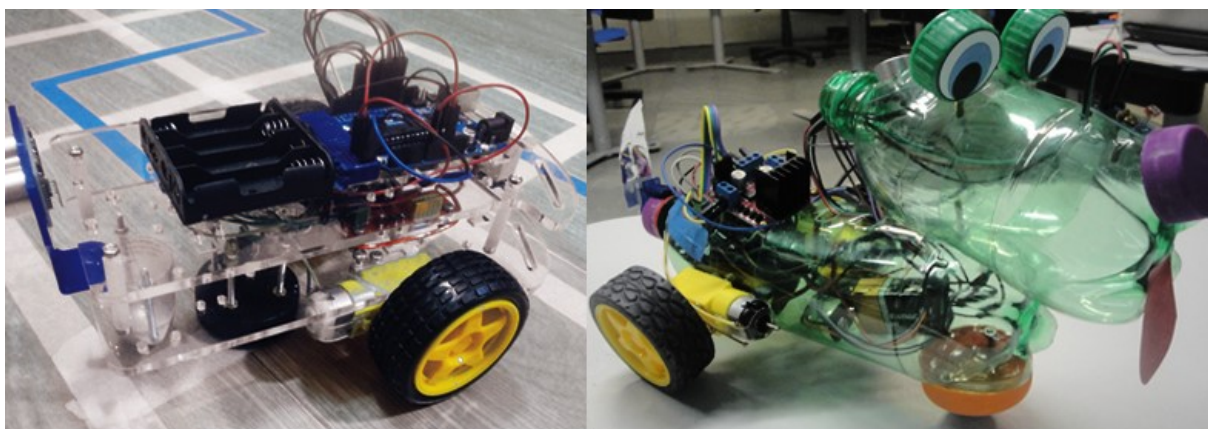


Figura 92: Generalização do uso dos dispositivos responsáveis pela locomoção do carrinho robótico

¹⁰³ Disponível também em CD anexo à dissertação impressa e, futuramente, também no site do LIVRE (<http://www.nce.ufrj.br/ginape/livre/>)

¹⁰⁴ Grosso modo, uma ponte H é um circuito eletrônico que permite, dentre outras coisas, controlar o sentido do giro de motores de corrente contínua (motor DC) a partir de sinais elétricos.

Esta adaptação do uso das rodas do carrinho para outro fim possibilitou aos alunos observarem uma generalização do uso desses dispositivos, à medida que, ao observarem que os dois robôs utilizavam o mesmo conjunto de dispositivos para se locomoverem, compreendiam ser possível utilizar o mesmo conjunto de blocos para programar a locomoção de ambos.

Assim como o carrinho robótico, o braço robótico também atraiu bastante a atenção das crianças, que, de modo geral, demonstraram muito interesse em desenvolver os programas para controlá-lo.

Havia a intenção inicial de se utilizar o braço robótico para que as crianças pudessem exercitar o entendimento de que um mesmo problema pode ser solucionado de diferentes maneiras e, que uma solução pode ser mais “eficiente” do que a outra. Esse exercício seria possível pois, dependendo da configuração do tabuleiro (posição dos círculos verde, alaranjado e branco, e dos peões verde e alaranjado (ver capítulo 5.3.5.1, Figura 63)) é possível programar diferentes sequências, com diferentes números de movimentos, para trocar os peões de lugar colocando-os nos círculos corretos. No entanto, uma vez que o braço robótico foi utilizado em uma única ocasião, não foi possível ser feita uma avaliação, ainda que preliminar, da abrangência das suas potencialidades como recurso pedagógico para o aprendizado de programação.

Apesar disso, os resultados apresentados pela aluna Ivana, quando da realização da atividade de programação do braço robótico (ver capítulo 5.3.5.2), indicam que seu uso possa ser interessante em exercícios de introdução ao aprendizado de programação com robótica para crianças cuja maturidade cognitiva exija a realização de atividades voltadas ao uso do pensamento concreto (ver capítulo 2.4).

8 CONSIDERAÇÕES FINAIS

Pouquíssimos estudos de caso terminarão exatamente como foram planejados. Inevitavelmente, você terá que fazer pequenas, quando não grandes, alterações, que variam da necessidade de tomar uma direção inesperada (uma alteração potencialmente pequena) à necessidade de identificar um novo "caso" para um estudo (alteração potencialmente grande). O pesquisador habilidoso deve lembrar do propósito inicial da investigação, mas aí, se ocorrerem eventos imprevistos, ele provavelmente desejará alterar os procedimentos ou os planos. (YIN, 2001, p.83),

No início deste trabalho, tinha-se por objetivo o desenvolvimento e avaliação da usabilidade de um Ambiente de Programação Visual baseado em Tecnologia Livre para o ensino de conceitos básicos de programação a crianças do Ensino Fundamental I por meio da Robótica Educacional. No entanto, durante a idealização e realização da oficina destinada aos testes de usabilidade, percebeu-se a oportunidade de uma investigação mais ampla: a busca por hipóteses acerca da possibilidade do desenvolvimento e exercício de determinadas habilidades do Pensamento Computacional em crianças do Ensino Fundamental I (estudantes de escolas públicas e oriundas de comunidades de baixa renda) a partir do aprendizado de conceitos básicos de programação por meio de recursos baseados em Tecnologia Livre e materiais de baixo custo. Assim, o propósito central deste estudo deixou de ser uma ferramenta e sua aplicabilidade para transformar-se em uma investigação mais abrangente, o que fez com que os objetivos do trabalho acabassem por se configurar conforme o apresentado no capítulo 1.2.

No que se refere ao objetivo geral da pesquisa, qual seja, a investigação acerca da possibilidade do desenvolvimento e exercício de habilidades do Pensamento Computacional em crianças do Ensino Fundamental I, os dados coletados indicam a viabilidade de se trabalhar, com crianças na faixa etária e pertencentes a contextos socioeconômicos semelhantes aos dos sujeitos participantes deste estudo, as seguintes habilidades:

- Capacidade de Abstração (mais especificamente a abstração empírica).
- Compreensão de fluxos de controle.
- Depuração e detecção sistemática de erros.
- Uso da lógica condicional.
- Decomposição estruturada de problemas.

(Ver introdução do capítulo 5)

A demonstração dessas habilidades, em conjunto com outras competências manifestadas pelas crianças durante a realização das atividades propostas na oficina, como a capacidade de reaproveitar código, a percepção da ideia de processamento, a construção mental das ações necessárias para a realização de uma determinada tarefa, o entendimento de que os mesmos comandos organizados de diferentes maneiras podem levar a diferentes resultados e de que um mesmo problema pode apresentar diferentes soluções, ajudam a reforçar as hipóteses levantadas no capítulo 2.4, acerca da possível existência de uma relação direta entre determinadas características cognitivas de crianças operatórias concretas e a realização de algumas atividades relacionadas à programação de computadores.

Das sete crianças participantes do estudo, três demonstraram ser capazes de desenvolver atividades que demandassem as habilidades acima mencionadas, três apresentaram dificuldades nesse sentido, e uma desistiu da oficina, tendo, no entanto, apresentado-se capaz de desenvolver todas as atividades propostas nas aulas em que esteve presente. A maior dificuldade encontrada pelas crianças refere-se a questões relacionadas à capacidade de abstração, como a utilização da noção de tempo e de laços de repetição, e a construção mental (ainda que realizada a partir de referenciais concretos) da sequência de ações necessárias para a obtenção de um determinado resultado e sua posterior materialização na forma de um programa de computador.

A hipótese aqui levantada é a de que essa dificuldade seja decorrente de questões relativas à maturidade cognitiva, uma vez que os sujeitos observados encontravam-se todos no estágio operatório concreto, no qual a capacidade de abstração ainda está em desenvolvimento. Além disso, uma vez que a maturidade cognitiva é dependente de fatores biológicos, educacionais e sociais, é possível que, em relação a certos aspectos, algumas das crianças observadas apresentassem uma maturidade pré-operatória, período no qual ainda não se consegue, de modo geral, realizar operações mentalmente (ver capítulo 2.4).

Em relação aos objetivos específicos, todos foram devidamente cumpridos, tendo-se que:

- a) O Ambiente de Programação Visual em Blocos DB4K foi desenvolvido e encontra-se plenamente funcional, tendo as crianças participantes deste estudo demonstrado bastante facilidade no seu manejo (ver capítulo 6). Além de ter sido utilizado nesta pesquisa, o DB4K foi também empregado em um *Creathon* Pedagógico realizado no NCE/UFRJ com

professores da rede pública de ensino do município do Rio de Janeiro, e em aulas de Robótica Educacional com crianças de seis a quatorze anos (com dificuldades de aprendizado ou comportamento) do CEREI (Centro de Referência de Educação Infantil) da cidade de São João del-Rei. De acordo com os pesquisadores do Núcleo de Robótica e Tecnologias Assistivas da UFSJ¹⁰⁵ (CyRoS), responsáveis pelas aulas desenvolvidas no CEREI, as crianças (dentre elas algumas que apresentam TDAH¹⁰⁶, dislexia e/ou dificuldades de aprendizagem e de alfabetização) demonstraram mais facilidade em utilizar o DB4K do que as ferramentas empregadas nas aulas anteriores, como o IDE Arduino¹⁰⁷ e o S4A¹⁰⁸. A equipe do CyRoS está adaptando suas aulas para utilizar o Duinoblocks4Kids com o kit CyRoS de robótica de baixo custo e pretende verificar a eficácia do uso do DB4K nas turmas com Transtorno do Neurodesenvolvimento¹⁰⁹ como a Síndrome de Down, o Transtorno do Aspecto Altista e a Dislexia.

- b) Desde setembro de 2015, além da versão *client-side* (utilizada nessa pesquisa e nos trabalhos acima mencionados), o DB4K conta também com uma versão online integrada ao portal LabVad, o que possibilita a crianças, e também jovens e adultos sem conhecimentos em programação de computadores, a utilização dos laboratórios remotos do LabVad para darem seus primeiros passos rumo a esse aprendizado.
- c) Foram elaborados quatorze planos de aula (ver Apêndice A¹¹⁰), com especificação detalhada dos objetivos, conteúdos, material necessário e desenvolvimento metodológico (onde são descritas as atividades a serem realizadas e os procedimentos a serem adotados). Dentre as atividades desenhadas, foram incluídas brincadeiras baseadas na técnica de computação desplugada que, conforme descrito no capítulo 7, apresentaram-se nitidamente válidas como veículo de motivação e de apoio à construção de um ambiente de aprendizagem divertido, afetuoso

¹⁰⁵ Universidade Federal de São João del-Rei.

¹⁰⁶ Transtorno do Déficit de Atenção com Hiperatividade

¹⁰⁷ <https://www.arduino.cc/>

¹⁰⁸ <http://s4a.cat/>

¹⁰⁹ “[...] grupo de condições observadas durante o período de desenvolvimento, manifestadas precocemente, com frequência antes da idade escolar e caracterizadas por déficits do desenvolvimento que ocasionam prejuízos sociais, acadêmicos, sociais e ocupacionais” (ASSUMPÇÃO JR, 2014, p.87)

¹¹⁰ Disponível também em CD anexo à dissertação impressa e, futuramente, também no site do LIVRE (<http://www.nce.ufrj.br/ginape/livre/>)

e acolhedor; e atividades com narrativas, que demonstraram ter potencial para manter as crianças mais atentas, focadas e curiosas.

- d) Por fim, foram desenvolvidos (com o uso de sucata, de materiais recicláveis ou de baixo custo) um conjunto de materiais de robótica (caixinha mágica, peixe robô, morcego robô e cachorro robô (ver capítulo 3.1)), para uso com o Ambiente de Programação em Blocos, que se revelaram aparatos lúdicos bastante apropriados para o aprendizado de programação por meio da Robótica Educacional.

Pode-se com isso concluir que o Ambiente de Programação em Blocos DB4K, em conjunto com os materiais de robótica e as atividades didáticas propostas por esta pesquisa, propicia o desenvolvimento e exercício de habilidades do Pensamento Computacional em crianças do Ensino Fundamental I, residentes em comunidades de baixa renda, a partir do aprendizado de conceitos básicos de programação.

8.2 PRINCIPAIS CONTRIBUIÇÕES

A abstração é o processo mais importante do Pensamento Computacional (WING, 2011), e a programação de computadores uma importante ferramenta de apoio ao desenvolvimento e exercício de uma série de habilidades desta forma de pensamento, bem como uma competência fundamental da Ciência da Computação (GROVER e PEA, 2013). De acordo com Dijkstra (1972, p.864, tradução nossa) “a exploração efetiva dos seus poderes de abstração deve ser considerada uma das atividades mais vitais de um programador competente”.

Há hoje um crescente movimento no sentido de se introduzir o ensino de programação já nos primeiros anos do Ensino Fundamental e, até mesmo, a partir da Educação Infantil. Torna-se assim imprescindível a realização de pesquisas acerca da relação existente entre a capacidade de abstração de crianças nessa faixa etária e as habilidades necessárias ao aprendizado de conteúdos relacionados à programação de computadores, de modo que esses conteúdos sejam abordados no momento mais propício e do modo mais adequado. Uma importante contribuição deste trabalho é chamar a atenção para este fato e contribuir com a investigação de hipóteses acerca desse tema.

Outra contribuição significativa é o kit DB4K, cujo conjunto de materiais pedagógicos demonstrou-se adequado ao desenvolvimento e exercício de algumas habilidades cognitivas do Pensamento Computacional por meio da programação de computadores. E, por ser baseado em Tecnologia Livre associada a materiais recicláveis ou de baixo custo, o kit apresenta-se economicamente mais acessível do que as soluções proprietárias atualmente disponíveis no mercado, podendo contribuir assim para uma maior democratização do acesso ao aprendizado de programação de computadores nos primeiros anos do Ensino Fundamental.

8.3 PUBLICAÇÕES

- Queiroz, Rubens L.; Sampaio, Fábio F. **DuinoBlocks4Kids: Um ambiente de programação em blocos para o ensino de conceitos básicos de programação a crianças do Ensino Fundamental I por meio da Robótica Educacional**. Anais do XXXVI Congresso da Sociedade Brasileira de Computação (XXIV WEI - Workshop sobre Educação em Computação). Porto Alegre. 2016. p.2086-2095. Disponível em: <http://ebooks.pucrs.br/edipucrs/anais/csbcc/assets/2016/wei/10.pdf> >. Acesso em: 26 jan 2017.
- Queiroz, Rubens L.; Sampaio, Fábio F.; dos Santos, Mônica P. **DuinoBlocks4Kids: Ensinando conceitos básicos de programação a crianças do Ensino Fundamental I por meio da Robótica Educacional**. Anais dos Workshops do V Congresso Brasileiro de Informática na Educação (II WAlgProg - Workshop de Ensino em Pensamento Computacional, Algoritmos e Programação). Uberlândia. 2016. p.1169-1178. Disponível em: <<http://br-ie.org/pub/index.php/wcbie/article/view/7042>>. Acesso em: 26 jan 2017.

8.4 TRABALHOS FUTUROS

- Criação de um site para disponibilização do Ambiente de Programação DB4K para download, tutoriais para o uso do ambiente, planos de aula, videotutoriais sobre a montagem dos materiais de robótica, bem como a disponibilização do código fonte do DB4K no GitHub¹¹¹.
- Desenvolvimento de novas oficinas para a realização de investigações mais detalhadas acerca das hipóteses levantadas por esse estudo.
- Desenvolvimento de oficinas mais longas, e em espaços destinados à realização de trabalhos manuais, de modo que se possa buscar avaliar os impactos da realização das seguintes atividades no aprendizado das crianças:
 - Auxílio na confecção de robôs de material reciclável mais elaborados (como o peixe robô).
 - Desenvolvimento de robôs, ficando a cargo das crianças, inclusive, a definição dos contextos nos quais os robôs projetados estariam inseridos (construção de narrativas próprias).
 - Prática rotineira de montagem de circuitos eletrônicos.
- Realização de um estudo mais amplo e detalhado acerca das teorias de desenvolvimento cognitivo, na busca pela identificação de relações existentes entre as características cognitivas de crianças entre seis e dez anos de idade e as habilidades necessárias para o aprendizado de programação de computadores, de modo a se colaborar com a construção de um arcabouço teórico mais extenso que possa contribuir com um aprofundamento das investigações acerca do desenvolvimento do Pensamento Computacional em crianças nessa faixa etária.
- Desenvolvimento de soluções, baseadas no kit DB4K, voltadas à deficientes visuais.
- Realização de estudos do uso do kit envolvendo crianças com dificuldades cognitivas.

¹¹¹ <https://github.com>

REFERÊNCIAS

- AGUILAR, L. J. **Fundamentos de Programação**: Algoritmos, estruturas de dados e objetos. 3. ed. Porto Alegre: AMGH, 2008. 706 p.
- ALFARO, S. A. **Robôs em Projetos Tecnológicos**. Anais da 58ª Reunião Anual da SBPC. Florianópolis: [s.n.]. 2006. Disponível em: <http://www.sbpnet.org.br/livro/58ra/atividades/TEXTOS/texto_884.html>. Acesso em: 12 jan 2017.
- ALIMISIS, D. Educational Robotis: Open questions and new challenges. **Themes in Science and Technology Education**, v. 6, n. 1, p. 63-71, 2013. Disponível em: <<http://earthlab.uoi.gr/theste/index.php/theste/article/view/119>>. Acesso em: 12 jan 2017.
- ALIMISIS, D. et al. **Robotics & Constructivism in Education**: the TERECoP project. Proceedings of the 11th European Logo Conference(Eurologo 2007. Bratislava: [s.n.]. 2007. p. 19-24. Disponível em: <http://users.sch.gr/adamopou/docs/syn_eurologo2007_alimisis.pdf>. Acesso em: 12 jan 2017.
- ALLEN, K. Using drones to save lives in Malawi. **BBC News**, mar 2016. Disponível em: <<http://www.bbc.com/news/world-africa-35810153>>. Acesso em: 29 jan 2017.
- ALVES, R. M. **Duinoblocks: Desenho e Implementação de um Ambiente de Programação Visual para Robótica Educacional**. Dissertação (Mestrado em Informática). Universidade Federal do Rio de Janeiro. Rio de Janeiro. 2013. 111 f.
- ARNOLD, K.; JAMES, G.; DAVID, H. **A linguagem de programação Java**. 4. ed. Porto Alegre: Bookman, 2009. 800 p.
- ASIMOV, I. **Eu, robô**. Tradução de Aline Storto Pereira. São Paulo: Aleph, 2014. 320 p.
- ASSUMPÇÃO JR, F. B. **Psiquiatria da Infância e da Adolescência**: Casos Clínicos. Porto Alegre: Artmed Editora, 2014. 334 p.
- AUGUSTO, C. A. et al. Pesquisa Qualitativa: rigor metodológico no tratamento da teoria dos custos de transação em artigos apresentados nos congressos da Sober (2007-2011). **Revista de Economia e Sociologia Rural**, Brasília, v. 51, n. 4, p. 745-764, out-dez 2013.
- AUTÔMATO. In: Instituto Antonio Houaiss. **Dicionário Eletrônico HOUAISS da língua Portuguesa**, versão 1.0. Editora Objetiva, 2001.
- BAILEY, K. **Methods of Social Research**. 4. ed. New York: The Free Press, 1994. 592 p.
- BANZI, M. How Arduino is open-sourcing imagination. **TEDGlobal**, 2012. Disponível em: <https://www.ted.com/talks/massimo_banzi_how_arduino_is_open_sourcing_imagination/transcript?language=pt-br>. Acesso em: 29 jan 2017.

BARBERO, A.; DEMO, B. G. **The art of programming in a Technical Institute after the Italian secondary school reform**. Proceedings ISSEP 2011. Bratislava: [s.n.]. 2011. Disponível em: <<http://pubshop.bmbf.gv.at/download.aspx?id=573>>. Acesso em: 12 jan 2017.

BARCELOS, T. S.; SILVEIRA, I. F. **Pensamento Computacional e Educação Matemática: Relações para o Ensino de Computação na Educação Básica**. Anais do XXXII CSBC. Curitiba: Sociedade Brasileira de Computação. 2012.

