



UNIVERSIDADE FEDERAL DO AMAZONAS - UFAM  
INSTITUTO DE COMPUTAÇÃO - ICOMP  
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA - PPGI

# UpRobotics: Robótica Educacional Utilizando Linguagem Visual Baseada em Blocos

Francisco Gabriel Teixeira Marinho

MANAUS-AM

2022



PODER EXECUTIVO  
MINISTÉRIO DA EDUCAÇÃO  
UNIVERSIDADE FEDERAL DO AMAZONAS  
INSTITUTO DE COMPUTAÇÃO  
PROGRAMA DE PÓS-GRADUAÇÃO EM  
INFORMÁTICA



UpRobotics: Robótica Educacional Utilizando Linguagem  
Visual Baseada em Blocos  
Francisco Gabriel Teixeira Marinho

Dissertação apresentada ao Programa de Pós-Graduação em Informática, da Universidade Federal do Amazonas, como parte dos requisitos necessários à obtenção do título de Mestre em Informática, na área de concentração em Sistemas Embarcados e Engenharia de Software.

**Orientador:** Raimundo da Silva Barreto, Dr.

MANAUS-AM

2022

## Ficha Catalográfica

Ficha catalográfica elaborada automaticamente de acordo com os dados fornecidos pelo(a) autor(a).

M338u Marinho, Francisco Gabriel Teixeira  
UpRobotics: robótica educacional utilizando linguagem visual baseada em blocos / Francisco Gabriel Teixeira Marinho . 2022  
197 f.: il. color; 31 cm.

Orientador: Raimundo da Silva Barreto  
Dissertação (Mestrado em Informática) - Universidade Federal do Amazonas.

1. Robótica Educacional. 2. Robótica Desplugada. 3. Pensamento Computacional. 4. Linguagem Visual Baseada em Blocos. 5. Geração Automática de Código. I. Barreto, Raimundo da Silva. II. Universidade Federal do Amazonas III. Título



PODER EXECUTIVO  
MINISTÉRIO DA EDUCAÇÃO  
INSTITUTO DE COMPUTAÇÃO

PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA



# FOLHA DE APROVAÇÃO

**"UpRobotics: Robótica Educacional Utilizando  
Linguagem Visual Baseada em Blocos"**

**FRANCISCO GABRIEL TEIXEIRA MARINHO**

Dissertação de Mestrado defendida e aprovada pela banca examinadora constituída pelos Professores:

Prof. Raimundo da Silva Barreto - PRESIDENTE

Prof. José Reginaldo Hughes Carvalho - MEMBRO INTERNO

Profa. Odette Mestrinho Passos - MEMBRO EXTERNO

Manaus, 04 de Março de 2022

*Ao meu saudoso tio Edmilson Natividade, minha saudosa tia Rosely Rodrigues e minha saudosa professora e amiga Odinéia Pamplona, três pessoas cujos trabalhos foram regados de amor, perseverança, esperança, fraternidade e humildade e são inspiração para muitas pessoas que amam fazer o bem ao próximo.*

# Agradecimentos

Agradeço primeiramente ao Deus de Abraão, Isaque e Israel, que também é o meu Deus, por ter me capacitado de todas as formas a iniciar, trilhar e concluir esta jornada, além de me permitir superar grandes adversidades ao longo da vida revigorando minhas forças com fé e esperança.

Agradeço meu pai Ricardo Cunha Marinho, uma grande autoridade da educação. Hoje ele é um pedagogo concursado em uma escola pública do estado do Amazonas, mas já foi comerciante, vigia noturno e por muito tempo trabalhou como vendedor de sacolas carregando uma caixa na garupa de sua bicicleta durante muitos dias de sol e chuva visando o sustento da família, mas mesmo diante dos desafios, passou no vestibular aos 36 anos e nos mostrou de forma prática que a educação é libertadora. Meu pai é uma grande fonte de inspiração. Agradeço por todo amor, carinho e dedicação dado por ele durante a minha sina acadêmica e de vida.

Agradeço minha mãe Zeliane da Silva Teixeira que heroicamente trabalhou dias e noites de plantão no hospital Padre Colombo do município de Parintins, na linha de frente contra o Covid-19. Apesar das tribulações, ela jamais deixou de ligar para demonstrar seu amor de mãe, sempre trazendo palavras de carinho, esperança, conforto e força. Ela é uma mulher poderosa e inspiradora, e que ilumina minha vida nos dias mais sombrios.

Agradeço a minha princesa Evelyn Vasconcelos por ser uma companheira incrível, parceira, amiga, amor da minha vida que sempre me apoiou de todas as formas durante essa trajetória. Agradeço também a toda sua família que sempre me incentivou e me acolheu com muito carinho.