BARR, V.; STEPHENSON, C. Bringing computational thinking to K-12: what is involved and what is the role of the computer science education community? **Acm Inroads**, v. 2, n. 1, p. 48-54, Março 2011. Disponível em: <<http://dl.acm.org/citation.cfm?id=1929905>>. Acesso em: 12 jan 2017.

BARROS, R. P. **Evolução, Avaliação e Validação do Software RoboEduc**. Dissertação (Mestrado em Ciências). Universidade Federal do Rio Grande do Norte. Natal. 2011. 92 f.

BELL, T. et al. Computer science unplugged: School students doing real computing without computers. **The New Zealand Journal of Applied Computing and Information Technology**, v. 13, n. 1, p. 20-29, 2009. Disponível em: <<http://www.computingunplugged.org/sites/default/files/papers/Unplugged-JACIT2009submit.pdf>>. Acesso em: 12 jan 2017.

BERS, M. U. et al. **A Curriculum Unit on Programming and Robotics**. DevTech Research Group. Tufts University. Medford. 2010. 70 f.

BHATTACHARYA, J. **Rudiments of Computer Science**. 3. ed. Calcutá: Academic Publishers, v. 1, 2016. 520 p.

BRASSCON. Procuram-se profissionais de TI. **Associação Brasileira das Empresas de Tecnologia da Informação e Comunicação**, 2013. Disponível em: <<http://www.brasscom.org.br/brasscom/Portugues/detNoticia.php?codNoticia=400&codArea=2&codCategoria=26>>. Acesso em: 4 abr 2017.

BROOKS, K. M. **Do story agents use rocking chairs? The theory and implementation of one model for computational narrative**. Proceedings of the Fourth ACM International Conference on Multimedia. Boston: ACM. 1997. p. 317–328. Disponível em: <<http://dl.acm.org/citation.cfm?id=244233>>. Acesso em: 12 jan 2017.

BUDD, T. **An Introduction to Object-Oriented Programming**. 3. ed. Oregon: Addison-Wesley, 2001. 648 p.

BUECHLEY, L.; EISENBERG, M.; ELUMEZE, N. **Towards a curriculum for electronic textiles in the high school classroom**. Proceedings of the 12th annual SIGCSE conference on Innovation and technology in computer science education. Dundee: [s.n.]. 2007. p. 28-32. Disponível em: <<http://dl.acm.org/citation.cfm?id=1268795>>. Acesso em: 12 jan 2017.

BURNETT, M. M. Visual programming. In: _____ **Wiley Encyclopedia of Electrical and Electronics Engineering**. New York: John Wiley & Sons, Inc, 1999. Disponível em: <<http://onlinelibrary.wiley.com/doi/10.1002/047134608X.W1707/full>>.

BUTCHER, S. E. Narrative as a Teaching Strategy. **Journal of Correctional Education**, Ashland, v. 57, n. 3, p. 195-208, set 2006. Disponível em: <<http://warehouse.olc.edu/~khecrow/webfolder/Research/Butcher.pdf>>. Acesso em: 9 jan 2017.

CLEMENTS, D. H.; GULLO, D. F. Effects of computer programming on young children's cognition. **Journal of Educational Psychology**, v. 76, n. 6, p. 1051-1058, 1984. Disponível em: <<http://psycnet.apa.org/journals/edu/76/6/1051/>>. Acesso em: 9 jan 2017.

COHEN, L.; MANION, L.; MORRISON, K. **Research Methods in Education**. 6. ed. New York: Routledge, 2007. 638 p.

CONNELLY, F. M.; CLANDININ, D. J. Stories of experience and narrative inquiry. **Educational Researcher**, v. 19, n. 5, p. 2-14, jun-jul 1990. Disponível em: <<http://www.jstor.org/stable/1176100>>. Acesso em: 9 jan 2017.

COSTA JR., A. D. O.; GUEDES, E. B. **Uma Análise Comparativa de Kits para a Robótica Educacional**. Anais do XXXV Congresso da Sociedade Brasileira de Computação. Recife: [s.n.]. 2015. Disponível em: <<http://www.lbd.dcc.ufmg.br/colecoes/wei/2015/012.pdf>>. Acesso em: 13 jan 2017.

CRESWELL, J. W. **Projeto de pesquisa: métodos qualitativo, quantitativo e misto**. 2. ed. Porto Alegre: Artmed, 2007.

DA SILVEIRA, S. Inclusão digital, software livre e globalização contra-hegemônica. In: DA SILVEIRA, S.; CASSINO, J. **Software Livre e Inclusão Digital**. São Paulo: Conrad Editora do Brasil, 2003. p. 17-47.

DARGAINS, A.. **Estudo exploratório sobre o uso da robótica educacional no ensino de programação introdutória**. Dissertação (Mestrado em Informática). Universidade Federal do Rio de Janeiro. Rio de Janeiro. 2015. 155 f.

DE FRANÇA, R. S.; DO AMARAL, H. J. C. **Ensino de Computação na Educação Básica no Brasil: Um Mapeamento Sistemático**. Anais do XXXIII Congresso da Sociedade Brasileira de Computação. Maceió: [s.n.]. 2013. p. 426-431. Disponível em: <<http://www.lbd.dcc.ufmg.br/colecoes/wei/2013/009.pdf>>. Acesso em: 25 jan 2017.

DENIS, B.; HUBERT, S. Collaborative learning in an educational roboTIC environment. **Computers in Human Behavior**, v. 17, n. 5, p. 465-480, set-nov 2001. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0747563201000188>>. Acesso em: 12 jan 2017.

DEWEY, J. **Como Pensamos**. 3. ed. São Paulo: Cia Nacional, 1959.

DIJKSTRA, E. W. The humble programmer. **Communications of the ACM**, New York, v. 15, n. 10, p. 859-866, out 1972. Disponível em: <<http://dl.acm.org/citation.cfm?id=361591>>. Acesso em: 27 jan 2017.

DOHME, V. D. **Atividades Lúdicas na Educação**. Petrópolis: Vozes, 2003. 184 p.

DUMAS, J.; REDISH, J. **A practical guide to usability testing**. Portland: Intellect Books, 1999. 404 p.

EGUCHI, A. **What is educational roboTIC? Theories behind it and practical implementation**. Proceedings of Society for Information Technology & Teacher Education International Conference 2010. Chesapeake: [s.n.]. 2010. p. 4006-4014.

ESTADÃO. Crise? Não no mercado de Tecnologia da Informação. **Estadão**, 2016. Disponível em: <<http://educacao.estadao.com.br/noticias/geral,crise-nao-no-mercado-de-tecnologia-da-informacao,10000023666>>. Acesso em: 4 abr 2017.

FISHER, W. R. **Human communication as narration: toward a philosophy of reason, value, and action**. Columbia: University of South Carolina Press, 1987. 201 p.

FLANNERY, L. P. et al. **Designing ScratchJr: support for early childhood learning through computer programming**. Proceedings of the 12th International Conference on Interaction Design and Children. New York: ACM. 2013. p. 1-10. Disponível em: <<http://dl.acm.org/citation.cfm?id=2485785>>. Acesso em: 13 jan 2017.

FLAVELL, J. H.; MILLER, P. ; MILLER, S. A. **Cognitive development**. 4. ed. Upper Saddle River: Prentice-Hall, 1989. 408 p.

FRASER, N. **Ten things we've learned from Blockly**. Blocks and Beyond Workshop (Blocks and Beyond). [S.l.]: IEEE. 2015. p. 49–50. Disponível em: <http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=7369000>. Acesso em: 13 set 2016.

FURTADO, O.; BOCK, A. M. B.; TEIXEIRA, M. D. L. T. **Psicologias: uma introdução ao estudo de psicologia**. 13. ed. São Paulo: Saraiva, 2001. 365 p.

GOLDENBERG, M. **A arte de pesquisar: Como fazer pesquisa qualitativa em ciências sociais**. 8. ed. Rio de Janeiro: Record, 2004. 111 p.

GOMES, A.; HENRIQUES, J.; MENDES, A. J. Uma proposta para ajudar alunos com dificuldades na aprendizagem inicial de programação de computadores. **Educação, Formação & Tecnologias**, v. 1, n. 1, p. 93-103, 2008. Disponível em: <<http://eft.educom.pt/index.php/eft/article/view/23>>. Acesso em: 11 jan 2017.

GÓMEZ, G. R.; FLORES, J. G.; JIMÉNEZ, E. G. **Metodología de la investigación**. 2. ed. Málaga: Ediciones Aljibe, 1999. 378 p.

GROVER, S.; PEA, R. Computational Thinking in K-12: A Review of the State of the Field. **Educational Researcher**, Florida, v. 42, n. 1, p. 38–43, jan 2013. Disponível em: <<http://journals.sagepub.com/doi/abs/10.3102/0013189X12463051>>. Acesso em: 11 jan 2017.

HAZZAN, O. Reflections on teaching abstraction and other soft ideas. **ACM SIGCSE Bulletin**, v. 40, n. 2, p. 40-43, 2008.

HITCHCOCK, G.; HUGHES, D. **Research and the Teacher: A Qualitative Introduction to School-based Research**. 2. ed. London: Routledge, 1995. 370 p.

ISO. ISO 9241-11: Ergonomics requirements for office with visual display terminals (VDTs). **International Organization for Standardization**, 1998. Disponível em: <http://www.iso.org/iso/catalogue_detail.htm?csnumber=16883>. Acesso em: 31 jan 2017.

KAMII, C. **A criança e o número**: implicações educacionais da teoria de Piaget para a atuação junto a escolares de 4 a 6 anos. 36. ed. Campinas: PAPIRUS, 1992. 124 p.

KIRAKOWSKI, J.; CORBETT, M. SUMI: The software usability measurement inventory. **British journal of educational technology**, v. 24, n. 3, p. 210-212, 1993. Disponível em: <<http://onlinelibrary.wiley.com/doi/10.1111/j.1467-8535.1993.tb00076.x/abstract>>. Acesso em: 11 jan 2017.

KLOC, A. E.; KOSCIANSKI, A.; PILATTI, L. A. **Robótica**: uma ferramenta pedagógica no campo da. Anais do I Simpósio Nacional de Ensino de Ciência e Tecnologia - UTFPR. Ponta Grossa: [s.n.]. 2009. p. 1394-1403. Disponível em: <http://www.sinect.com.br/anais2009/artigos/11%20TICnoensinoaprendizagemdecienciaetecnologia/TICnoensinoaprendizagemdecienciaetecnologia_artigo15.pdf>. Acesso em: 12 jan 2017.

KOSTARAS, N. et al. Combining experimental and inquiry methods in software usability evaluation: : The paradigm of LvS educational software. **Journal of Systems and Information Technology**, v. 12, n. 2, p. 120–139, mai 2010. Disponível em: <<http://www.emeraldinsight.com/doi/abs/10.1108/13287261011042921>>. Acesso em: 11 jan 2017.

KRAMER, J. Is abstraction the key to computing? **Communications of the ACM**, New York, v. 50, n. 4, p. 36–42, abr 2007. Disponível em: <<http://dl.acm.org/citation.cfm?id=1232745>>. Acesso em: 12 jan 2017.

LISTER, R. **Concrete and other neo-Piagetian forms of reasoning in the novice programmer**. Proceedings of the Thirteenth Australasian Computing Education Conference-Volume 114. Perth: Australian Computer Society, Inc. 2011. p. 9-18. Disponível em: <<http://dl.acm.org/citation.cfm?id=2459938>>. Acesso em: 11 jan 2017.

LÓPEZ-RODRÍGUEZ, F. M.; CUESTA, F. Andruino-A1: Low-Cost Educational Mobile Robot Based on Android and Arduino. **Journal of Intelligent & Robotic Systems**, v. 81, n. 1, p. 63-76, maio 2015. Disponível em: <<http://link.springer.com/article/10.1007/s10846-015-0227-x>>. Acesso em: 11 jan 2017.

MALHOTRA, N. K. **Pesquisa de marketing**. 3. ed. Porto Alegre: Bookman, 2001.

MARCHAND, H. Em torno do pensamento pós-formal. **Análise Psicológica**, Lisboa, v. 20, n. 2, p. 191-202, abr 2002. Disponível em: <http://www.scielo.mec.pt/scielo.php?pid=S087082312002000200001&script=sci_arttext&lng=en>. Acesso em: 13 jan 2017.

MEC. **Base Nacional Comum Curricular**. Proposta Preliminar. Segunda Versão Revisada. 2016.

MEIRINHOS, M.; OSÓRIO, A. O estudo de caso como estratégia de investigação em educação. **EduSer - Revista de Educação**, Bragança, v. 2, n. 2, p. 49-65, 2010. Disponível em: <<http://hdl.handle.net/10198/3961>>. Acesso em: 18 jan 2017.

MERRIAM, S. B. **Case study research in education: A qualitative approach**. San Francisco: Jossey-Bass, 1988. 226 p.

MICHALAKIS, N. DRTECHNIKO. **How To Train Your Robot**, 2013. Disponível em: <<https://drtechniko.com/2012/04/09/how-to-train-your-robot/>>. Acesso em: 11 jan. 2017.

MOREIRA, M. A. **Teorias de aprendizagem**. São Paulo: EPU, 1999. 195 p.

NETO, A. F. M. **Software (livre) na arte computacional**. Dissertação (Mestrado em Artes). Universidade de Brasília. Brasília. 2010. 111 f.

NIELSEN, J.; LANDAUER, T. K. **A mathematical model of the finding of usability problems**. Proceedings of the INTERACT'93 and CHI'93 conference on Human factors in computing systems. Amsterdam: ACM. 1993. p. 206-213. Disponível em: <<http://dl.acm.org/citation.cfm?id=169166>>. Acesso em: 11 jan 2017.

NISBET, J.; WATT, J. Case study. In: BELL, J., et al. **Conducting Small-Scale Investigations in Educational Menagement**. London: Harper & Row, 1984. p. 79–92.

O'CONNELL, D. M. **Leitura: Interpretação e nível cognitivo-valorativo**. Dissertação (Mestrado em Linguística). Universidade Federal de Uberlândia. Uberlândia. 2007. 114 f.

OLIVEIRA, M. F. D. **Metodologia científica: um manual para a realização de pesquisas em administração**. Manual (pós-graduação). Universidade Federal de Goiás. Catalão. 2011. 72 f.

PAPERT, S. **Mindstorms: Children, Computers and Powerful Ideas**. 2. ed. New York: BasicBooks, 1993. 252 p.

PAPERT, S. **The Children's Machine: Rethinking School In The Age Of The Computer**. New York: BasicBooks, 1994. 256 p.

PAUL, A. M. Teaching computer science-without touching a computer. **The Education Digest**, v. 80, n. 5, p. 23-26, jan 2015. Disponível em: <<http://search.proquest.com/openview/3a95fce88e3339324502ebd5fd0978fd/1?pq-origsite=gscholar&cbl=25066>>. Acesso em: 11 jan 2017.

PEDROZO, M. K. As fases do Desenvolvimento Infantil parte 4: Estágio Operatório Concreto. **Espaço Mediação: Pedagogia e Psicologia**, 2014. Disponível em: <<http://psicopedagogiacuritiba.com.br/fases-desenvolvimento-infantil-parte-4-estagio-operatorio-concreto/>>. Acesso em: 11 jan 2017.

PELAES, M. L. W. Uma reflexão sobre o conceito de criatividade e o ensino da arte no ambiente escolar. **Revista educação-UNG**, v. 5, n. 1, p. 5-13, 2010. Disponível em: <<http://www.revistas.ung.br/index.php/educacao/article/viewArticle/537>>. Acesso em: 13 jan 2017.

PEREZ, A. L. F. et al. **Uso da Plataforma Arduino para o Ensino e o Aprendizado de Robótica**. International Conference on Interactive Computer aided Blended Learning (ICBL). Florianópolis: IAOE. 2013. p. 230-232. Disponível em: <http://www.icbl-conference.org/proceedings/2013/papers/Contribution77_a.pdf>. Acesso em: 11 jan 2017.

PIAGET, J. Intellectual evolution from adolescence to adulthood. **Human Development**, Basel, v. 15, n. 1, p. 1-12, 1972. Disponível em: <<http://www.karger.com/Article/Abstract/271225>>. Acesso em: 13 jan 2017.

PIAGET, J. **To understand is to invent**. New York: Grossman Publishers, 1974. 148 p.

PIAJET, J. **Investigaciones sobre la Abstracción Reflexionante**. Tradução de Alcía Entel. Buenos Aires: Crea AS, 1980.

PINTO, M. D. C. **Aplicação de Arquitetura Pedagógica em Curso de Robótica Educacional com Hardware Livre**. Dissertação (Mestrado em Informática). Universidade Federal do Rio de Janeiro. Rio de Janeiro. 2011. 158 f.

PRENSKY, M. Digital Natives, Digital Immigrants Part 1. **On the Horizon**, v. 9, n. 5, p. 1-6, set-out 2001. Disponível em: <<http://www.emeraldinsight.com/doi/pdfplus/10.1108/10748120110424816>>. Acesso em: 11 jan 2017.

RAZA, A.; CAPRET, L. ; AHMED, F. An open source usability maturity model (OS-UMM). **Computers in Human Behavior**, v. 28, n. 4, p. 1109-1121, jul 2012. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0747563212000209>>. Acesso em: 11 jan 2017.

READ, J. C.; MACFARLANE, S. **Using the fun toolkit and other survey methods to gather opinions in child computer interaction**. Proceedings of the 2006 conference on Interaction design and children. Tampere: ACM. 2006. p. 81-88. Disponível em: <<http://dl.acm.org/citation.cfm?id=1139096>>. Acesso em: 11 jan 2017.

RESNICK, M. Aprender a programar, programar para aprender. **Transformar 2014**, 2014. Disponível em: <<https://www.youtube.com/watch?v=hRGJUc2opQ4>>. Acesso em: 11 jan 2017.

RESNICK, M. et al. Scratch: programming for all. **Communications of the ACM**, New York, v. 52, n. 11, p. 60-67, nov 2009. Disponível em: <<http://dl.acm.org/citation.cfm?id=1592779>>. Acesso em: 11 jan 2017.

RIBEIRO, V. M. Alfabetismo funcional: referências conceituais e metodológicas para a pesquisa. **Educação & Sociedade**, v. 18, n. 60, p. 144-158, dez 1997. Disponível em: <<http://www.scielo.br/pdf/%0D/es/v18n60/v18n60a8.pdf>>.

Acesso em: 11 jan 2017.

RUBIN, J. **Handbook of Usability Testing: How to Plan, Design, and Conduct Effective Tests**. New York: JohnWiley & Sons, 1994. 352 p.

RUS, D. Teaching robot everywhere. **IEEE RoboTIC & Automation Magazine**, v. 13, n. 1, p. 15-94, mar 2006. Disponível em:

<https://www.eng.yale.edu/grablab/roboticscourseware/Rus_RAM.pdf>.

Acesso em: 11 jan 2017.

RUSK, N. et al. New Pathways into Robotics: Strategies for Broadening Participation. **Journal of Science Education and Technology**, v. 17, n. 1, p. 59–69, fev 2008.

Disponível em: <<http://link.springer.com/article/10.1007/s10956-007-9082-2>>.

Acesso em: 11 jan 2017.

SALVINI, P.; KORSAH, A.; NOURBAKHSH, I. Special Issue on Educational Robotics: Call for Papers. **IEEE Robotics & Automation Magazine**, 2015.

Disponível em: <<http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=7124590>>.

Acesso em: 12 jan 2017.

SBC. SBC realiza reunião para discutir a nova versão da Base Nacional Comum Curricular. **Sociedade Brasileira de Computação**, 2016a. Disponível em:

<<http://www.sbc.org.br/noticias/1693-sbc-realiza-reuniao-para-discutir-a-nova-versao-da-base-nacional-comum-curricular>>. Acesso em: 16 jan 2017.

SBC. SBC realiza reunião com o Ministro da Educação. **Sociedade Brasileira de Computação**, 2016b. Disponível em: <<http://www.sbc.org.br/noticias/1868-sbc-realiza-reuniao-com-o-ministro-da-educacao>>.

Acesso em: 16 jan 2017.

SCAICO, P. D. et al. Um Relato de Experiências de Estagiários da Licenciatura em Computação com o Ensino de Computação para Crianças. **RENOTE: Revista Novas Tecnologias na Educação-UFRGS**, v. 10, n. 3, 2012. Disponível em:

<<http://www.seer.ufrgs.br/renote/article/download/36377/23471>>.

Acesso em: 11 jan 2017.

SCHAFER, G. J. et al. **Designing the LIT KIT, an interactive, environmental, cyber-physical artifact enhancing children's picture-book reading**. Proceedings of the 12th International Conference on Interaction Design and Children. New York:

ACM. 2013. p. 281-284. Disponível em: <<http://dl.acm.org/citation.cfm?id=2485795>>.

Acesso em: 14 jan 2017.

SELLTIZ, C.; WRIGHTSMAN, L. S.; COOK, S. W. **Métodos de pesquisa das relações sociais**. São Paulo: Herder, 1965.

SETZER, V. W. **Computador a Papel, Uma atividade pedagógica para a**

introdução dos conceitos básicos dos computadores, 2005. Disponível em:

<<https://www.ime.usp.br/~vwsetzer/paper-comp.html>>. Acesso em: 11 jan 2017.

SHOTWELL, J.; WOLF, D.; GARDNER, H. Exploring early symbolization: Styles of achievement. In: SUTTON-SMITH, B. **Play and learning**. New York: Gardner Press., 1979. p. 127–156.

SMITH, J. E.; NAIR, R. **Virtual Machines**: Versatile Platforms for Systems and Processes. San Francisco: Elsevier, 2005. 656 p.

SOARES, R. F.; BORGES, M. A. F. **Robótica**: aprendizado em informática de forma lúdica. Anais do XXXI Congresso da Sociedade Brasileira de Computação. Natal: [s.n.]. 2011. p. 1516-1519. Disponível em:

<<http://www.dimap.ufrn.br/csbc2011/anais/eventos/contents/CADERNO.pdf>>.

Acesso em: 12 jan 2017.

SOLOMON, C. J.; PAPERT, S. **A case study of a young child doing Turtle Graphics in LOGO**. Proceedings of the June 7-10, 1976, national computer conference and exposition. New York: ACM. 1976. p. 1049-1056. Disponível em: <<http://dl.acm.org/citation.cfm?id=1499945>>. Acesso em: 11 jan 2017.

SOUZA, M. A. M.; DUARTE, J. R. R. Low-cost educational robotics applied to physics teaching in Brazil. **Physics Education**, v. 50, n. 4, p. 482-488, jun 2015. Disponível em: <<http://iopscience.iop.org/article/10.1088/0031-9120/50/4/482/meta>>. Acesso em: 11 jan 2017.

SOUZA, N. M. D.; WECHSLER, A. M. Reflexões sobre a teoria piagetiana: o estágio operatório concreto. **Cadernos de Educação: Ensino e Sociedade**, Bebedouro, v. 1, n. 1, p. 134-150, abr 2014. Disponível em:

<<https://profamaurybarreto.files.wordpress.com/2014/03/piaget.pdf>>.

Acesso em: 12 jan 2015.

SOUZA, P. R. D. A. et al. **LabVad**: Laboratório Remoto para o Desenvolvimento de Atividades Didáticas com Robótica. Memórias del XIX congresso Internacional Informática Educativa, TISE 2014. Fortaleza: [s.n.]. 2014. p. 690-694. Disponível em: <http://www.tise.cl/volumen10/TISE2014/tise2014_submission_218.pdf>.

Acesso em: 12 jan 2017.

SOUZA, V. C. D. A. **Os desafios da energia no contexto da termoquímica: modelando uma nova ideia para aquecer o ensino de química**. Dissertação (Mestrado em Educação). Faculdade de Educação da UFMG. Belo Horizonte. 2007. 205 f.

SUBRAMANIAM, M. et al. Sci-fi, storytelling, and new media literacy. **Knowledge Quest**, v. 41, n. 1, p. 22-27, set/out 2012. Disponível em:

<<http://search.proquest.com/openview/19545642176534c835b11c89b5b25926/1?pq-origsite=gscholar>>. Acesso em: 11 jan 2017.

SULLIVAN, A.; BERS, M. U. Robotics in the early childhood classroom: learning outcomes from an 8-week robotics curriculum in pre-kindergarten through second grade. **International Journal of Technology and Design Education**, v. 26, n. 1, p. 3–20, fev 2016. Disponível em: <<http://link.springer.com/article/10.1007/s10798-015-9304-5>>. Acesso em: 11 jan 2017.

SZURMAK, J.; THUNA, M. **Tell me a story**: The use of narrative as a tool for instruction. ACRL 2013 Proceedings. Indianápolis: [s.n.]. 2013. p. 546–552.

Disponível em:

<http://www.ala.org/acrl/sites/ala.org.acrl/files/content/conferences/confsandpreconfs/2013/papers/SzurmakThuna_TellMe.pdf>. Acesso em: 12 jan 2017.

TAUB, R.; ARMONI, M.; BEN-ARI, M. CS Unplugged and Middle-School Students' Views, Attitudes, and Intentions Regarding CS. **ACM Transactions on Computing Education (TOCE)**, New York, v. 12, n. 2, p. (art. 8), abr 2012. Disponível em:

<<http://dl.acm.org/citation.cfm?id=2160551>>. Acesso em: 12 jan 2017.

TAXA, F. D. O. S. **Problemas Multiplicativos e processo de abstração em crianças na terceira série do ensino fundamental**. Tese (Doutorado em Educação). UNICAMP. Campinas. 2001. 237 f.

TCHÁPEK, K. **A Fábrica de Robôs**. Tradução de Vera Machac. São Paulo: Hedra, 2012. 148 p.

TERRA, M. R. O desenvolvimento humano na teoria de Piaget. **UNICAMP**, 2010.

Disponível em: <<http://www.ceap.br/material/MAT31082010183131.doc>>.

Acesso em: 12 jan 2017.

VIEIRA, A.; PASSOS, O.; BARRETO, R. **Um relato de experiência do uso da técnica computação desplugada**. Anais do XXXIII Congresso da Sociedade Brasileira de Computação. Maceió: [s.n.]. 2013. p. 671–680. Disponível em:

<<http://www.lbd.dcc.ufmg.br/colecoes/wei/2013/0031.pdf>>. Acesso em: 12 jan 2017.

WHITEHOUSE. Office of the Press Secretary - Fact Sheet. **President Obama Announces Computer Science For All Initiative**, 2016. Disponível em:

<<https://www.whitehouse.gov/the-press-office/2016/01/30/fact-sheet-president-obama-announces-computer-science-all-initiative-0>>. Acesso em: 12 jan 2017.

WING, J. M. Computational thinking. **Communications of the ACM**, New Yoirk, v. 49, n. 3, p. 33-35, mar 2006. Disponível em:

<<http://dl.acm.org/citation.cfm?id=1118178.1118215&coll=portal&dl=ACM>>.

Acesso em: 11 jan 2017.

WING, M. Computational Thinking-What and Why? **The Link Magazine**, Pittsburgh, p. 20-23, Spring 2011. Disponível em: <http://www.cs.cmu.edu/sites/default/files/11-399_The_Link_Newsletter-3.pdf>. Acesso em: 12 jan 2017.

YACUZZI, E. El estudio de caso como metodología de investigación: Teoría, mecanismos causales, validación. **Documentos de Trabajo: Universidad del CEMA**, Buenos Aires, n. 296, jul 2005. Disponível em:

<<https://www.econstor.eu/handle/10419/84390>>. Acesso em: 19 jan 2017.

YIN, R. K. **Estudo de Caso: Planejamento e Métodos**. 2. ed. Porto Alegre: Bookman, 2001. 205 p.

ZIKMUND, W. G. **Business research methods**. 5. ed. Dryden: Fort Worth, 2000.

APÊNDICE A - PLANOS DE AULA

1 RELAÇÃO DE CONTEÚDOS POR ENCONTRO	186
2 LEGENDA	188
3 PLANOS	188
3.1 AULA 1 - O ROBÔ HUMANO	188
3.2 AULA 2 - A CAIXINHA MÁGICA	192
3.3 AULA 3 - O INTERIOR DA CAIXINHA MÁGICA, MOTOR DC E LED: MONTANDO MEU PRIMEIRO CIRCUITO ELETRÔNICO	195
3.4 AULA 4 - LED E OS BLOCOS REPETIR E ESPERAR: MONTANDO MEU SEGUNDO CIRCUITO ELETRÔNICO	200
3.5 AULA 5 - LED E OS BLOCOS REPETIR E ESPERAR (PARTE 2)	204
3.6 AULA 6 - LED E OS BLOCOS REPETIR E ESPERAR (PARTE 3)	207
3.7 AULA 7 - O LCD E O DISPLAY DE 7 SEGMENTOS / TESOURO SUBMARINO: CONHECENDO O PEIXE ROBÔ	210
3.8 AULA 8 - CONSTRUINDO PROGRAMAS A PARTIR DA OBSERVAÇÃO	214
3.9 AULA 9 - LCD, OS SENSORES DE LUZ E TEMPERATURA E O BLOCO ENQUANTO	217
3.10 AULA 10 - PROGRAMANDO O PEIXE ROBÔ	221
3.11 AULA 11 - A LEGIÃO DE VAMPIROS: O MORCEGO ROBÔ	223
3.12 AULA 12 - A LEGIÃO DE VAMPIROS: O CARRINHO ROBÔ VAMPIRO	226
3.13 AULA 13 - O CACHORRO ROBÔ E O BRAÇO ROBÓTICO: A MISSÃO ESPACIAL	228
3.14 AULA 14 - FINALIZANDO A OFICINA: PROJETANDO UM ROBÔ	232

1 RELAÇÃO DE CONTEÚDOS POR ENCONTRO

Encontro	Conteúdo Robótica	Conteúdo Programação
Aula 1	Noções Gerais de Robótica	Noções Gerais de Programação de Computadores
Aula 2	- Noções Gerais sobre Atuadores - Noções Gerais sobre Sensores	- O Ambiente DuinoBlocks4Kids (DB4K) - Comando Único (Motor DC e LEDs)
Aula 3	- Placa Arduino - Jumpers - Montagem de Circuitos Simples - Motor DC	- Estrutura de Programação Sequencial (EPS) - Atraso (delay) - DB4K
Aula 4	- LEDs - Montagem de Circuitos Simples - Placa Arduino - Jumpers	- Estrutura de Repetição Contada (ERCctda) [Repita <n> Vezes] - EPS - Atraso - DB4K

Aula 5	• LEDs	• EPS • ERCtda • Atraso • DB4K
Aula 6	• LEDs	• EPS • ERCtda • Atraso • DB4K
Aula 7	• LCD • Display de 7 Segmentos • LEDs	• EPS • ERCtda • Atraso • DB4K
Aula 8	• Display de 7 Segmentos • LEDs • Motor DC	• EPS • ERCtda • Atraso • DB4K
Aula 9	• LCD • LEDs • Motor DC • Sensor de Luz • Sensor de Temperatura	• Estrutura de Repetição Condicional (ERCond) [Enquanto <condição> Faça] • EPS • ERCtda • Atraso • DB4K
Aula 10	• Sensor de Luz • Buzzer • LED RGB • LEDs • Motor DC	• EPS • ERCtda • ERCond • Atraso • DB4K
Aula 11	• Sensor de distância • LED • LCD	• EPS • ERCond • Atraso • DB4K
Aula 12	• Carrinho Robótico • LED • Sensor de distância • Sensor de luz	• EPS • ERCond • Atraso • DB4K
Aula 13	• Servo Motor • Braço Robótico • Sensor de Luz • Sensor de Distância	• EPS • ERCond • Atraso • DB4K

Aula 14	• Display LCD • Display de 8 Segmentos • Sensor de Luz • Sensor de Calor • Sensor de Distância • Servo-Motor • Buzzer • LED • LED-RGB • Motor DC	• EPS • ERCtda • ERCond • Atraso • DB4K
----------------	---	---

2 LEGENDA

- **ORG:** Organização da turma: distribuição de materiais, etc.
- **DEB:** Debate: exposição oral dialogada com ou sem o apoio de recursos multimídia.
- **DEM:** Demonstração: exibição do funcionamento de algum dispositivo.
- **BRC:** Brincadeira: brincadeiras que visam introduzir ou exercitar de forma lúdica os conteúdos sendo trabalhados.
- **NAR:** Narrativas: uso de narrativas para a criação de um "pano de fundo" para as atividades práticas de programação e robótica.
- **PRT:** Atividades práticas que podem incluir programação, robótica e montagem de circuitos.
- **VER:** Atividade utilizada para verificar o entendimento dos alunos em relação aos conteúdos trabalhados.

3 PLANOS

3.1 AULA 1

Unidade didática: O Robô Humano

Duração: 90 min

Objetivo Geral: Compreender o que é um robô, o que é programação de computadores e como esses dois temas estão relacionados.

Objetivos Específicos

1. Entender o que é um robô.
2. Entender o que é programar.
3. "Escrever" programas para que o professor (robô humano) realize tarefas pré-determinadas.

Conteúdos

1. Noções gerais de robótica
2. Noções gerais de programação de computadores

Desenvolvimento Metodológico

1) **Boas vindas [20 min]** Tempo dedicado às apresentações iniciais.

2) **DEB/VER [20 min]** O professor promoverá um debate, com o auxílio de fotos e vídeos, buscando o entendimento do que vem a ser um robô.

Robôs são máquinas que podem seguir instruções para executar ações no mundo real e, em alguns casos, mudar suas ações ao perceber alguma alteração no ambiente.

Pode-se também, durante o debate, fazer os seguintes questionamentos (sempre acompanhados de exemplos):

- Robôs são máquinas? (SIM)
- Robôs têm que ter partes móveis? (SIM) (Será?)
- Robôs pensam sozinhos? (NÃO) (Não?)
- Robôs têm que se parecer com seres humanos? (NÃO)
- Robôs têm que se movimentar no ambiente? (NÃO)
- Alguns Robôs podem dizer como está o ambiente ao seu redor? (SIM)
- Pessoas dizem aos robôs como se comportar usando instruções chamadas de programas? (SIM)
- Etc.

(BERS *et al*, 2010)¹¹²

Para **verificar** o entendimento dos alunos, o professor deverá apresentar imagens e/ou vídeos de dispositivos que sejam robôs, e outros que não sejam, e perguntar explicitamente aos alunos: Isto é um Robô? Sim? Não? Por quê? (Ver material de apoio)

2) **DEM [20 min]** Serão apresentados os dispositivos com os quais os alunos irão trabalhar nas próximas aulas (Caixinha Mágica, Carro Robô e Braço Robótico). Durante a apresentação pode-se fazer um paralelo entre os dispositivos apresentados e a discussão realizada sobre o que vem a ser um robô. Pode-se mostrar a diferença entre uma “máquina” que segue instruções e outra que não, mostrando o carrinho robô andando sozinho (sem bater nas coisas, por exemplo) e um carrinho de brinquedo que não anda sozinho (BERS *et al*, 2010).

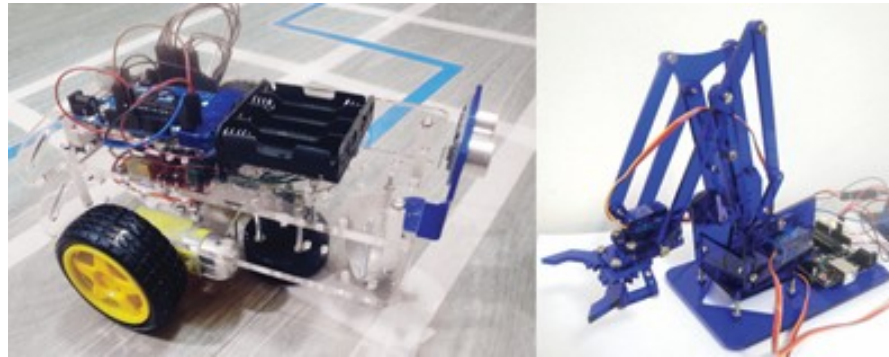
- O carrinho robô estará com o programa de desviar de obstáculos carregado. As crianças serão estimuladas a aproximarem as mãos do robô para vê-lo mudar de direção. É importante dizer às crianças que elas aprenderão a programar o carrinho robô para agir dessa forma.
- O braço robótico estará carregado com um programa básico de trocar os peões de lugar no tabuleiro (ver plano da Aula 13 e material de apoio¹¹³). Deve-se explicar

¹¹² BERS, M. U. et al. **A Curriculum Unit on Programming and Robotics**. DevTech Research Group. Tufts University. Medford. 2010. 70 f.

¹¹³ Vídeos tutoriais e arquivos com materiais para download serão em breve disponibilizados no site do LlvVRE(<http://www.nce.ufrj.br/ginape/livre/>)

às crianças que o robô foi programado para executar aquela tarefa, e que eles aprenderão a programá-lo para colocar os peões no local desejado.

- A caixinha mágica estará carregada com um programa que utilize todos os atuadores.
- Importante comentar com as crianças que elas não construirão robôs na oficina, mas sim, aprenderão a programar pequenos protótipos de robôs, bem como poderão também ajudar a construir alguns desses protótipos.



3) BRC/VER [30 min] Os alunos, divididos em 4 grupos, programarão o professor (robô humano), por meio de cartelas com comandos preestabelecidos, a realizar uma tarefa também predeterminada.

Para essa atividade, a sugestão é que se crie um tabuleiro quadriculado no chão, onde cada quadrado represente um passo do professor. Dentro do tabuleiro podem ser colocados objetos, e o professor deverá chegar até eles, pegá-los e levá-los até o destino.

É interessante que se coloque mais de um objeto e em posições que possibilitem diferentes possibilidades de caminhos para a realização da tarefa. Assim, é possível que os grupos façam programas diferentes e pode-se avaliar, por exemplo, junto com os alunos, que grupo criou o programa onde o robô deu menos passos (os alunos não devem ser avisados de que essa avaliação poderá ser feita).

Os comandos deverão estar presentes em cartões que se assemelhem aos blocos de programação do DB4K, preferencialmente incluindo ícones (ver material de apoio):



É importante que o professor comece com objetivos simples, como, por exemplo: “dar um passo, pegar objeto, dar um passo, soltar objeto”. Aos poucos a dificuldade vai sendo aumentada

Importante **verificar**, por meio de perguntas, se os alunos entendem que um programa de computador é, basicamente, uma sequência de comandos (ou ordens), para um computador executar (obedecer).

Se os alunos desejarem, pode-se colocá-los, um por vez, para ocuparem o lugar do robô humano, executando os comandos ditados pelos colegas.

Recursos Didáticos

- Sala com espaço livre (amplo)
- 1 computador com o DB4K instalado
- Projetor
- Caixinha mágica
- Carrinho robô
- Braço robótico
- Cartões com comandos para o robô humano
- Fita crepe (Para montar a pista [Tabuleiro quadriculado] onde o robô humano irá caminhar)
- Objetos para o robô humano pegar
- Carrinho de brinquedo com uma cordinha para puxar

Avaliação

1	Percebe que alguma coisa não está funcionando como esperado	5	4	3	2	1	0	NA
2	Conserva a meta original	5	4	3	2	1	0	NA
3	Tem uma hipótese para a causa do problema	5	4	3	2	1	0	NA
4	Tenta solucionar o problema	5	4	3	2	1	0	NA

1	Entende o que é um Robô	5	4	3	2	1	0
2	Entende o que é um Programa	5	4	3	2	1	0

Obs:

3.2 AULA 2

Unidade didática: A Caixinha Mágica

Duração: 90 min

Objetivo Geral: Conhecer superficialmente os dispositivos básicos utilizados em robótica educacional (atuadores, sensores, jumpers e placa Arduino) e ter um primeiro contato com a Ferramenta DuinoBlocks4Kids.

Programação: Comando único e o ambiente DB4K

Objetivos Específicos

1. Conhecer, sem detalhes, os seguintes atuadores: Display de 7 segmentos, LCD, LEDs, Motor DC, Servo Motor e Buzzer.
2. Conhecer, sem detalhes, os seguintes sensores: sensor de distância, sensor de luz e sensor de temperatura.
3. Aprender a conectar a caixinha mágica ao computador.
4. Conhecer o Duinoblocks4Kids.
5. Fazer o Motor DC Girar na caixinha mágica (programa de um único comando).