Agradeço também minha maravilhosa irmã Ana Graziela, mulher inteligente, empreendedora, empoderada, alegre e que sempre me apoiou com palavras e atitudes revigorantes. Agradeço também ao meu irmão Davi Gustavo, o cara mais inteligente, perseverante e perspicaz que conheço, usou muito de sua proatividade para me ajudar e apoiar durante essa jornada.

Dedico também a meus tios (em ordem alfabética) Adanilson Gomes, Adriane Teixeira, Afonso Marinho, Alberto Marinho, Arimar (vulgo Xarope), Arlindo Souza, Augusto Cunha, Cízimo, Edmilson Natividade (in memorian), Fátima, Francisca Cunha, Kennedy, Marcelo, Marcos, Marlon, Moacir, Olga Marinho, Rosiane Almeida, Sandra Praia e Zelimar (in memorian). Cada um deles me proporcionou grandes momentos durante minha vida acadêmica, sempre com palavras de incentivo aos estudos.

Ao meus avós maternos, Paulo e Oscarina, por terem sempre me incentivado a estudar e apesar de eles não estarem mais presentes na minha vida, sempre os levo no meu coração. Dedico também aos meus avós paternos, Maria do Carmo e Francisco Marinho (meu eterno velho do rio), por serem os melhores avós do mundo.

Ao meu orientador, professor Raimundo Barreto por ser um orientador excepcional e valioso, sempre dedicado, conselheiro, dando o suporte necessário em todos os momentos durante todo esse período. Mesmo nos momentos psicologicamente difíceis, professor tem um dom extraordinário de acreditar no nosso trabalho e nos trazer palavras de incentivo que nos ajudam a revigorar nossas forças e não desistir dos nossos sonhos. Suas cobranças, críticas, sugestões e ensinamentos con-

tribuíram tanto para a minha formação acadêmica quanto para o meu crescimento pessoal e profissional.

Aos meus grandes amigos que tive a honra de conhecer no PPGi, Edwin Monteiro e Ronem Lavareda (por ordem alfabética). A parceria nos estudos, a constante troca de ideias e o incentivo mútuo nos estudos mostram o quanto eles nasceram para ser protagonistas nesse teatro social, pois são pessoas extraordinárias e excelentes amigos.

É importante frisar a importância da ajuda de meus colegas de laboratório de sistemas embarcados, tais como Anderson e Lucas sempre que precisei. Agradeço também a meus amigos Márcio Alencar e Gabriel Leitão, por suas incríveis palavras de incentivo e pela grande ajuda que proporcionaram em momentos difíceis durante essa pandemia, no que tange a se dedicar aos trabalhos de pesquisa do mestrado mesmo diante das adversidades.

E para fechar com chave de ouro, dedico esse trabalho também a uma das pessoas mais incríveis que Deus me proporcionou conhecer: professora Dra. Odette Mestrinho Passos. Minha mãe acadêmica, a primeira pessoa que me apresentou esse mundo extraordinário da pesquisa científica ainda na graduação. Suas palavras de incentivo e ajuda foram alguns dos fatores diferenciais que me permitiram hoje eu estar concluindo um Mestrado em Computação. Então, para mim, é uma grande honra dizer que já fui o seu aluno e orientando.

*"Aprender a ler significa entender o mundo,  
aprender a escrever significa mudá-lo"*

*(Paulo Freire)*

# Resumo

A tecnologia é um recurso fundamental que traz impacto em diversos setores da vida cotidiana. Dessa forma, há uma demanda crescente por profissionais com habilidades e competências na área de computação tanto no setor industrial quanto científico. No entanto, ainda há uma carência latente no que tange ao desenvolvimento do pensamento computacional no ensino básico, em particular, em sociedades com baixo nível socioeconômico. Existem diversas razões que vão muito além de recursos limitados ou falta de interesse por alguns alunos, por exemplo a dificuldade de entendimento e abstração de conhecimentos dessas tecnologias, tanto por parte de educadores, quanto de alunos. Desta forma, este trabalho apresenta um método para exercitar o pensamento computacional utilizando estratégias bem sucedidas no campo científico: a robótica educacional e a utilização de linguagens visuais baseadas em blocos gráficos arrastáveis. O UpRobotics é o artefato principal desse projeto de pesquisa, pois é uma ferramenta que busca auxiliar professores do público escolar infantil, capaz de explorar conhecimentos científicos tanto matemáticos quanto computacionais dos alunos, capacitando-os para manipular robôs educacionais, físicos ou virtuais, de acordo com a lógica de cada estudante. Deste modo, para compreender o estado da arte, obter análises de trabalhos relacionados ao tema, métodos, técnicas ou conhecer ferramentas disponíveis, foi utilizado um método baseado em

princípios de Engenharia de Software Experimental que se baseia na condução de estudos primários e secundários em fases diferentes da investigação. Assim sendo, através de experimentos realizados com alunos em uma escola utilizando conceitos de matemática básica aplicados ao conceito de robótica educacional e pensamento computacional, esse projeto trouxe contribuições na área de Robótica Educacional para o apoio do ensino-aprendizagem dos alunos, tudo isso aliado a ambientes de simulação 3D e linguagens visuais baseadas em blocos.