Conteúdos Novos

1. Atuadores
2. Sensores
3. Placa Arduino
4. DuinoBlocks4Kids
5. Comando Único (Fazer um programa de um único comando)

Desenvolvimento Metodológico

- 1) **ORG [5 min]** A turma será dividida em 4 grupos e cada grupo receberá uma "caixinha mágica" com todos os atuadores e sensores.
- 2) **DEM/DEB/VER [30 min]** O professor apresentará, no projetor, imagens de todos os atuadores e sensores presentes nas caixinhas (ver material de apoio). A cada dispositivo apresentado, pedirá para os alunos que mostrem onde ele se encontra na caixinha. Importante que o

professor saliente que todos esses dispositivos são utilizados por robôs e que, para se construir um robô, é preciso conhecê-los e aprender a programá-los.

Durante a apresentação o professor explicará:

O que são atuadores?

Atuadores são dispositivos (eletrônicos ou eletromecânicos) que permitem ao robô executar certas ações e se comunicar com o mundo. Por exemplo, os motores permitem aos robôs se movimentarem, como os músculos permitem que as pessoas se movimentem. Para se comunicar com o mundo as pessoas falam, escrevem; os robôs podem se comunicar com o mundo utilizando displays, buzinas, lâmpadas. Isso feito o professor pode solicitar aos alunos que deem exemplos de atuadores em seus corpos (braços, pernas, mãos, boca, etc.)

O que são sensores?

As pessoas utilizam seus sentidos para perceber o mundo e tomar decisões. Por exemplo, com a pele a pessoa percebe se está frio ou calor e assim pode decidir se vai sair de casa de camiseta ou de casaco. Com os olhos a pessoa pode perceber se está dia ou noite e assim pode decidir ficar brincando na rua enquanto está dia e voltar para casa quando for noite. Os robôs não têm olhos nem pele. Eles não são seres vivos, não são animais nem plantas. Os robôs utilizam os sensores para perceber o mundo e tomar decisões. Por exemplo, temos o sensor de distância que permite ao robô saber se ele vai bater em alguma coisa ou não (do mesmo modo que os morcegos fazem), temos o sensor de luz, que permite ao robô saber se está claro ou escuro. O sensor de calor, que permite ao robô saber se está quente ou frio, e vários outros sensores. O robô utiliza esses sensores para tomar decisões.

O Professor pode então pedir aos alunos que deem exemplos de sensores em seus corpos. Pode também fazer paralelos, como por exemplo: Os robôs utilizam um sensor de luz para verificar se está claro ou escuro, qual parte do corpo de vocês permite que vocês saibam se está claro ou escuro? E assim por diante. O professor pode também perguntar para as crianças se elas lembram de algum sensor que elas tenham visto em algum lugar (por exemplo: os sensores de movimento de portas automáticas).

Essas informações serão passadas em forma de debate, com os alunos em círculo e o professor lançando perguntas e colhendo ideias, além de poder utilizar imagens para ilustrar os temas sendo debatidos.

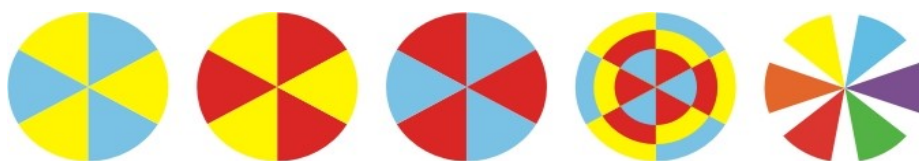
Após apresentar, sem profundidade, todos os atuadores e sensores, o professor abrirá sua própria caixinha para que os alunos vejam a conexão dos dispositivos, via jumpers, à placa Arduino. Como as caixinhas são transparentes, os alunos poderão observar nas suas próprias caixinhas que elas também têm uma placa Arduino. O professor explicará que essa placa é um “pequeno” computador, e que esse computador é o cérebro do Robô. É importante que o professor, sempre que for falar de algum atuador, sensor, placa, jumper, etc., pergunte aos alunos: Como é mesmo o nome desses fios? Como é mesmo o nome desse motor? Como é mesmo o nome desse display? Etc.

As crianças, claramente, não consolidarão o entendimento do que são sensores e atuadores nesta primeira aula. Isto é apenas uma introdução. Esse entendimento se dará no decorrer das aulas, onde cada dispositivo será trabalhado separadamente e depois em conjunto com os demais, de modo que as crianças se familiarizem com os nomes dos dispositivos e com suas funções.

3) **DEM/PRT/DEB [45min]** O professor ensinará aos alunos a conectarem suas caixinhas mágicas ao computador e solicitará que eles abram o ambiente DB4K. Isso feito, dirá que é por meio desse programa que eles irão aprender a programar os sensores e atuadores utilizados pelos robôs.

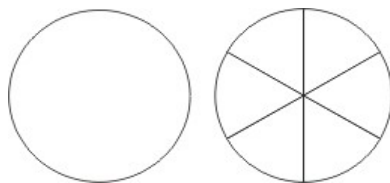
Os alunos abrirão o DB4K e o professor dirá que é por meio de blocos de encaixar que os sensores e atuadores serão programados nesse ambiente. Dirá também que existem outras maneiras de programar, por exemplo, escrevendo os comandos, e que essa, encaixando blocos, é uma delas. Isso feito, o professor seguirá a seguinte sequência de passos:

- Dirá aos alunos que eles irão fazer seu primeiro programa simples com o DB4K: fazer um motorzinho, chamado motor DC, girar em diferentes velocidades. Dirá também que no próximo encontro eles conhecerão melhor esse motor, mas, que já poderão fazê-lo funcionar agora.
- Pedirá para os alunos procurarem o bloco que faz o motor DC girar.
- Pedirá para os alunos colocarem esse bloco na área de trabalho (mostrando no projetor onde é a área de trabalho) e escolher a velocidade rápida.
- Perguntará para os alunos como eles acham que aquele comando colocado na área de trabalho pode ser enviado para a caixinha. Vinda a resposta ele pedirá que todos os grupos cliquem no botão *enviar programa* para que eles vejam o motor girando na caixinha.
- Isso feito, o professor solicitará que eles mudem a velocidade para devagar e cliquem no botão *enviar* para que os alunos vejam o efeito na caixinha mágica.
- O professor solicitará então que os alunos cliquem no botão para fazer o programa parar de ser executado na caixinha.
- Finalizada esta etapa, o professor entregará círculos coloridos impressos em papel grosso para que os alunos vejam o resultado da mistura das cores quando os círculos forem girados pelo motor DC (ver material de apoio).



- Os alunos colocarão, um a um, os círculos no motor e verão o que acontece quando o motor girar (não é preciso desligar o motor a cada troca, pois isso fará com que a atividade se estenda muito). O professor deve perguntar se eles entendem por que a cor está mudando e se eles ficaram surpresos com algum resultado (por exemplo, com o círculo com todas as cores que, ao girar, fica cinza clarinho, quase branco)
- O professor dirá então que os alunos farão um segundo programa simples: Mandar o LED acender.
- Solicitará que procurem o bloco para acender o LED.
- Uma vez o bloco tendo sido colocado na área de trabalho, solicitará que o comando seja enviado para a caixinha.
- O professor solicitará então que os alunos escolham outras cores de LED no bloco e enviem para a caixinha para ver o que acontece.

- 4) **BRC [10 min]** O Professor entregará lápis coloridos e dois círculos de papel cartão na cor branca para cada criança (ver material de apoio), e solicitará que elas pintem esses círculos com duas ou mais cores para depois observar que cores e efeitos surgem quando esses círculos girarem no motor DC.



Finalizada a pintura dos círculos, o professor solicitará que os alunos novamente mandem o motor DC girar e coloquem seus círculos coloridos no motor para observarem o efeito resultante.

Recursos Didáticos

- Datashow
- Computadores com o DB4K instalado
- 4 Caixinhas Mágicas
- Círculos Coloridos (ver material de apoio)
- Círculos Brancos (dois para cada aluno – ver material de apoio)
- Imagens dos sensores e atuadores para serem exibidas no projetor.

Avaliação

1	Percebe que alguma coisa não está funcionando como esperado	5	4	3	2	1	0	NA
2	Conserva a meta original	5	4	3	2	1	0	NA
3	Tem uma hipótese para a causa do problema	5	4	3	2	1	0	NA
4	Tenta solucionar o problema	5	4	3	2	1	0	NA

1	Consegue usar o DuinoBlocks4Kids para fazer o MOTOR DC girar	5	4	3	2	1	0
2	Consegue usar o DuinoBlocks4Kids para Acender os LEDs	5	4	3	2	1	0

Obs:

3.3 AULA 3

Unidade didática: O Interior da Caixinha Mágica, Motor DC e LED: Montando meu primeiro circuito eletrônico

Duração: 90 min

Objetivo Geral: Conhecer o funcionamento do Motor DC, sua montagem na placa Arduino e sua programação.

Programação: Estrutura de Programação Sequencial (EPS) e atraso (bloco esperar)
--

Objetivos Específicos

- | |
|--|
| <ol style="list-style-type: none"> 1. Conhecer a placa Arduino uno e compreender sua utilização. 2. Conhecer os jumpers e aprender como utilizá-los 3. Aprender a conectar o motor DC com jumpers à placa Arduino Uno
Desenvolver programas para controlar o motor DC. 4. Desenvolver programas para controlar o robô "gira-gira" (Ver material de Apoio). |
|--|

Conteúdos Novos

- | |
|---|
| <ol style="list-style-type: none"> 1. Placa Arduino 2. Jumpers 3. Montagem de circuitos simples 4. Motor DC 5. Estrutura de Programação Sequencial |
|---|

Desenvolvimento Metodológico

- | |
|--|
| <p>1) BRC [15 min] O professor explicará que todos irão participar de uma brincadeira para aprenderem como montar os programas que farão na aula.</p> |
|--|

Ele entregará então, para cada um, uma varinha mágica (podem ser hastes de plástico coloridas para balões) e dirá que os alunos são magos e que dependendo de como girarem as suas varinhas, todos juntos, poderão fazer com que o professor faça determinadas coisas. Em seguida dirá para os alunos qual o feitiço para fazer com que o professor cumpra uma determinada "prenda".

Antes de passar para os alunos os feitiços ele mostrará como girar lentamente e rapidamente a varinha. No momento do feitiço ele será também o regente das "giradas", contando com os alunos os giros.

Os "feitiços" serão coisas do tipo:

Ex1. Para fazer o professor imitar um cachorro, vocês precisam girar a varinha mágica devagar por 3 segundos (contando até três), parar, e depois girar a varinha mágica rápido por 2 segundos (contando até dois), e depois dizer: late cachorro, late!

Ex2. Para fazer o professor pular igual a um sapo vocês precisam girar a varinha rápido por dois segundos, parar, depois girar a varinha rápido por 4 segundos, parar, depois dar dois pulos e então dizer: Pula Sapo, pula!

O Professor, preferencialmente, avisará a sequência dos "comandos" conforme os alunos forem fazendo os feitiços (para eles não se perderem), ou seja, o professor dirá, por exemplo: - Vamos lá, vamos fazer o feitiço:

Professor: - Girar a varinha devagar por 3 segundos

Todos juntos - 1, 2, 3

Professor: Parar

Professor: - Girar a varinha rápido por 2 segundos

Todos juntos: - 1,2

Professor: Parar.

Professor: - Dizer: Late, Late!

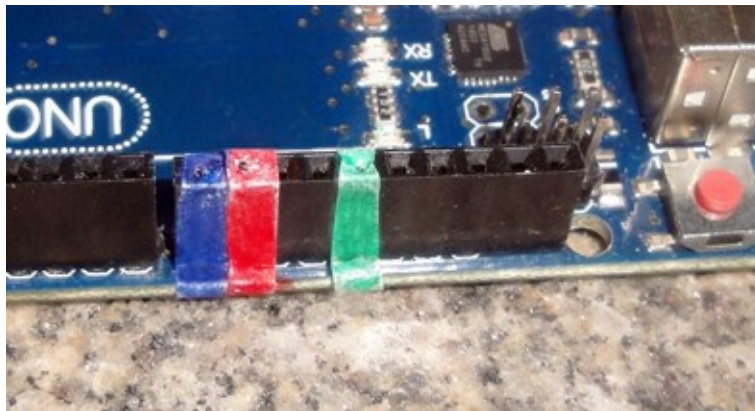
Todos juntos: - Late, Late!

(O professor então imita um cachorro latindo)

2) **ORG [5 min]** A turma será dividida em 4 grupos e cada grupo receberá uma placa Arduino Uno, Jumpers e um motor DC (do tipo usado em DVDs). O professor então relembra o que é o motor DC e que existem diferentes modelos.

3) **DEM [5 min]** O professor mostrará como conectar o motor à placa (Lembrar que os pinos a serem utilizados são pré-definidos no arquivo de configuração do Db4K).

- Os Pinos onde os alunos irão colocar o motor DC deverão estar marcados com fita crepe pintada da cor dos fios, facilitando a identificação pelos alunos.



- O Motor DC deverá estar preso a jumpers, previamente, para facilitar sua conexão com a placa.
- O Motor DC deverá ser do modelo de DVDs, com o fundo chato, para que os alunos possam deixá-lo na posição horizontal.
- Para essa explicação o professor pode, se julgar necessário, utilizar recursos de vídeo/fotos, para que os alunos vejam em tamanho maior, no projetor, como trabalhar com os componentes que eles acabaram de receber.
- O professor deve explicar também que é por meio dos jumpers (fios) que o cérebro do computador (a placa Arduino) envia os comandos para os atuadores, nesse caso, o motor.

4) **PRT/VER [10 min]** Os alunos, com o auxílio do professor, montarão o circuito com o motor DC. O professor deve prender, com fita crepe, o motor DC à mesa para que ele não saia do lugar no momento em que estiver girando. Para isso, a fita deve ser presa nos jumpers bem próximo ao motor DC.

5) **PRT/VER [20 min]** Os alunos montarão um programa (com o auxílio do professor) para controlar o motor DC.

O professor entregará cata-ventos para os alunos fixarem nos motores. Preferencialmente esses cata-ventos terão duas cores para que elas se misturem formando novas cores como na atividade com os círculos coloridos (ver material de apoio).



Isso feito, dirá então aos alunos que o motor DC agora é uma varinha mágica eletrônica e eles irão programá-la para realizar um feitiço que fará aparecer 2 bombons na mochila do professor. O feitiço é o seguinte:

Girar o motor devagar por 3 segundos
 Parar o motor por 1 segundo
 Girar o motor rápido 5 segundos
 Parar o motor por 2 segundos

Essa sequência deverá estar apresentada no projetor para os alunos poderem consultar sempre que quiserem e também entregue impressa, uma para cada grupo (ver material de apoio).

Durante a construção do programa o professor apresentará o bloco esperar para os alunos e explicará qual a sua função.

Conforme as duplas forem finalizando seus programas, devem chamar o professor para ver se o feitiço está programado corretamente. Se estiver, o professor irá até sua mochila e pegará dois bombons para entregar aos alunos, dizendo algo do tipo: - Será que os bombons apareceram? Vamos ver?

O professor deverá observar se os alunos percebem que a sequência que eles programaram está sendo executada “sem parar”. Aproveitará então esse momento para explicar que o Arduino repete indefinidamente a sequência de comandos. Por que isso? Para que o robô possa ficar sozinho, fazendo o que tem que fazer, indefinidamente.

O Professor pode preparar outras ideias de programa para os alunos fazerem no caso de sobrar tempo. Possivelmente outros feitiços que farão outras coisas, ou mais bombons, aparecerem.

6) **PRT/VER [30 min]** - O Professor apresentará o Robô Gira-Gira (Ver material de apoio) para os alunos e entregará as caixinhas mágicas somente com o motor DC e os LEDs.

(A partir desta aula os dispositivos serão acoplados à caixinha à medida que forem ser utilizados. Ao final da oficina a caixinha estará novamente completa e as crianças terão trabalhado com todos os sensores e atuadores)

O professor solicitará que as crianças façam um programa para controlar o robô gira-gira, mandando ele girar e acender um LED. O programa, ao final, deverá fazer com que o robô termine parado com o LED apagado, para isso deve-se usar o bloco [parar programa].

Se o professor olhar o programa e observar que não está correto, deve pedir para os alunos carregarem o mesmo na caixinha e observar se está fazendo o que eles querem.

Para que a dinâmica de carga e execução do programa no Robô gira-gira funcione de forma adequada os seguintes cuidados devem ser tomados:

- Sempre que uma equipe carregar o programa no robô, este deverá estar com a bateria desconectada. Então o robô deverá ser levado ao centro da sala e a bateria será conectada e assim que a roda estiver girando ele será colocado no chão. O primeiro comando será de fazer o robô girar rápido por 10 segundos com o LED vermelho aceso.
- Uma vez terminada a execução do programa, a bateria continuará conectada e o professor levará o robô para ter o programa carregado pela próxima equipe. Uma vez o programa carregado a bateria é desconectada e repete-se o procedimento.

O programa a ser construído pelas crianças é:

Girar rápido por 10 segundos com o LED vermelho aceso
Parar o motor
Apagar o LED
Parar o programa

Esse desafio deverá ser entregue impresso para os alunos ou exibido no projetor

Caso sobre tempo o professor pode refazer a brincadeira da mágica com o cata-vento na placa para dar mais bombons aos alunos.

Obs. Todos os programas solicitados deverão ser expostos no projetor para auxiliar as crianças.

Recursos Didáticos

- Projetor
- Vídeos e/ ou fotos com a demonstração da conexão dos componentes à placa Arduino
- Computadores com o DB4K instalado
- 4 Caixinhas Mágicas
- 4 “mini” Kits Arduino (ver material de apoio)
- Cata-ventos
- Robô Gira-Gira
- 16 bombons para a atividade da varinha mágica eletrônica (2 por aluno)
- Hastes plásticas coloridas para balões
- Os algoritmos das magias para serem apresentados no projetor.

- Possíveis materiais para outras "magias" da atividade da varinha mágica eletrônica.
- Folhas com os algoritmos/desafios de programação impressos (ver material de apoio).

Avaliação

1	Percebe que alguma coisa não está funcionando como esperado	5	4	3	2	1	0	NA
2	Conserva a meta original	5	4	3	2	1	0	NA
3	Tem uma hipótese para a causa do problema	5	4	3	2	1	0	NA
4	Tenta solucionar o problema	5	4	3	2	1	0	NA

1	Monta o circuito proposto	5	4	3	2	1	0
2	Constrói corretamente o programa para fazer o MOTOR DC girar em diferentes velocidades em conjunto com um LED	5	4	3	2	1	0
3	Entende o uso do bloco ESPERAR	5	4	3	2	1	0
4	Utiliza corretamente o bloco ESPERAR	5	4	3	2	1	0
5	Utiliza corretamente o bloco PARAR programa	5	4	3	2	1	0

Obs:

3.4 AULA 4

Unidade didática: LED e os blocos Repetir e Esperar – Montando meu segundo circuito eletrônico

Duração: 90 min

Objetivo Geral: Conhecer melhor os LEDs e aprender a programá-los usando a Estrutura de Repetição Contada.

Programação: EPS e Estrutura de Repetição Contada (ERCtda) [Repita <n> vezes] e atraso (bloco esperar)

Objetivos Específicos

1. Conhecer os LEDs em mais detalhes e aprender para que servem.
2. Aprender a conectar um LED à placa Arduino Uno.
3. Desenvolver programas para controlar o LED na placa.
4. Desenvolver programas para controlar os 3 LEDs + o Motor DC da caixinha mágica.
5. Aprender a utilizar a estruturas de repetição contada repita n vezes.