**Palavras-chave:** Robótica Educacional, Robótica Desplugada, Pensamento Computacional, Linguagem Visual Baseada em Blocos, Geração Automática de Código.

# Abstract

Technology is a fundamental resource that has an impact on many sectors of everyday life. Thus, there is a growing demand for professionals with skills and competences in the area of computing in the industrial sector in scientific terms. However, there is still a latent lack regarding the development of computational thinking in basic education, particularly in societies with low socioeconomic status. Several reasons that go beyond limited resources or lack of interest by some students, due to difficulty in understanding and abstracting knowledge of these technologies, both on the part of educators and students. In this way, it presents the computational training method used in a successful method of graphics in the scientific field: educational robotics and the use of visual languages in draggable blocks. UpRobotics is the main search engine of this research project, as it is a possibility that assists teachers with both mathematical and computational knowledge of children's schools, enabling the public to manipulate manipulators, real or virtual, according to the logic of each student. Thus, for the state of the art, comprehensible studies related to the topic, techniques or available techniques, a method based on software engineering principles was used, which is based on conducting primary and secondary studies different from the investigation. Therefore, experiments carried out with students in a school using concepts to support basic education applied to educational robotics teaching

and computational thinking, this project brought to the area of educational robotics contribution to the students' concept-learning, all this just 3D simulation and block visual languages.

**Keywords:** Educational Robotics, Unplugged Robotics, Computational Thinking, Visual Language Based on Blocks, Automatic Code Generation.

# Lista de Figuras

1.1	Metodologia de pesquisa adotada (adaptado de Spínola et al., 2008) . . . . .	25
1.2	Metodologia de pesquisa Fase de Concepção (Spínola et al., 2008) . . . . .	26
2.1	Braço Robótico Canadam2 na Estação Espacial Internacional. (Fonte: NASA) . . . . .	31
2.2	Scratch, ferramenta desenvolvida pelo MIT . . . . .	35
3.1	Seleção no 1 <sup>o</sup> Filtro . . . . .	44
3.2	Seleção no 2 <sup>o</sup> Filtro . . . . .	45
3.3	Panorama: critérios de aceitação na extração de dados . . . . .	45
3.4	Panorama: critérios de exclusão na extração de dados . . . . .	46
3.5	Artigos Seleccionados por ano . . . . .	50
3.6	Panorama geral sobre os métodos encontrados para responder a QP1 . . . . .	53
3.7	Panorama: Principais ferramentas encontradas na literatura . . . . .	55
4.1	A interface de usuário magnética tangível utilizada no trabalho de Miller et. al. (2018). . . . .	59
4.2	DuinoBlocks . . . . .	62
5.1	Modelo Espiral descrito por Pressman e Maxim (2016) . . . . .	66

---

5.2	Arquitetura do UpRobotics. . . . .	70
5.3	Diagrama de atividades da ferramenta . . . . .	72
5.4	Bloco importar motores . . . . .	74
5.5	Bloco ativar motores . . . . .	75
5.6	Bloco definir portas . . . . .	76
5.7	Bloco controle de movimentos . . . . .	77
5.8	Bloco inicializar braço . . . . .	77
5.9	Blocos abrir garra e fechar garra . . . . .	78
5.10	Blocos girar braço para direita ou para a esquerda . . . . .	78
5.11	Blocos mover altura do braço . . . . .	78
5.12	Bloco mover ângulo do braço . . . . .	79
6.1	Autômato Finito da Linguagem Baseada em Bloco. . . . .	82
6.2	Exemplo de programação em blocos com o intuito de fazer o braço robótico apontar um objeto a frente dele e logo em seguida voltar a posição anterior. . . . .	83
6.3	Página inicial da ferramenta criada. . . . .	84
6.4	Parte principal da ferramenta criada. . . . .	85
6.5	Exemplo de blocos interligados. . . . .	86
6.6	Código gerado a partir dos blocos da Figura 6.5 . . . . .	86
6.7	Tela principal do UpRobotics . . . . .	88
6.8	Selecionar robô no UpRobotics . . . . .	89
6.9	Tela home do UpRobotics . . . . .	90
6.10	Tela de tutoriais do UpRobotics . . . . .	91
6.11	Tela sobre o UpRobotics . . . . .	91