Conteúdos Novos

1. LEDs
2. Estrutura de Repetição Contada (Repita <n> vezes)

Desenvolvimento Metodológico

- 1) **DEB [10 min]** Com a turma em círculo o professor fará um debate sobre todos os conteúdos vistos até o momento (o que é um programa, o que são robôs, o que é um motor DC e para o que ele serve, etc.).
- 2) **DEM/DEB [10 min]** O professor explicará o que é um LED. É importante que o professor peça exemplos de lugares onde os LEDs podem ser utilizados e lugares que eles lembrem de ter visto LEDs funcionando, para que reforcem bem a ideia de que o LED é um tipo de “lâmpada” muito utilizada para mostrar se algo está ligado ou desligado (televisores, por exemplo) e para iluminação simples, como no caso das lâmpadas feitas de LED (seria interessante levar uma lâmpada de LEDs para os alunos verem). Interessante também mostrar vídeos de cubos de LED (ver material de apoio).
- 3) **BRC/DEB [15 min]** O professor fará a brincadeira do “**Robô Humano**” com os alunos, que deverão seguir uma sequência de “comandos” para levantarem e baixarem cartões com duas cores (Azul e Verde, por exemplo, ver material de apoio).



(Cartões com aproximadamente 20 cm de largura)

Esses comandos serão apresentados no projetor em um programa em blocos semelhante aos do DB4K. O professor lerá os comandos do projetor e os alunos deverão executá-los conforme o professor for anunciando.

Caso o professor deseje criar um programa diferente do aqui sugerido, e não tenha a possibilidade de desenhar os blocos que deseja, ele pode simplificar, utilizando retângulos coloridos com os textos escritos e figurinhas que representem as ações dos blocos, mas, sem serem necessariamente tão semelhantes aos do DB4K.

Uma vez finalizada a brincadeira, o professor deve levantar um debate sobre a mesma, com perguntas como: porque alguns comandos eram feitos mais de uma vez? Onde estava sendo dito que os comandos deveriam ser repetidos? Que comandos eram repetidos? etc.



Programa para ser utilizado com as maçãs e estrelas

- 4) **ORG [5 min]** A turma será dividida em 4 grupos e cada grupo receberá uma placa Arduino Uno, Jumpers, um Resistor, um LED e a Caixa Mágica (as caixinhas estarão equipadas apenas com 4 LEDs e o Motor DC).
- 5) **PRT/VER [15 min]** O professor mostrará como ligar o LED à placa Arduino e os alunos, com o auxílio do professor, montarão o circuito com 2 LEDs (da mesma cor dos cartões a serem utilizados na próxima atividade). Assim como na atividade com o Motor DC, os pinos a serem utilizados na placa Arduino deverão estar marcados com a mesma cor dos jumpers que deverão ser conectados a eles.

Durante a explicação da montagem o professor apresentará brevemente os resistores, explicando apenas que são dispositivos utilizados para os LEDs não queimarem. Ele pode, se achar apropriado, explicar que os LEDs “diminuem” a quantidade de “energia” que chega aos LEDs pois se eles recebessem muita energia eles poderiam estragar

(Lembrar que os pinos a serem utilizados são pré-definidos no arquivo de configuração do DB4K.)

6) **PRT/VER [30 min]** Os alunos montarão um programa para piscar os LEDs no circuito montado. Este programa deverá executar a mesma sequência que os alunos executaram com os cartões (piscar 2 vezes o LED amarelo e depois 3 vezes o LED Verde).

7) **PRT/VER [extra]** Para as crianças que terminarem o desafio anterior, o professor pode aplicar o desafio da palavra código. Nesta atividade, os alunos devem criar um programa para escrever a palavra ROBÔ através de um código no qual cada letra da palavra correspondia a um número de piscadas do LED de uma determinada cor:

R					
O					
B					
Ô					

(Uma outra variante dessa atividade, onde as crianças recebem cartões e um gabarito para elas mesmas montarem suas palavras e depois as outras crianças tentarem descobrir o que a caixinha está dizendo por meio dos LEDs piscando, encontra-se no plano da aula 8)

Recursos Didáticos

- Projetor
- Computadores com o DB4K instalado
- 4 Caixinhas Mágicas
- 4 Kits Arduino (ver material de apoio)
- Cartões coloridos (como a maçã e a estrela)
- Imagem com o programa para o robô humano.
- Os algoritmos/desafios para serem apresentados no projetor (e/ou impressos e entregues aos alunos)
- Vídeos e Fotos com exemplos de uso de LEDs
- Imagem da palavra código para ser apresentada no projetor

Avaliação

1	Percebe que alguma coisa não está funcionando como esperado	5	4	3	2	1	0	NA
2	Conserva a meta original	5	4	3	2	1	0	NA
3	Tem uma hipótese para a causa do problema	5	4	3	2	1	0	NA
4	Tenta solucionar o problema	5	4	3	2	1	0	NA

1	Entende para que servem os LEDs	5	4	3	2	1	0
2	Monta o circuito proposto	5	4	3	2	1	0
3	Desenvolve corretamente os programas para piscar os LEDs	5	4	3	2	1	0
4	Percebe quanto utilizar o bloco REPITA <N> VEZES em seus programas	5	4	3	2	1	0
5	Utiliza corretamente o bloco REPITA	5	4	3	2	1	0
6	Utiliza corretamente o bloco ESPERAR	5	4	3	2	1	0
7	Faz reaproveitamento de código	5	4	3	2	1	0

Obs:

3.5 AULA 5

Unidade didática: LED e os blocos Repetir e Esperar (Parte 2)

Duração: 90 min

Objetivo Geral: Trabalhar melhor o conjunto de comandos para fazer os LEDs piscarem e aprender a programá-los usando a estrutura de repetição contada.

Programação: EPS e Estrutura de Repetição Contada (ERCtda) [Repita <n> vezes] e atraso (bloco esperar)

Objetivos Específicos

1. Trabalhar, por meio de uma demonstração com uma lanterna, o entendimento do funcionamento do bloco esperar para fazer o LED piscar. Trabalhar o entendimento do funcionamento do bloco repetir por meio da brincadeira do robô humano.
2. Desenvolver programas com para o controle dos LEDs fazendo com que LEDs de diferentes cores pisquem, cada um, um determinado número de vezes.

Conteúdos Novos

Desenvolvimento Metodológico

- 1) **DEM/DEB [15 min]** O professor, fazendo uso de uma lanterna, mostrará para as crianças as ações necessárias para fazer ela piscar, quais sejam: acendê-la, mantê-la acesa por algum tempo, apagá-la, mantê-la apagada por algum tempo, acendê-la novamente, mantê-la acesa por algum tempo, apagá-la novamente, mantê-la apagada por algum tempo, e assim por diante. A ideia é as crianças perceberem a necessidade de manter-se a luz acesa por algum tempo e apagada por algum tempo também. O professor deve reforçar a ideia de que se conseguíssemos acender e apagar a lanterna numa velocidade muito, muito, rápida não conseguiríamos ver ela piscar. O computador, por outro lado, executa essas ações

numa velocidade muito rápida e nós então precisamos dizer para ele que é para ele esperar um pouquinho entre uma ação e outra.

- 2) **DEB [5 min]** O professor revisará um por um os blocos necessários para fazer os LEDs piscarem. Depois, montará com os alunos a sequência necessária para fazê-los piscar, buscando reforçar o entendimento da função do bloco esperar.
- 3) **ORG [5 min]** A turma será dividida em 4 grupos e cada grupo receberá uma caixinha mágica contendo apenas os LEDs e o motor DC.
- 4) **PRT/VER [30 min]** O professor solicitará então que as crianças construam, em sequência, os seguintes programas (sem o uso do bloco repetir):
 1. O **LED azul** piscar por (1 segundo aceso e 1 segundo apagado)
 2. O **LED azul** mais de vagar (2 segundos aceso e 2 segundos apagado)
 3. O **LED azul** piscar, depois o **LED amarelo** piscar
 4. O **LED azul** piscar **duas vezes** e depois o **LED amarelo** piscar **uma vez**
 5. O **LED azul** piscar **duas vezes** e depois o **LED amarelo** piscar **uma vez** e depois o **LED Vermelho** piscar **1 vez**
 6. Cada um desses LEDs piscar em **velocidades diferentes**.
 7. O **LED amarelo** piscar **duas vezes**, depois o **LED azul** piscar **uma vez** e depois o **LED vermelho** piscar **2 vezes**
- 5) **BRC/DEB [10 min]** O professor apresentará, com o projetor, um programa para o robô humano, com blocos iguais aos da primeira aula. Nesse programa o robô terá que dar um determinado número de passos para frente até pegar um objeto. As crianças lerão os comandos em voz alta e o professor realizará as ações (ver plano da Aula 1). Depois, o professor apresentará, ao lado do primeiro programa, um segundo programa que realiza a mesma tarefa do anterior, mas, utilizando o bloco repetir.



Durante a atividade o professor deve fazer as crianças refletirem sobre o fato de que o bloco repetir serve para que não seja necessário escrever várias vezes os mesmos comandos, e também que, apenas os comandos posicionados dentro do bloco repetir são repetidos.

Chamar também a atenção para o fato de que o bloco repetir é um bloco diferente, um bloco que “abraça” outros blocos, sendo que os blocos que estão dentro desse “abraço” é que são repetidos.

6) **PRT/VER [25 min]** O professor apresentará a seguinte sequência de exercícios, nos quais alguns programas utilizam o bloco repetir e outros não.

- Mandar o **LED vermelho** piscar 2 vezes (acender por 1 segundo apagar por 1 segundo) usando o bloco repetir.
- Mandar o **LED amarelo** piscar 3 vezes, sem o bloco repetir.
- Mandar o **LED amarelo** piscar 3 vezes, usando o bloco repetir.
- Mandar o **LED vermelho** piscar 2 vezes, depois o **LED amarelo** piscar 4 vezes, utilizando o bloco repetir.
- Mandar o **LED vermelho** piscar 3 vezes, depois o **LED azul** piscar 2 vezes, depois **LED amarelo** piscar 2 vezes, utilizando o bloco repetir.

O professor deve observar se os alunos reaproveitam código. Por exemplo, se os alunos fizeram um programa para o LED azul piscar duas vezes sem o bloco repetir, se o próximo programa for realizar a mesma ação, mas, com o uso do bloco repetir, verificar se os alunos apagam uma das sequências e reaproveitam a outra colocando-a dentro da estrutura de repetição contada. O professor pode, no termino do desafio, mostrar no projetor esse tipo de reaproveitamento de código. Desse modo o professor reforça a ideia para quem fez o reaproveitamento e a apresenta para quem não fez.

Recursos Didáticos

- Projetor
- 5 Computadores com o DB4K instalado
- 4 caixinhas mágicas
- 1 lanterna
- Imagem do programa em blocos para o robô humano
- Os algoritmos/desafios para serem apresentados no projetor (e/ou impressos e entregues aos alunos)

Avaliação

1	Percebe que alguma coisa não está funcionando como esperado	5	4	3	2	1	0	NA
2	Conserva a meta original	5	4	3	2	1	0	NA
3	Tem uma hipótese para a causa do problema	5	4	3	2	1	0	NA
4	Tenta solucionar o problema	5	4	3	2	1	0	NA

1	Entende para que servem os LEDs	5	4	3	2	1	0
3	Desenvolver corretamente programas para piscar os LEDs	5	4	3	2	1	0
5	Perceber quanto utilizar o bloco REPITA <N> VEZES em seus programas.	5	4	3	2	1	0
6	Utiliza corretamente o bloco REPITA	5	4	3	2	1	0
7	Utiliza corretamente o bloco ESPERAR	5	4	3	2	1	0
8	Faz reaproveitamento de código	5	4	3	2	1	0

Obs:

3.6 AULA 6

Unidade didática: LED e os blocos Repetir e Esperar (Parte 3)

Duração: 90 min

Objetivo Geral: Trabalhar melhor o conjunto de comandos para fazer os LEDs piscarem e aprender a programá-los usando a estrutura de repetição contada, bem como fazer as crianças refletirem sobre os efeitos da mudança na ordem dos comandos de um programa (fluxo de controle).

Programação: EPS e Estrutura de Repetição Contada (ERCtda) [Repita <n> vezes] e atraso (bloco esperar)

Objetivos Específicos

1. Realizar uma sequência de exercícios simples de programação que ajudem as crianças a perceberem o efeito do uso do bloco esperar em conjunto com os blocos acender e apagar LED.
2. Reforçar o entendimento do bloco repetir e sua aplicabilidade por meio da brincadeira do robô humano, comparando-se a execução de programas com e sem o uso do bloco repetir e buscando o entendimento de que com o mesmo conjunto de comandos, apenas mudando-se a ordem, pode-se fazer com que o robô realize diferentes tarefas.
3. Desenvolver programas para o controle dos LEDs, fazendo com que LEDs de diferentes cores pisquem, cada um, um determinado número de vezes.

Conteúdos Novos

Desenvolvimento Metodológico

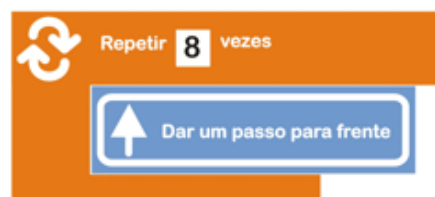
- 1) **ORG [5 min]** A turma será dividida em 4 grupos e cada grupo receberá uma Caixa Mágica (as caixinhas ainda possuirão apenas os LEDs e o Motor DC).

2) **PRT/DEB [30 min]** O professor fará mais alguns exercícios com os blocos LED e esperar, propondo variações no tempo em que o LED permanece aceso e apagado. O professor fará o exercício junto com os alunos e pedirá para eles enviarem, um a um, os programas para a caixinha mágica e observarem o que acontece. O objetivo é que as crianças percebam que sem o uso do bloco esperar não se consegue observar os LEDs piscarem e que, modificações nos tempos de espera farão com que o LED fique aceso ou apagado por mais ou menos tempo, o que resultará em “piscadas” mais rápidas ou mais lentas:

- Acender o **LED vermelho** e Apagar o **LED vermelho**
- Acender o **LED vermelho** por 1 segundo e apagar o **LED vermelho** por 1 segundo
- Acender o **LED vermelho** por 1 segundo e apagar o **LED vermelho** por 3 segundos
- Acender o **LED vermelho** por 3 segundos e apagar o **LED vermelho** por 1 segundo
- Piscar um **LED de azul** uma vez e o **LED amarelo** 1 vez, sendo que o **LED azul** deve piscar mais rápido que o **LED amarelo**.

O professor deve lançar questões durante a realização desse exercício, perguntando, por exemplo, por que no terceiro exercício o LED ficou mais tempo aceso do que apagado e no quarto exercício ele ficou mais tempo apagado do que aceso.

3) **BRC [20 min]** O professor fará novamente a brincadeira do robô humano para o estudo do funcionamento do bloco repetir. No entanto, desta vez, os alunos também participarão como robôs e os exercícios buscarão tornar mais explícita a “vantagem” no uso do bloco repetir, como por exemplo, mandar o robô dar 8 passos sem e com uso do bloco repetir.



Além disso, serão executados programas com modificação apenas dos parâmetros do bloco repetir e/ou na posição de determinados comandos, de modo que as crianças percebam que com os mesmos blocos é possível realizar diferentes sequências de ações (ver material de apoio). Importante lembrar que as crianças devem ler um a um os comandos para o robô executar. No caso das repetições, sugerir que as crianças leiam por exemplo: dar 3 passos para frente: 1, 2, 3 (um número para cada passo que o robô deve dar).

Na sequência, as crianças serão o robô em uma série de exercícios como o que segue (ver material de apoio):



4) **PRT/VER [35 min]** O professor lançará uma série de desafios para os alunos programarem os LEDs para piscarem na caixinha mágica sem e com o uso do bloco repetir, observando se os alunos fazem reaproveitamento de código e chamando a atenção para essa possibilidade ao final da tarefa, mostrando com o projetor como fazer o reaproveitamento de código em um exemplo no qual primeiro se faz os LEDs piscarem sem o uso do bloco repetir e posteriormente com o uso desse bloco. Pode-se usar para esta tarefa a seguinte sequência de desafios:

- Mandar o **LED azul** piscar 2 vezes, depois o **LED amarelo** piscar 1 vez, sem o uso do bloco repetir.
- Mandar o **LED azul** piscar 2 vezes, depois o **LED amarelo** piscar 1 vez, com o uso do bloco repetir.
- Mandar o **LED azul** piscar 2 vezes, depois o **LED amarelo** piscar 2 vezes, sem o uso do bloco repetir.
- Mandar o **LED azul** piscar 2 vezes, depois o **LED amarelo** piscar 2 vezes, depois o **LED vermelho** piscar 1 vez, com o uso do bloco repetir.
- Mandar o **LED azul** piscar 5 vezes, depois o **LED amarelo** piscar 3 vezes, depois o **LED vermelho** piscar 1 vez, com o uso do bloco repetir.
- No desafio anterior, fazer o **LED azul** piscar mais rápido e o **LED amarelo** mais devagar.

Recursos Didáticos								
• Projetor • Computadores com o DB4K instalado • 4 Caixinhas Mágicas • Imagens com os programas para os robôs humanos • Os algoritmos/desafios para serem apresentados no projetor (e/ou impressos e entregues aos alunos)								
Avaliação								
1	Percebe que alguma coisa não está funcionando como esperado	5	4	3	2	1	0	NA
2	Conserva a meta original	5	4	3	2	1	0	NA
3	Tem uma hipótese para a causa do problema	5	4	3	2	1	0	NA
4	Tenta solucionar o problema	5	4	3	2	1	0	NA
1	Entende para que servem os LEDs	5	4	3	2	1	0	
3	Desenvolve corretamente os programas para piscar os LEDs	5	4	3	2	1	0	
5	Percebe quanto utilizar o bloco REPITA <N> VEZES em seus programas.	5	4	3	2	1	0	
6	Utiliza corretamente o bloco REPITA	5	4	3	2	1	0	
7	Utiliza corretamente o bloco ESPERAR	5	4	3	2	1	0	
8	Faz reaproveitamento de código	5	4	3	2	1	0	
Obs:								

3.7 AULA 7

Unidade didática: O LCD e o display de 7 segmentos / Tesouro submarino: conhecendo o Peixe Robô

Duração: 90 min

Objetivo Geral: Conhecer o display de 7 segmentos e o LCD. Revisar o uso dos blocos acender LED, apagar LED, esperar e repetir.

Programação: EPS – ERCtda e atraso.

Objetivos Específicos

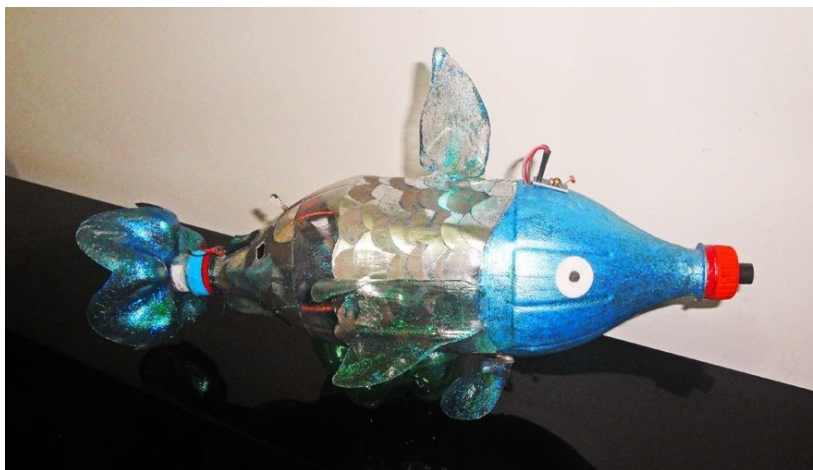
1. Apresentar o peixe robô à turma.
2. Conhecer com mais detalhes o display de 7 segmentos e seu funcionamento.
3. Conhecer com mais detalhes o LCD e seu funcionamento.

Conteúdos Novos

1. Display de 7 segmentos
2. LCD

Desenvolvimento Metodológico

- 1) **NAR/DEM [20 min]** O Professor Apresentará o peixe-robô para os alunos, como motivação para aulas futuras, nas quais o peixe robô será trabalhado.



O professor mostrará o protótipo de peixe robô para os alunos e contará a história abaixo, sendo que, enquanto conta a história, o professor deve apresentar ilustrações do ambiente e das personagens, como a caverna submarina e o peixe monstro mutante (ver material de apoio):

Uma equipe de exploradores descobriu um navio afundado em uma caverna submarina muito profunda e acredita que haja um tesouro escondido nos destroços desse navio. Nessa parte do oceano profundo, onde está localizada essa caverna, existem peixes monstros enormes, terríveis, sempre muito famintos e que conseguem produzir luz própria e iluminar o ambiente para encontrar suas presas. E o que é pior, eles adoram devorar exploradores.

Para proteger os mergulhadores enquanto eles vasculham a caverna à procura do tesouro, uma equipe de engenheiros teve a ideia de desenvolver um peixe robô vigia, sensível à luz, que ficará nadando em círculos na entrada da caverna.

O peixe robô usará um motor DC para se locomover em círculos e, sempre que um monstro submarino se aproximar da entrada da caverna, sua luz iluminará o robô e, enquanto esta luz o estiver iluminando, ele deverá alertar os mergulhadores sobre a presença do monstro soando um alarme e piscando luzes.

Vocês fazem parte da equipe de programadores contratada para desenvolver o programa que fará o peixe robô funcionar conforme o planejado pela equipe de engenheiros.

Nesse momento o professor pede para os alunos se aproximarem e começa a mostrar as partes do peixe robô: a placa Arduino que fará o controle, rodando o programa que eles irão desenvolver, a bateria que fornecerá a energia para o peixe funcionar, o motor DC responsável por fazer o rabo girar, o sensor de luz responsável por detectar a presença dos peixes monstros, o buzzer responsável por fazer o som do alarme e alguns LEDs.

Feita toda a apresentação o professor então apagará a luz da sala e fará uma demonstração do funcionamento do peixe robô utilizando uma lanterna para fazer o alarme soar.

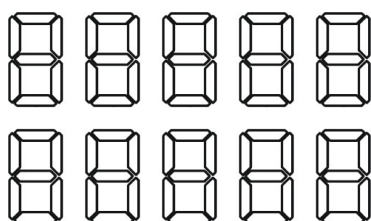
Isso feito, o professor irá propor aos alunos uma brincadeira na qual um aluno será o peixe monstro que, com sua luz (lanterna), iluminará o peixe fazendo o alarme soar. Os demais alunos serão os mergulhadores em busca do tesouro. Sempre que o peixe-robô soar o alarme os mergulhadores devem procurar um esconderijo na sala. Quando o alarme parar de soar, eles voltam a procura o tesouro.

Caso mais de um aluno queira ser o peixe, pode-se repetir a brincadeira algumas vezes.

- 2) **DEB [5 min]** O professor fará uma breve revisão do uso do bloco esperar junto com os blocos acender e apagar LED para fazer os LEDs piscarem (uma revisão simples, apenas mostrando no projetor a montagem da sequência de comandos para fazer o LED piscar).

- 3) **BRC [15 min]** O professor fará a seguinte brincadeira:

Ele desenhará no quadro branco 10 números 8 vazados, como os que existem no display de 7 segmentos (o professor pode imprimir esses 8 vazados em folhas A4 e fixar no quadro, o que tornará o processo mais prático).



Feito isso, o professor sorteará números de 0 a 9 e o nome de um aluno, esse aluno deverá então ir até o quadro e, com um marcador, pintar os "segmentos" necessários para formar o número sorteado. Os colegas podem ajudá-lo a decidir que segmentos pintar.

- 4) **DEB [5 min]** O professor explicará o que é o display de 7 segmentos e para que serve. Para tanto ele deverá apresentar uma imagem de um display de 7 segmentos com o projetor.

Durante essa explicação é importante que o professor faça referência à brincadeira anterior, explicando que no display de 7 segmentos existem 7 LEDs parecidos com aqueles que foram trabalhados nas outras aulas, e dependendo de quais LEDs ficam acesos e quais ficam apagados formam-se os números, exatamente como aconteceu na brincadeira, onde, ao invés de acender os alunos pintavam os segmentos. Importante também salientar que o bloco do DB4K para programar o display de 7 segmentos já decide quais segmentos acender e quais apagar para formar os números que estamos pedindo para ele formar.

- 5) **DEB/PRT/VER [15 min]** O professor entregará uma caixinha mágica para cada grupo (a caixinha estará equipada com os LEDs, o motor DC, o LCD e o display de 7 segmentos) e solicitará aos alunos que façam um programa para escrever os números de 0 até 9 no display de 7 segmentos da caixinha mágica. Ele reforçará que quer "ver" os 9 números aparecendo. Ele perguntará para as crianças que bloco deve ser utilizado para escrever no bloco e mostrará

o bloco no projetor. O professor deve também mostrar o bloco de limpar o display de 7 segmentos e explicar sua função.

O professor deve observar se as crianças percebem a necessidade do uso do bloco esperar, seja no momento inicial da programação ou após ver o programa executando. Caso as crianças enviem o programa para a caixinha sem o uso do bloco esperar e não entendam por que não estão conseguindo ver os números, o professor deve buscar fazê-las descobrir, com base em arguições, que modificação elas precisam fazer no programa para ele funcionar conforme o desejo delas.

- 6) **DEB/PRT/VER [30 min]** O professor apresentará o LCD por meio de uma imagem exibida pelo projetor. Ele dirá que o LCD tem duas linhas e que em cada linha pode-se escrever 16 letras (ou espaços em branco). Isso feito, ele então mostrará o bloco do LCD e perguntará para as crianças como dizer à caixinha mágica em que linha do LCD ela deve escrever. O professor apresentará também o bloco limpar LCD. Depois, o professor solicitará aos alunos que façam um programa utilizando o LCD em conjunto com o display de 7 segmentos e os LEDs. O Programa deverá fazer algum LED piscar um certo número de vezes, sendo que no LCD deve ser escrito a cor do LED e no display de 7 segmentos o número de vezes que esse LED irá piscar.

Recursos Didáticos

- Projetor
- Computadores com o DB4K instalado
- 4 Caixinhas Mágicas
- Quadro branco
- Marcadores para quadro branco.

Avaliação

1	Percebe que alguma coisa não está funcionando como esperado	5	4	3	2	1	0	NA
2	Conserva a meta original	5	4	3	2	1	0	NA
3	Tem uma hipótese para a causa do problema	5	4	3	2	1	0	NA
4	Tenta solucionar o problema	5	4	3	2	1	0	NA

1	Entende o funcionamento do display de 7 segmentos	5	4	3	2	1	0
2	Percebe a necessidade do uso do bloco esperar para ver os números sendo escritos no display	5	4	3	2	1	0
3	Entende o funcionamento do LCD	5	4	3	2	1	0

Obs:

3.8 AULA 8

Unidade didática: Construindo programas a partir da observação

Duração: 90 min

Objetivo Geral: Revisar o display de 7 segmentos e a sequência de comandos para fazer o LED piscar, bem como construir programas a partir da observação do funcionamento da caixinha mágica.

Programação: EPS - ERCtda – Atraso (delay) - Estrutura de Repetição Condicional (ERCond)

Objetivos Específicos

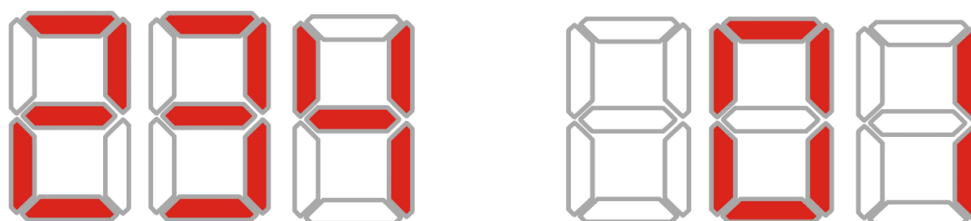
1. Revisar o uso do bloco esperar para fazer o LED piscar.
2. Revisar o funcionamento e programação do display de 7 segmentos.
3. Desenvolver a habilidade de observar um evento no mundo real e construir um programa que resulte no evento observado.

Conteúdos Novos

Desenvolvimento Metodológico

(Obs. Na oficina realizada durante a pesquisa, antes da primeira atividade, foi feita novamente a apresentação do peixe robô, uma vez que algumas crianças haviam faltado a aula anterior)

- 1) **DEB/VER [5 min]** O professor fará, novamente, uma revisão sobre o uso do bloco esperar juntamente com os blocos acender e apagar LED, verificando, por meio das respostas das crianças, o entendimento do uso do bloco esperar e deste em conjunto com os blocos dos LEDs.
- 2) **ORG [5 min]** A turma será dividida em 4 grupos e cada grupo receberá uma Caixinha Mágica (cada caixinha possuirá 4 LEDs, o Motor DC, o display de 7 segmentos e o LCD).
- 3) **DEB [5 min]** O Professor fará uma revisão sobre o uso do display de 7 segmentos, mostrando com o projetor algumas imagens de números formados com um grupo de displays desse tipo (como no exemplo abaixo), reforçando o entendimento de que o display de 7 segmentos é formado por LEDs e, dependendo de quais LEDs estão acesos ou apagados, os números são formados:



4) PRT/VER [20 min] O professor solicitará que os alunos façam os seguintes exercícios:

1)

- Escrever o número 3 no Display de 7 segmentos
- Escrever o número 5 no Display de 7 segmentos
- Escrever o resultado de $3 + 5$ no Display de 7 segmentos

2)

- Escrever o número 9 no Display de 7 segmentos
- Escrever o número 4 no Display de 7 segmentos
- Escrever o resultado de $9 - 4$ no Display de 7 segmentos

5) PRT/VER [30 min] O professor exibirá dois vídeos mostrando a caixinha mágica realizando uma sequência de ações. A cada vídeo apresentado o professor solicitará que as crianças construam no DB4K um programa que faça a caixinha mágica funcionar conforme observado.

O primeiro vídeo mostrará o LED amarelo piscando 2 vezes, depois o LED vermelho piscando 5 vezes, logo em seguida o motor DC girando por 5 segundos e, por fim, o motor DC parando e o ciclo recomeçando.

O segundo vídeo diferirá do primeiro apenas na ordem da realização das ações. O motor DC, ao invés de girar depois do LED vermelho piscar, irá girar entre as 2 piscadas do LED amarelo e as 5 piscadas do LED vermelho.

É importante o professor observar se as crianças farão o reaproveitamento do código ou se apagarão um programa para fazer o outro. Interessante também que, independentemente de as crianças terem ou não reaproveitado o código, ao final da atividade, o professor mostre as soluções com o projetor, chamando a atenção para a possibilidade do reaproveitamento.

6) PRT/VER [30 min] O professor aplicará novamente a brincadeira da palavra código, na qual cada letra do alfabeto corresponde a um certo número de piscadas de um LED de determinada cor. No entanto, desta vez, as crianças receberão alguns quadradinhos de papel com as cores dos LEDs, uma cartela para montagem da palavra código, e será exibido no projetor um gabarito com o código referente a cada letra (ver material de apoio). As crianças deverão montar palavras curtas, com três ou quatro letras. Uma vez cada grupo tendo montado suas palavras, eles farão a programação da caixinha para exibir a palavra em código por meio dos LEDs. Programadas as caixinhas, os grupos, um a um, observarão as caixinhas uns dos outros e, com base no gabarito exibido pelo projetor, tentarão descobrir qual a palavra a caixinha está escrevendo. Importante que as crianças de cada grupo trabalhem de forma cooperativa, por exemplo: enquanto alguns são responsáveis por contar as piscadas dos LEDs, outros são responsáveis por anotar esse número, outros por procurar, no gabarito, a letra correspondente ao número de piscadas e outros por anotar as letras, formando então a palavra.

(Obs. Na oficina realizada durante a pesquisa, esta atividade não foi realizada)

A	■	B	■■	C	■■■	D	■■■■
E	■■■■■	F	■■■■■■	G	■■■■■■■	H	■■■■■■■■
I	■	J	■■	K	■■■	L	■■■■
M	■■■■■	N	■■■■■■	O	■■■■■■■	P	■■■■■■■■
Q	■	R	■■	S	■■■	T	■■■■
U	■■■■■	V	■■■■■■	W	■■■■■■■	X	■■■■■■■■

SEPARA LETRAS

Exemplo de gabarito



Cartões para construção da palavra código

Recursos Didáticos

- Projetor
- Computadores com o DB4K instalado
- 4 Caixinhas Mágicas
- Imagens ilustrativas da formação de números com display de 7 segmentos (ver material de apoio)
- Vídeos com a caixinha mágica realizando uma sequência de ações (ver material de apoio)
- Imagem do gabarito da palavra código e cartões para a construção das palavras (ver material de apoio)

Avaliação

1	Percebe que alguma coisa não está funcionando como esperado	5	4	3	2	1	0	NA
2	Conserva a meta original	5	4	3	2	1	0	NA
3	Tem uma hipótese para a causa do problema	5	4	3	2	1	0	NA
4	Tenta solucionar o problema	5	4	3	2	1	0	NA

1	Utiliza corretamente o bloco do Display de 7 Segmentos	5	4	3	2	1	0
3	Utiliza corretamente o bloco repetir	5	4	3	2	1	0
5	Percebe a necessidade do uso do Bloco esperar	5	4	3	2	1	0
6	Entende a lógica de construção do trecho de programa para piscar o LED	5	4	3	2	1	0
7	Consegue algoritmizar os programas apresentados nos vídeos	5	4	3	2	1	0

Obs.

3.9 AULA 9

Unidade didática: LCD, os sensores de Luz e Temperatura e o Bloco Enquanto

Duração: 90 min

Objetivo Geral: Conhecer os sensores de luz e temperatura e exercitar o uso do LCD

Programação: EPS - ERCtda – Atraso (delay) - Estrutura de Repetição Condicional (ERCond)

Objetivos Específicos

1. Exercitar o uso do LCD
2. Conhecer o sensor de Luz
3. Conhecer o sensor de temperatura
4. Entender que os sensores percebem modificações no ambiente e pode-se programar o robô para, com base nessa percepção, mudar o seu comportamento.
5. Conhecer a estrutura de repetição condicional *enquanto <condição> faça*

Conteúdos Novos

1. Sensor de Luz
2. Sensor de Temperatura
3. Estrutura de Repetição Condicional (Bloco Enquanto <condição> faça)

Desenvolvimento Metodológico

1) **ORG [5 min]** A turma será dividida em 4 grupos e cada grupo receberá uma lanterna e uma Caixa Mágica (cada caixinha possuirá 4 LEDs, o motor DC, o display de 7 segmentos, o LCD, o sensor de luz e o sensor de temperatura).

2) **DEB/PRT/VER [40 min]** O Professor fará uma revisão sobre o uso do LCD, mostrando a sua imagem com o projetor, reforçando o fato de que ele possui duas linhas e pode-se escrever 16 letras e/ou espaços em branco em cada linha. O professor deve lembrar aos alunos também que o DB4K não possibilita que se escreva acentos ou cedilha. Isso feito, o professor solicitará que os alunos façam os dois programas a seguir, observando se as crianças lembram de usar o bloco esperar e, caso esqueçam, se percebem, ao olhar a execução do programa na caixinha, que o resultado observado não é o esperado devido à ausência dos atrasos.

1)

- Escrever o nome na linha 1 do LCD.
- Escrever o último sobrenome na linha 2 do LCD.
- Limpar o LCD
- Escrever a palavra Curso na linha 1 do LCD.
- Escrever “Robótica” na linha 2 do LCD.
- Limpar o LCD

2)

- Escrever o que quiser na linha 1 do LCD.
- Escrever o que quiser na linha 2 do LCD.
- Limpar o LCD

3) **DEB/BRC [10 min]** O professor fará a brincadeira do robô humano para introdução da estrutura de repetição condicional. Ele deverá, inicialmente, fazer uma breve revisão sobre o bloco repetir, conversando com as crianças a respeito do seu funcionamento. Isso feito, falará que existe um bloco chamado “enquanto” que também faz os comandos serem repetidos, mas, ao invés de o programador dizer o número de repetições, ele dirá em qual “situação”, sob que “condição” essa repetição ocorrerá. O professor então mostrará o programa em blocos no projetor e pedirá para as crianças explicarem o seu funcionamento. Uma vez tendo-se debatido o funcionamento do programa, a brincadeira terá início: as crianças lerão os comandos em sequência e dançarão enquanto a música estiver tocando.



4) **DEB [10 min]** O professor irá apresentar no projetor o bloco enquanto e, novamente, fará um paralelo entre os blocos enquanto e repetir, reforçando que: o bloco repetir repete os

comandos um número de vezes que “a gente” estipula, o bloco “enquanto” repete os comandos enquanto “alguma coisa estiver acontecendo”, por exemplo, a música tocar.

Apresentado o bloco, o professor explicará que os robôs utilizam os sensores para perceber o ambiente, assim eles podem saber se está quente, frio, se existe muita luz ou pouca luz no ambiente, entre outras coisas. Desse modo pode-se programar o robô, por exemplo, para realizar algum comando somente enquanto luz estiver forte ou somente enquanto estiver calor.

Isso feito, o professor mostrará, no projetor, o sensor de luz e o bloco do sensor de luz, e logo em seguida, um exemplo de como utilizar o bloco enquanto com os sensores. Ele então pegará o bloco do sensor de luz, encaixará no bloco enquanto, e comentará com as crianças que é possível programar o robô, com o bloco do sensor de luz, para verificar se existe muita ou pouca luz no ambiente. O professor dirá então que é assim, utilizando o bloco enquanto junto com o bloco do sensor e luz, que se programa o robô para só realizar alguma ação, por exemplo, se o ambiente estiver escuro.

5) PRT/VER [10 min] O Professor montará, junto com as crianças, passo a passo, o programa abaixo, sempre questionando e debatendo o porquê de cada bloco utilizado.

- Acender o LED Verde,
- Enquanto houver muita luz
 - Apagar o LED Verde
 - Girar o Motor DC Rápido
- Parar o Motor DC

Finalizado o programa, as crianças o enviarão para a caixinha e observarão seu funcionamento. O professor solicitará então que as crianças localizem o sensor de luz na caixinha mágica e o iluminem com as lanternas, observando a mudança de funcionamento da caixinha. Isso feito, o professor perguntará às crianças o que acontece enquanto a lanterna fica iluminando a caixinha e o que acontece quando a lanterna não a está iluminando.

5) DEB [5 min] O Professor apresentará para as crianças o sensor de temperatura (mostrando a imagem deste sensor com o projetor) e explicará que ele percebe alterações de temperatura no ambiente. Além disso, o professor explicará que o corpo humano tem uma temperatura por volta de 36 graus (pode inclusive comentar sobre a febre, quando a mãe das crianças mede a temperatura delas, etc.) e que a sala de aula está com menos de 36 graus, por causa do ar condicionado que deixa ela geladinha (se for possível o professor pode mostrar a temperatura da sala no display do ar-condicionado, se houver. Se a sala não for refrigerada, basta o professor dizer que ela está com menos de 36 graus ou levar um termômetro de ambiente para mostrar a temperatura da sala). Isso explicado, o professor dirá aos alunos que, como a temperatura deles está mais alta que a do ambiente, se eles segurarem o sensor de temperatura com os dedos, a temperatura no sensor irá aumentar e “a caixinha” poderá ser programada para perceber esse aumento.

6) PRT/DEB/VER [10 min] Uma vez tendo sido dadas as explicações sobre o sensor de temperatura, o professor solicitará que as crianças substituam o bloco do sensor de luz pelo bloco do sensor de temperatura no programa da atividade anterior. Isso feito, o professor solicitará que as crianças enviem o programa para a caixinha e iluminem o sensor de luz para ver o

que acontece. Depois disso, perguntará para as crianças o porquê de o comportamento da caixinha não ter modificado.

Finalizado o debate, o professor solicitará que as crianças segurem o sensor de temperatura com os dedos e aguardem um tempo para ver o que acontece, e então promoverá um novo debate, agora para que se discuta porque, desta vez, segurar o sensor de temperatura, ao invés de iluminar o sensor de luz, é o que está fazendo a caixinha mudar de comportamento.

O professor pode também pedir para as crianças colocarem o dedo sobre o sensor de luz (ou iluminar o sensor de temperatura) para que elas percebam que nada acontece, promovendo então um debate sobre esta situação.

Recursos Didáticos

- Projetor
- Computadores com o DB4K instalado
- 4 Caixinhas Mágicas
- 4 Lanternas
- Programa para o robô humano a ser apresentado no projetor
- Música para a atividade do robô humano
- Caixas de som para a atividade do robô humano

Avaliação

1	Percebe que alguma coisa não está funcionando como esperado	5	4	3	2	1	0	NA
2	Conserva a meta original	5	4	3	2	1	0	NA
3	Tem uma hipótese para a causa do problema	5	4	3	2	1	0	NA
4	Tenta solucionar o problema	5	4	3	2	1	0	NA

1	Utiliza corretamente os blocos do LCD	5	4	3	2	1	0
2	Percebe a necessidade do uso do bloco esperar	5	4	3	2	1	0
3	Entende a utilização do Bloco do sensor de Luz	5	4	3	2	1	0
4	Entende a utilização do Bloco do sensor de temperatura	5	4	3	2	1	0
5	Entende a utilização do Bloco enquanto.						

Obs:

3.10 AULA 10

Unidade didática: Programando o Peixe Robô

Duração: 90 min

Objetivo Geral: Apresentar um exemplo prático do uso do sensor de luz em conjunto com alguns atuadores

Programação: EPS – ERctda - ERCond - Atraso

Objetivos Específicos

1. Exercitar o desenvolvimento de um programa de computador a partir da observação de um evento do mundo real.
2. Aprender a programar o sensor de luz em combinação com alguns atuadores.
3. Aprender a programar o LED RGB.
4. Exercitar a distinção entre os blocos repetir e enquanto.
5. Apresentar uma aplicação prática do motor DC, do buzzer e do sensor de luz.

Conteúdos Novos

1. Buzzer
2. LED RGB

Desenvolvimento Metodológico

1) **ORG [5 min]** A turma será dividida em 4 grupos, e cada grupo receberá uma lanterna e uma Caixinha Mágica (cada caixinha possuirá 4 LEDs, o LED RGB, o Motor DC, o display de 7 segmentos, o LCD, o buzzer o sensor de luz e o sensor de temperatura).

2) **PRT/VER/DEB [30 min]** O professor exibirá dois vídeos mostrando a caixinha mágica realizando uma sequência de ações. Após a apresentação de cada vídeo o professor solicitará que as crianças construam no DB4K um programa que faça a caixinha mágica funcionar conforme o observado.

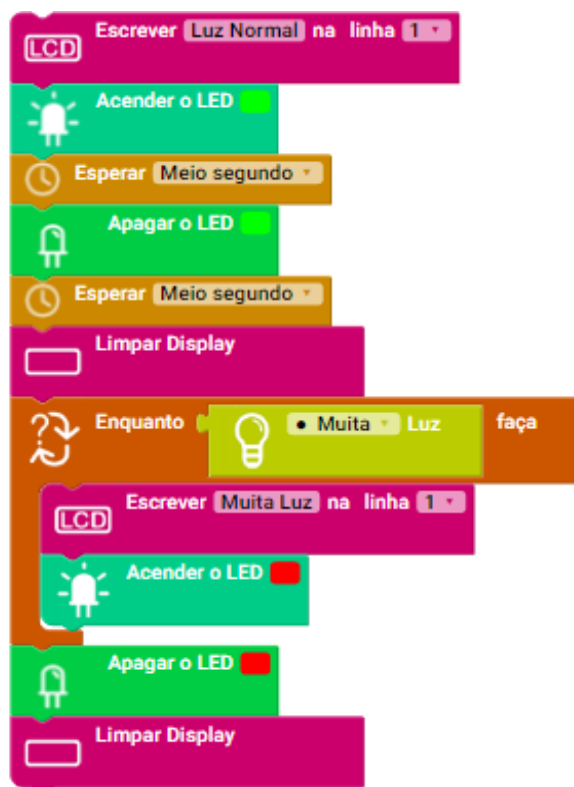
Durante a exibição, o professor deve solicitar aos alunos que descrevam oralmente o que estão observando.

O primeiro vídeo mostrará um loop contínuo da seguinte sequência de ações: a palavra “azul” aparecendo no LCD, o LCD limpo, o número 4 aparecendo no display de 7 segmentos, o display de 7 segmentos apagado, o LED azul piscando 4 vezes

Antes de apresentar o segundo vídeo, o professor deve fazer uma revisão sobre o uso do bloco enquanto. Esse segundo vídeo mostrará a caixinha exibindo dois comportamentos distintos: um enquanto o sensor de luz estiver sendo iluminado por uma lanterna e outro quando ele não estiver sendo iluminado pela lanterna (ver a imagem a seguir).

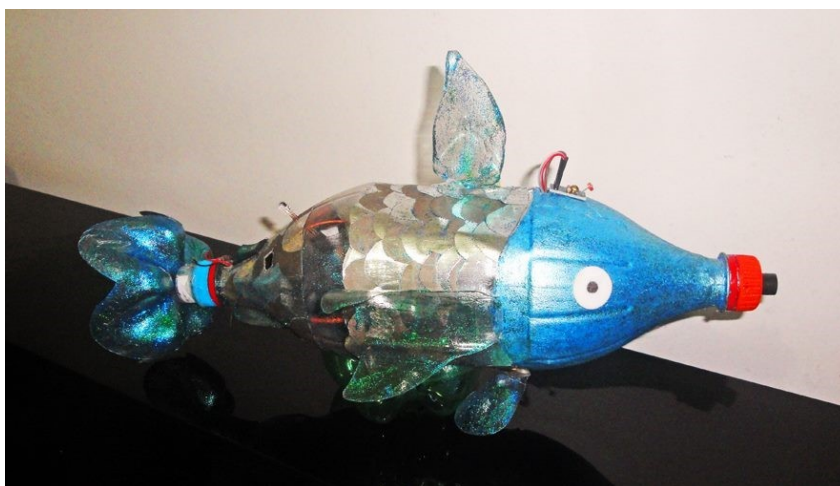
O professor deve aproveitar a oportunidade para comentar com as crianças que, quando se utiliza o bloco enquanto, o tempo que uma ação será repetida será definido pelo ambiente e não mais determinado na programação. Ele deve fazer isso utilizando o exemplo do programa construído e mostrar que o LED vermelho ficará aceso durante o tempo que a lanterna permanecer iluminando o sensor de luz, e o LED verde ficará piscando durante o tempo em que o sensor de luz não estiver sendo iluminado. É

importante que o professor mostre isso aos alunos iluminando o sensor e contando em voz alta o tempo que o LED vermelho permanece aceso, bem como contando em voz alta o número de vezes que o LED verde pisca enquanto não há luz forte.



Programa a ser executado pela caxinha com o uso do bloco enquanto.

- 2) **DEM/DEB [10 min]** O professor lembrará brevemente a história do peixe robô e o reapresentará, em detalhes, lançando perguntas sobre os dispositivos utilizados na sua construção, bem como sobre o seu funcionamento. O professor aproveitará a demonstração para apresentar o buzzer e o LED RGB, dispositivos ainda não visto em detalhe pelas crianças.



O peixe robô possui um motor DC no rabo, um sensor de luz no topo da cabeça, um buzzer na “boca”, um LED RGB no dorso e um LED vermelho na parte interna, onde estão posicionadas também a placa Arduino uno e a bateria (ver material de apoio).

3) **PRT/VER [45 min]** O professor solicitará que os alunos construam programas para fazer o peixe-robô funcionar conforme o observado. As duplas devem testar os programas na caixinha mágica, uma vez que ela possui todos os dispositivos presentes no peixe robô. Por fim, quando acreditarem que o programa está correto, as duplas carregarão seu programa no peixe-robô para certificar-se de que tudo ocorrerá conforme o planejado e testado na caixinha.

Recursos Didáticos

- Projetor
- Computadores com o DB4K instalado
- 4 caixinhas mágicas
- Vídeos para construção do programa a partir da observação do funcionamento da caixinha mágica (ver material de apoio)
- Peixe robô (ver material de apoio)
- 4 lanternas

Avaliação

1	Percebem que alguma coisa não está funcionando como esperado	5	4	3	2	1	0	NA
2	Conservam a meta original	5	4	3	2	1	0	NA
3	Têm uma hipótese para a causa do problema	5	4	3	2	1	0	NA
4	Tentam solucionar o problema	5	4	3	2	1	0	NA

1	Conseguem “algoritimizar” o funcionamento da caixinha a partir dos vídeos apresentados	5	4	3	2	1	0
2	Usa corretamente o bloco enquanto	5	4	3	2	1	0
3	Usa corretamente o bloco repetir	5	4	3	2	1	0
4	Usa adequadamente o bloco Esperar para visualizar a “atuação” dos dispositivos da caixinha	5	4	3	2	1	0

Obs:

3.11 AULA 11

Unidade didática: A Legião de Vampiros: O Morcego Robô

Duração: 90 min

Objetivo Geral: Conhecer o sensor de distância e aprender a programá-lo em conjunto com alguns atuadores.

Programação: EPS - ERCtda – ERCond – Atraso

Objetivos Específicos

1. Conhecer o sensor de distância e para que serve.
2. Exercitar o desenvolvimento de um programa de computador a partir da observação de um evento do mundo real.
3. Desenvolver programas para o sensor de distância em combinação com alguns atuadores.

Conteúdos Novos

1. Sensor de distância

Desenvolvimento Metodológico

- 1) **DEB [10 min]** O professor explicará o que é o Sensor de Distância e para que serve, incluindo nessa explicação um pequeno vídeo que mostra como os morcegos “enxergam” a noite (<https://www.youtube.com/watch?v=l4iUD-CAFY0>) e aproveitará para relembrar a conversa do segundo encontro onde foi feito um debate sobre os sensores.
- 2) **NAR/PRT/VER [80 min]** O professor comentará que agora todos os alunos da turma fazem parte de uma legião de vampiros super conectados e modernos que estão desenvolvendo um esquadrão de robôs morcego. O projeto dos morcegos está sendo feito por engenheiros vampiros, sendo que um primeiro protótipo já foi desenhado. A função desses morcegos é ficar de vigia nas imediações do esconderijo dos vampiros e, quando algum intruso se aproximar, disparar um sinal de rádio para avisar da presença do invasor e girar uma hélice para voar para longe dele.



O morcego robô possui um sensor de distância nos “olhos”, um LED vermelho no “nariz” e um motor DC no topo da cabeça. Ele possui uma abertura na parte traseira onde é encaixada uma placa Arduino Uno (ver material de apoio).

O professor dirá então que os alunos da turma são vampiros programadores e que ficaram responsáveis por desenvolver o primeiro programa de teste dos morcegos robôs. Dirá também que, embora o Morcego robô possua uma hélice responsável por fazê-lo voar, nesse primeiro protótipo, o motor ainda não possui potência suficiente para fazer o morcego levantar voo. Além disso, o dispositivo para enviar o sinal de rádio também não está disponível e será usado um LED vermelho para simular esse sinal.

O professor mostrará um morcego robô pronto e apresentará seu funcionamento: Ficar parado piscando o LED vermelho e, enquanto o morcego estiver a menos de 15 cm de

distância de algum objeto (ou seja, quando algo se aproximar do morcego robô), parar de piscar o LED, mantendo-o aceso (simulando a mudança no sinal de rádio enviado à base dos vampiros), e girar a hélice rápido para escapar voando.

O professor distribuirá então os morcegos robô semi montados para os alunos e entregará os materiais para a confecção das asas e das hélices. Cada aluno fará duas asas e uma hélice e as fixará no morcego robô no momento de testar a programação (cada dupla receberá um morcego)



Robôs morcego semi montados (ver material de apoio)

Uma vez as asas e as hélices finalizadas, os alunos deverão programar os morcegos para funcionar da mesma forma que o do professor.

Recursos Didáticos

- Projetor
- Computadores com o DB4K instalado
- 4 Caixinhas Mágicas
- 4 morcegos de garrafa pet semi montados (ver material de apoio)
- 4 placas Arduino Uno
- 4 motores DC
- Material para as crianças fazerem as asas, a hélice a boca e os dentes do morcego robô (ver material de apoio)
- Vídeo sobre a “visão” dos morcegos (ver material de apoio)
- Fita adesiva transparente

Avaliação

1	Percebe que alguma coisa não está funcionando como esperado	5	4	3	2	1	0	NA
2	Conserva a meta original	5	4	3	2	1	0	NA
3	Tem uma hipótese para a causa do problema	5	4	3	2	1	0	NA
4	Tenta solucionar o problema	5	4	3	2	1	0	NA

1	Entende o funcionamento do sensor de distância	5	4	3	2	1	0
2	Utiliza corretamente o bloco Enquanto <condição> faça, inserindo o bloco com o sensor correto como argumento	5	4	3	2	1	0
3	Programa corretamente o sensor de distância em conjunto com o motor DC e o LED	5	4	3	2	1	0

Obs:

3.12 AULA 12

Unidade didática: A Legião de Vampiros: O Carrinho Robô Vampiro

Duração: 90 min

Objetivo Geral: Reforçar o entendimento da programação do sensor de distância. Aprender a programar o carrinho robô em conjunto com alguns sensores e atuadores, bem como utilizar mais de um bloco de repetição condicional em um mesmo programa.

Programação: EPS - ERCond – Atraso

Objetivos Específicos

1. Apresentar uma aplicação prática da combinação dos sensores de luz e distância e do buzzer.
2. Apresentar uma aplicação prática da utilização de duas estruturas de repetição condicionais.
3. Exercitar o desenvolvimento de um programa de computador a partir da observação de um evento do mundo real.
4. Exercitar o uso da estrutura de repetição condicional.

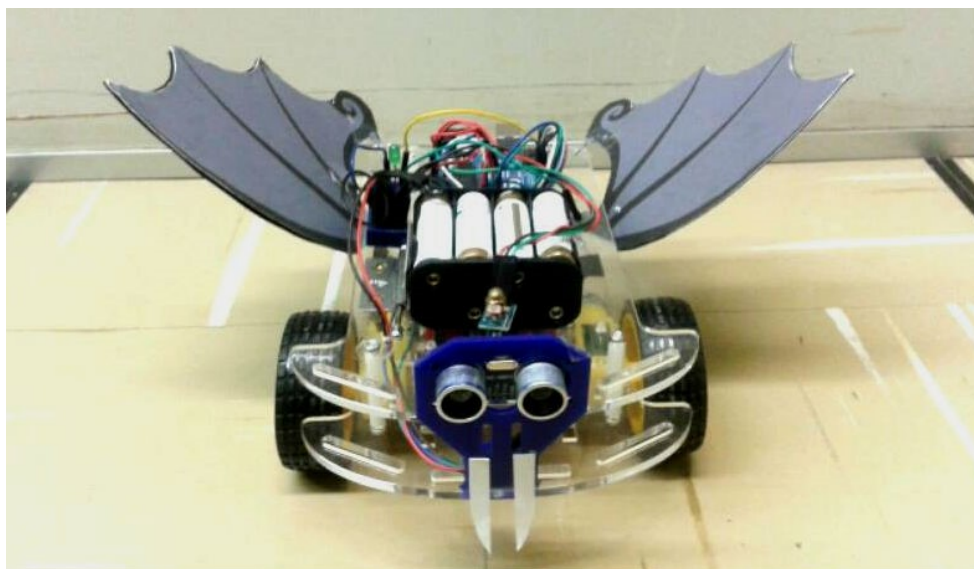
Conteúdos Novos

1. Carrinho Robótico

Desenvolvimento Metodológico

- 1) **DEM/DEB/VER [10 min]** O professor fará uma revisão sobre o uso do bloco enquanto, montando alguns programas simples (exibidos no projetor), colocando comandos fora e dentro do bloco e perguntando qual o resultado esperado em cada uma das situações.
- 2) **PRT/VER [20 min]** O professor distribuirá os morcegos robô construídos na última aula e solicitará que as crianças programem esses robôs para funcionarem de uma forma diferente daquela que eles programaram na aula anterior, ou seja, à escolha deles.
- 1) **NAR/DEM/DEB/PRT [60 min]** O professor dirá aos alunos que os engenheiros da legião de vampiros criaram um novo robô: um carrinho robô vampiro. A função desse carrinho é ir na frente dos vampiros da legião quando eles estiverem explorando lugares desconhecidos, soando um alarme sempre que entrarem em algum lugar iluminado, avisando assim para os vampiros manterem distância daquele local. Os alunos, como vampiros programadores, têm a missão de criar um programa para o carrinho vampiro funcionar da maneira que eles irão observar: andar para frente desviando dos objetos e, quando for iluminado, soar um alarme com o buzzer, andar um pouco para trás, depois virar para algum lado e voltar a andar para frente desviando de objetos e investigando o ambiente.

O carrinho robô será então reapresentado aos alunos, agora ornado com duas asas de morcego e equipado com um buzzer e com os sensores de distância e de Luz, de modo que as crianças possam observar que os sensores que eles aprenderam a programar podem ser utilizados para fazer com que o carrinho “reaja” ao ambiente, e que o buzzer pode, por exemplo, ser utilizado para indicar que o carrinho “percebeu” alguma mudança nesse ambiente.



O professor mostrará então o carrinho robô para a turma, equipado com os sensores de luz e distância, com um buzzer, um LED, e ornado com duas asas de morcego. As crianças poderão brincar um pouco com o carrinho para observar seu funcionamento.

Isso feito, o professor solicitará que os alunos desenvolvam um programa para o carrinho vampiro funcionar conforme o observado. Uma vez esse programa finalizado e testado no carrinho, os alunos que desejarem poderão criar programas para o carrinho funcionar de outra maneira, distinta daquela apresentada pelo professor, utilizando inclusive o LED.

(Quando da realização desta atividade na oficina realizada durante a pesquisa, o buzzer não foi utilizado)

Recursos Didáticos

- Projetor
- Computadores com o DB4K instalado
- 1 Carrinho Robótico (caracterizado de morcego, com a inclusão de duas asas e dois dentes)
- 4 lanternas
- Asas e dentes para o carrinho Vampiro (Ver material de apoio)
- Morcegos robô de garrafa pet montados na aula anterior

Avaliação

1	Percebe que alguma coisa não está funcionando como esperado	5	4	3	2	1	0	NA
2	Conserva a meta original	5	4	3	2	1	0	NA
3	Tem uma hipótese para a causa do problema	5	4	3	2	1	0	NA
4	Tenta solucionar o problema	5	4	3	2	1	0	NA

1	Percebe a necessidade do uso da estrutura de repetição condicional (enquanto <condição> faça) para resolver os desafios propostos	5	4	3	2	1	0
2	Utiliza corretamente o bloco "enquanto <condição> faça", inserindo o bloco com o sensor correto como argumento.	5	4	3	2	1	0
3	Percebe a necessidade do uso de dois blocos enquanto, um para cada um dos sensores.	5	4	3	2	1	0
4	Utiliza corretamente o bloco esperar	5	4	3	2	1	0

Obs:

3.13 AULA 13

Unidade didática: O Cachorro Robô e o Braço Robótico:
A Missão Espacial

Duração: 90 min

Objetivo Geral: Trabalhar o servo-motor e o braço robótico

Programação: EPS - ERCond – Atraso

Objetivos Específicos

1. Aprender a programar o servo-motor.
2. Praticar a construção de programas a partir da observação de eventos no mundo real.
3. Praticar a construção de programas com o uso de mais de um laço de repetição condicional (uso de mais de um sensor).
4. Aprender a programar o braço-robótico.
5. Apresentar aplicações práticas do uso do servo-motor.

Conteúdos

1. Servo motor

Desenvolvimento Metodológico

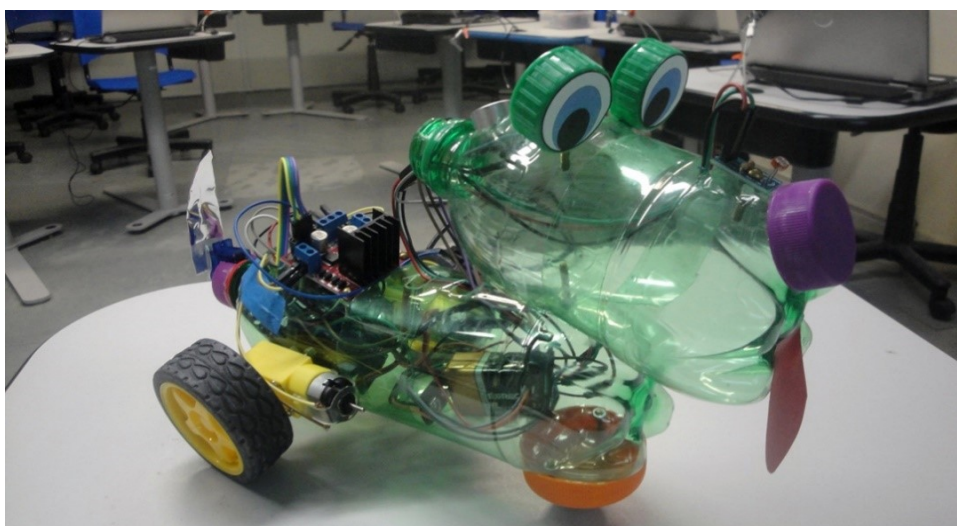
- 1) **DEM/DEB [10 min]** O professor apresentará o cachorro robô à turma e mostrará em detalhes cada uma de suas partes, perguntando às crianças se elas reconhecem os dispositivos que estão vendo, se eles já foram utilizados em outros robôs trabalhados em aula, etc. É

interessante observar se elas percebem que as rodas do cachorro são as mesmas do carrinho robô, e chamar a atenção para este fato: pode-se utilizar os mesmos dispositivos para diferentes aplicações.

Isso feito, o professor mostrará o funcionamento do cachorro robô: sempre que alguém faz carinho na cabeça dele, ele balança o rabinho, e sempre que alguém ilumina o focinho ele vai em direção dessa pessoa.

Algumas crianças certamente dirão ao professor que querem brincar com o cachorro, utilizando a lanterna para fazê-lo mover-se. O professor dirá então que, para isso, elas mesmas terão que construir o programa de controle do cachorro robô.

Durante a apresentação do funcionamento do cachorro, o professor estimulará as crianças, com base em questionamentos, a descreverem os comportamentos apresentados pelo robô, de modo que elas possam, a partir dessa observação, construir o programa de controle.



O cachorro robô possui um sensor de distância no topo da cabeça, um sensor de luz no focinho, um servo motor para controlar o rabo, duas rodas com redutores na parte traseira, uma roda esférica na parte dianteira e, internamente, possui uma placa Arduino Uno, uma ponte h e uma bateria (ver material de apoio).

- 2) **DEB/ [5 min]** O professor apresentará no projetor uma imagem do servo motor e também explicará, com o auxílio de uma animação (feita no PowerPoint), o seu funcionamento (ver material de apoio). A animação deve simular um movimento contínuo do ponteiro do servo de 45 até 135 graus, simulando o movimento do rabinho do cachorro. O professor deverá perguntar às crianças, ao mostrar a animação, se elas viram alguma coisa funcionando daquela maneira, caso a resposta não venha, ele informa: o rabinho do cachorro funciona assim.
- 3) **PRT/VER [10 min]** O professor distribuirá as caixinhas mágicas (que agora estará completa, com todos os sensores e atuadores) e fixará uma miniatura do rabinho do cachorro robô em cima dos ponteiros dos servo-motores (ver material de apoio). Isso feito, ele mostrará o bloco do servo motor e como utilizá-lo no programa, ou seja, como definir a posição para a qual deseja-se movimentar o ponteiro do servo, e solicitará às crianças que programem o servo para funcionar como o rabinho do cachorro.

4) PRT/VER [10 min] O professor perguntará às crianças o que faz com que o cachorro robô balance o rabo. Se a resposta não vier, ele dirá: aproximar a mão do sensor de distância presente na cabeça dele. Feita esta constatação, o professor solicitará às crianças que programem a caixinha mágica de modo que o servo-motor só se movimente quando a mão delas se aproximar do sensor de distância presente na caixinha.

5) PRT/VER [20 min] O professor solicitará aos alunos que construam a parte do programa responsável por fazer o cachorro mover-se para frente quando o focinho dele for iluminado por uma lanterna. Ele perguntará que bloco elas acreditam que deve ser utilizado para isso, se a resposta não vier, ele pode perguntar: O cachorro não utiliza os mesmos dispositivos que o carrinho para andar? Será que não podemos então utilizar os blocos do carrinho para programar os movimentos do cachorro?

Conforme as crianças forem terminando os programas ele será carregado no cachorro para que elas observem se ele está funcionando conforme o esperado

NAR/DEM/DEB [35 min] O professor rerepresentará o braço robótico à turma (ver material de apoio) e contará a seguinte história:

Uma equipe de astronautas foi enviada ao espaço para posicionar duas câmeras de altíssima resolução em um satélite responsável por permitir à Assistência Humanitária Internacional obter imagens da terra com maior precisão e, assim, ter mais sucesso nas missões de ajuda em caso de desastres de grandes proporções em qualquer país do mundo.

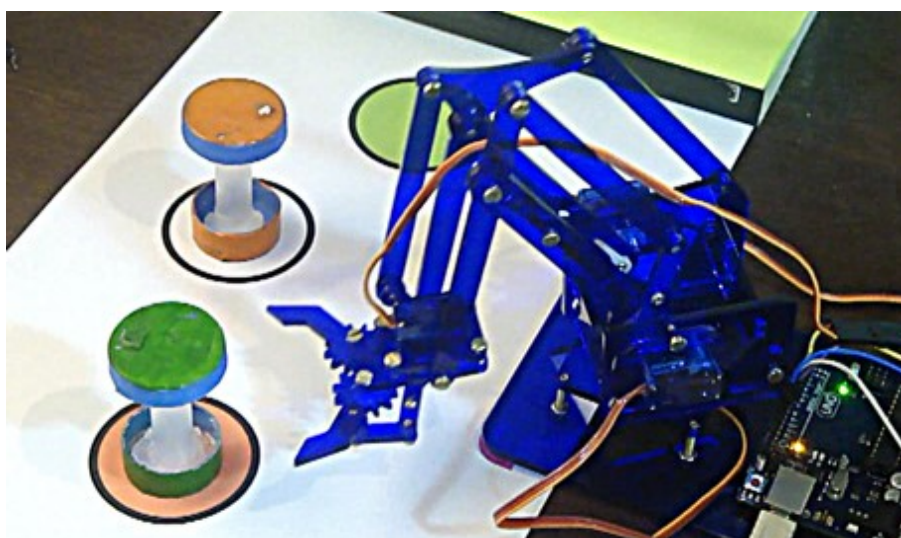
No entanto, depois que os astronautas voltaram, percebeu-se que houve um erro no posicionamento das duas câmeras no satélite e, assim, elas precisam ser trocadas de lugar.

Não há como enviar uma nova missão tripulada em pouco tempo, ou seja, uma missão com astronautas. Então, será preciso enviar um braço robótico ao espaço para que ele realize a troca da posição das câmeras.

Esse braço deverá ser programado na terra e, chegando no espaço, realizar exatamente a sequência de ações definidas em seu programa.

Vocês fazem parte da equipe de programadores da agência espacial responsável por enviar o braço robótico ao espaço, e deverão programar o braço para fazer a seguinte troca:

O professor mostrará então um tabuleiro com 3 círculos (sendo um branco, um alaranjado e outro verde) e dois peões (um verde e um alaranjado) os peões não estarão posicionados nos círculos da mesma cor (ver material de apoio). O professor dirá aos alunos que os peões são as câmeras e que os círculos representam o local correto onde elas deveriam estar encaixadas. O professor perguntará então aos alunos como fazer para colocar as câmeras nos encaixes da mesma cor, sendo que elas só podem ser posicionadas nos encaixes (ou seja, não podem ser largadas soltas no espaço) e só é possível movimentar uma câmera de cada vez.



Uma vez as crianças tendo descoberto os movimentos necessários (uma dentre as possibilidades), o professor reapresentará o braço robótico a elas, mostrando que os seus movimentos são controlados por 4 servo-motores.

Isso feito, o professor ligará o braço robótico e mostrará ele trocando os peões de lugar, posicionando-os sobre os círculos corretos.

Finalizada a demonstração, o professor solicitará às crianças que construam um programa para o braço realizar aquela sequência de movimentos. Para a realização dessa tarefa, o professor dará a cada criança um tabuleiro idêntico ao utilizado na demonstração, e duas tampinhas (uma verde e uma alaranjada) para que elas possam simular os movimentos do braço no momento de construírem o programa.

Conforme as crianças forem terminando os programas, eles serão carregados no braço robótico para que elas possam observar a sua execução.

(Na oficina realizada durante a pesquisa as crianças não receberam os tabuleiros com as tampinhas)

Recursos Didáticos

- Projetor
- Computadores com o DB4K instalado
- 4 caixinhas mágicas
- Um cachorro robô (ver material de apoio)
- 4 lanternas
- Um braço robótico (ver material de apoio)
- 1 tabuleiro e 2 peões para o braço robótico (ver material de apoio)
- 1 cópia tabuleiro do braço robótico e 2 tampinhas (uma alaranjada e outra verde) para cada criança (ver material de apoio)

Avaliação

1	Percebe que alguma coisa não está funcionando como esperado	5	4	3	2	1	0	NA
2	Conserva a meta original	5	4	3	2	1	0	NA
3	Tem uma hipótese para a causa do problema	5	4	3	2	1	0	NA
4	Tenta solucionar o problema	5	4	3	2	1	0	NA

1	Entende o funcionamento do servo motor	5	4	3	2	1	0
	Programa corretamente o servo motor	5	4	3	2	1	0
1	Percebe a necessidade do uso da estrutura de repetição condicional (enquanto <condição> faça) para resolver os desafios propostos	5	4	3	2	1	0
2	Utiliza corretamente o bloco "enquanto <condição> faça", inserindo o bloco com o sensor correto como argumento.	5	4	3	2	1	0
3	Percebe a necessidade do uso de dois blocos enquanto, um para cada um dos sensores.	5	4	3	2	1	0
4	Utiliza corretamente o bloco esperar	5	4	3	2	1	0
5	Programa corretamente o braço robótico para realizar os movimentos desejados.	5	4	3	2	1	0

Obs:

3.14 AULA 14**Unidade didática:** Finalizando a oficina: Projetando um Robô**Duração:** 90 min**Objetivo Geral:** Fazer um apanhado geral dos conteúdos trabalhados durante a oficina**Objetivos Específicos**

1. Observar se as crianças conseguem produzir um texto descrevendo o resultado esperado quando da execução de um determinado código a elas apresentado.
2. Verificar se as crianças sabem os nomes de todos os atuadores e sensores trabalhados em aula.
3. Observar se as crianças conseguem verificar se um determinado código corresponde a um comportamento exibido pela caixinha mágica.
4. Verificar se as crianças conseguem “projetar” robôs que utilizem atuadores e sensores trabalhados em aula, definindo de forma coerente a função de cada um desses dispositivos nos robôs projetados.

Conteúdos Novos

Desenvolvimento Metodológico

- 1) **VER [25 min]** O professor apresentará no projetor um programa feito no DB4K e, através de um texto, as crianças deverão descrever o seu funcionamento (ver material de apoio):



Exemplo de uma possível descrição para o programa anterior: Esse programa fará a caixinha mágica piscar o LED vermelho 3 vezes, depois girar o motor DC devagar por 4 segundos, depois repetir 4 vezes isso: acender a luz vermelha do LED RGB por meio segundo, depois a azul por meio segundo, fazendo a gente ver outra cor, depois deixar o LED RGB apagado por meio segundo. A caixinha vai ficar fazendo isso sem parar.

- 2) **VER [5 min]** O professor entregará para as crianças lápis e folhas A4 onde em uma coluna haverá o nome de alguns sensores e atuadores trabalhados em aula e, em uma coluna ao lado, imagens desses sensores e atuadores. As crianças deverão ligar os nomes dos dispositivos às imagens corretas (ver material de apoio).

Ligue os dispositivos aos seus nomes



LED



Display 7 Segmentos



Sensor de Luz



Sensor de Distância



Servo-Motor



Motor DC



Buzzer



LCD

- 3) **VER [10 min]** O professor apresentará no projetor um programa feito no DB4K e, ao lado, um vídeo com a caixinha mágica executando um determinado programa. As crianças deverão verificar se o programa exibido corresponde ao que a caixinha está executando, caso contrário, indicar quais modificações devem ser realizadas no código para que ele corresponda ao programa sendo executado pela caixinha mágica. O professor pode fazer mais de um desses exercícios, uns nos quais os dois programas sejam os mesmos e outros nos quais haja diferenças entre o programa apresentado e aquele sendo executado pela caixinha (ver material de apoio).
- 4) **VER [30 min]** as crianças receberão papéis cartão A4, lápis, borracha e lápis de cor para desenhar um robô que deverá, necessariamente, usar alguns dos sensores e atuadores trabalhados em aula. Por fim, ao terminarem o desenho, os alunos deverão explicar ao professor o funcionamento dos robôs por eles projetados. As crianças devem ser estimuladas a fazerem o robô que quiserem, por mais maluco que possa parecer. Elas podem, inclusive,

usar nos robôs, outros sensores e atuadores além dos vistos em aula, inclusive sensores e atuadores que não existam, como por exemplo, um sensor de sentimentos.

5) **ENCERRAMENTO [20 min]** Tempo reservado para uma conversa final sobre a oficina, despedidas, etc.

Recursos Didáticos

- Projetor
- Papel cartão A4 branco (um para cada aluno)
- Lápis
- Lápis de cor
- Borrachas
- Folhas pautadas (uma para cada aluno)
- Vídeos e imagens dos programas para a atividade 3 (ver material de apoio)
- Folhas impressas para a atividade 2 (ver material de apoio)

Avaliação

1	Desenvolve um texto descrevendo o resultado esperado quando da execução do código apresentado	5	4	3	2	1	0
2	Identifica se o código apresentado corresponde ao comportamento apresentado pela caixinha, indicando os pontos corretos e incorretos do programa?	5	4	3	2	1	0
3	Identificar os atuadores e sensores trabalhados em aula	5	4	3	2	1	0
4	Utiliza adequadamente sensores e atuadores no projeto do robô e descreve (oralmente) para o robô projetado, um comportamento coerente com os atuadores e sensores utilizados.	5	4	3	2	1	0

Obs:

APÊNDICE B – ROTEIRO DE AVALIAÇÃO DE USABILIDADE E *FUN*

1 ROTEIRO DE OBSERVAÇÃO (AMBIENTE DE PROGRAMAÇÃO DB4K)	236
2 QUESTIONÁRIOS (AMBIENTE DE PROGRAMAÇÃO E BRINCADEIRAS)	237
2.1 QUESTIONÁRIO 1 - Aplicado na primeira aula da oficina	237
2.2 QUESTIONÁRIO 2 - Aplicado na segunda aula da oficina	238
2.3 QUESTIONÁRIO 3 - Aplicado na terceira aula da oficina	239
2.4 QUESTIONÁRIO 4 - Aplicado na quarta aula da oficina	240
2.5 QUESTIONÁRIO 5 - Aplicado na nona aula da oficina	241

1 ROTEIRO DE OBSERVAÇÃO (Ambiente de Programação DB4K)

1.1 Modo de busca dos blocos nos menus

- a) Observar se, com a prática, os alunos passam a acessar diretamente o item de menu da caixa de ferramentas correspondente ao bloco que precisam ou se navegam nos menus de um em um até encontrarem o bloco desejado.

1.2 Modo de manipulação dos blocos na área de trabalho

Observar a habilidade com que os alunos:

- b) arrastam os blocos da caixa de ferramentas para a área de trabalho;
- c) encaixam, desencaixam e movimentam os blocos na área de trabalho;
- d) fazem o "scrolling" da área de trabalho;
- e) apagam os blocos presentes na área de trabalho;
- f) copiam e colam blocos na área de trabalho;
- g) selecionam os parâmetros nos blocos

1.3 Utilização das funções auxiliares.

- a) Observar com que frequência os alunos utilizam a função zoom.
- b) Observar a habilidade com que os alunos salvam os programas.
- c) Observar a habilidade com a qual os alunos carregam programas salvos.

2 QUESTIONÁRIOS (Ambiente de Programação e Brincadeiras)

2.1 QUESTIONÁRIO 1 - Aplicado na primeira aula da oficina



Nome: _____

1	O que você achou da brincadeira de programar o robô humano?	 Ruim	 Mais ou menos	 Bom	 Legal	 Muito Legal
2	O que você achou dos vídeos e fotos sobre os Robôs?	 Ruim	 Mais ou menos	 Bom	 Legal	 Muito Legal
3	O que você achou da aula de hoje?	 Ruim	 Mais ou menos	 Bom	 Legal	 Muito Legal

Faça um X nas imagens onde aparecem robôs



2.2 QUESTIONÁRIO 2 - Aplicado na segunda aula da oficina



Nome: _____

1	O que você achou de programar a caixinha mágica?	 Ruim	 Mais ou menos	 Bom	 Legal	 Muito Legal
---	--	---	--	--	--	--

2	O que você achou da atividade de ver os círculos coloridos que o professor entregou mudando de cor na caixinha mágica?	 Ruim	 Mais ou menos	 Bom	 Legal	 Muito Legal
3	O que você achou da atividade de pintar os círculos coloridos e depois colocá-los na caixinha pra mudar de cor?	 Ruim	 Mais ou menos	 Bom	 Legal	 Muito Legal

2.3 QUESTIONÁRIO 3 - Aplicado na terceira aula da oficina



Nome: _____

1	Você achou fácil ou difícil encontrar os blocos que precisava para fazer seu programa?	 Muito Difícil	 Difícil	 Mais ou Menos	 Fácil	 Muito Fácil
2	Os ícones dos blocos ajudaram você a encontrar o bloco que você precisava?	 Não	 Mais ou Menos	 Sim		
3	Você achou fácil ou difícil encaixar e desencaixar os blocos ?	 Muito Difícil	 Difícil	 Mais ou Menos	 Fácil	 Muito Fácil
4	Você achou fácil ou difícil salvar e abrir um programa ?	 Muito Difícil	 Difícil	 Mais ou Menos	 Fácil	 Muito Fácil

1	O que você achou de programar a caixinha mágica ?	 Ruim	 Mais ou menos	 Bom	 Legal	 Muito Legal
2	O que você achou de programar a o robô gira-gira ?	 Ruim	 Mais ou menos	 Bom	 Legal	 Muito Legal

2.4 QUESTIONÁRIO 4 - Aplicado na quarta aula da oficina


DuinoBlocks
for KIDS

Nome: _____

1	Você gostou da brincadeira do robô humano com as maçãs e estrelas?	 Ruim	 Mais ou menos	 Bom	 Legal	 Muito Legal
2	A brincadeira das estrelas e maçãs ajudou você a entender como funciona o comando de programação REPETIR ?	 Não	 Mais ou Menos	 Sim		

3	O que você achou de montar o circuito com os LEDs?	 Ruim	 Mais ou menos	 Bom	 Legal	 Muito Legal
4	O que você achou de programar a caixinha mágica hoje?	 Ruim	 Mais ou menos	 Bom	 Legal	 Muito Legal

5	Você gostou da brincadeira da varinha mágica da aula passada?	 Não	 Mais ou Menos	 Sim		
---	---	--	--	--	--	--

2.5 QUESTIONÁRIO 5 - Aplicado na nona aula da oficina



Nome: _____

1	O que você está achando do DuinoBlocks4Kids?	 Ruim	 Mais ou menos	 Bom	 Legal	 Muito Legal
2	O DuinoBlocks4Kids é fácil ou difícil de usar?	 Difícil	 Um pouco Difícil	 Mais ou Menos	 Fácil	 Muito Fácil
3	Você está achando fácil ou difícil encontrar os blocos que precisa para fazer seu programa?	 Difícil	 Um pouco Difícil	 Mais ou Menos	 Fácil	 Muito Fácil
4	Os desenhos que têm nos blocos ajudam você a encontrar o bloco que você está procurando?	 Não	 Mais ou Menos	 Sim		
5	Você acha fácil ou difícil salvar um programa?	 Difícil	 Um pouco Difícil	 Mais ou Menos	 Fácil	 Muito Fácil
6	O que você está achando de programar a caixinha mágica?	 Ruim	 Mais ou menos	 Bom	 Legal	 Muito Legal
7	O que você achou da atividade de ver o vídeo e construir o programa para a caixinha funcionar igual ao vídeo?	 Ruim	 Mais ou menos	 Bom	 Legal	 Muito Legal