---

6.12	Circuito no TinkerCAD que representa um semáforo simples programável no UpRobotics . . . . .	93
6.13	Exemplo de algoritmo montado no UpRobotics para o semáforo . . . . .	94
6.18	Thymio II no cenário "Alvo" . . . . .	94
6.14	Exemplo de código gerado no UpRobotics para o semáforo . . . . .	95
6.19	Thymio II no cenário "Obstáculos" . . . . .	95
6.15	Braço robótico utilizado no experimento . . . . .	96
6.20	Exemplo de programação do ThymioII . . . . .	96
6.16	Circuito no TinkerCad que representa os servomotores que movimentam o braço robótico . . . . .	97
6.21	Exemplo do ThymioII na aba simulador do UpRobotics . . . . .	97
6.17	Exemplo de programação do braço no UpRobotics . . . . .	98
6.22	Pioneer 3D-X no cenário "Seguidor de Parede" . . . . .	99
6.23	Exemplo de programação do Pioneer 3D-X . . . . .	99
7.1	Braço robótico deixando um objeto em cima de uma pilha de livros. . . . .	102
7.2	Relação entre <i>Joysticks</i> controladores e movimentos do braço. . . . .	103
7.3	Máquinas com acesso ao UpRobotics preparadas para o experimento . . . . .	107
7.4	Aluno programando um semáforo . . . . .	108
7.5	Alunos observando uma explicação sobre o funcionamento de uma semáforo no UpRobotics . . . . .	109
7.6	Smileyometer . . . . .	110
7.7	Fácil ou Difícil . . . . .	111
7.8	Legal ou Ruim . . . . .	111
7.9	Sim ou Não . . . . .	111
7.10	Termo de Consentimento Livre e Esclarecido . . . . .	112

---

7.11	Questionário de Caracterização dos Participantes . . . . .	113
7.12	Faixa etária e nível escolar dos participantes . . . . .	114
7.13	Aluno respondendo o questionário . . . . .	114
C.1	QUESTIONÁRIO 1 – Sobre a aula onde o aluno programou um Semáforo usando conhecimentos de grandezas e medidas . . . . .	153
C.2	QUESTIONÁRIO 2 – Sobre a aula onde o aluno programou um Braço Robótico usando conhecimentos de Ângulos. . . . .	154
C.3	QUESTIONÁRIO 3 – Sobre a aula onde o aluno programou um robô carrinho Thymio II usando conhecimento de Ângulos. . . . .	155
C.4	QUESTIONÁRIO 4 - Sobre a ferramenta UpRobotics, num contexto geral (parte 1) . . . . .	156
C.5	QUESTIONÁRIO 4 - Sobre a ferramenta UpRobotics, num contexto geral (parte 2) . . . . .	157
E.1	Semáforo: definir portas no Arduino . . . . .	185
E.2	Semáforo: aguardar N segundos . . . . .	185
E.3	Semáforo: acender o LED vermelho . . . . .	185
E.4	Semáforo: apagar o LED vermelho . . . . .	185
E.5	Semáforo: acender o LED verde . . . . .	186
E.6	Semáforo: apagar o LED verde . . . . .	186
E.7	Semáforo: acender o LED amarelo . . . . .	186
E.8	Semáforo: apagar o LED amarelo . . . . .	186
E.9	Braço Robótico - Modo básico: Definir portas no arduino . . . . .	187
E.10	Braço Robótico - Modo básico: resetar movimento . . . . .	187
E.11	Braço Robótico - Modo básico: movimentar de uma vez só . . . . .	187
E.12	Braço Robótico - Modo básico: base . . . . .	188

---

E.13 Braço Robótico - Modo básico: altura . . . . .	188
E.14 Braço Robótico - Modo básico: ângulo/orientação . . . . .	188
E.15 Braço Robótico - Modo básico: garra . . . . .	188
E.16 Braço Robótico - Modo básico: pausa . . . . .	188
E.17 Braço Robótico - Modo avançado: inicializar braço robótico a partir de uma determinada posição . . . . .	189
E.18 Braço Robótico - Modo avançado: fechar a garra N graus . . . . .	189
E.19 Braço Robótico - Modo avançado: abrir a garra N graus . . . . .	189
E.20 Braço Robótico - Modo avançado: girar o braço para a esquerda N graus . . . . .	189
E.21 Braço Robótico - Modo avançado: girar o braço para a direita . . . . .	190
E.22 Braço Robótico - Modo avançado: mover a altura do braço N graus .	190
E.23 Braço Robótico - Modo avançado: mover o ângulo/orientação do braço N graus . . . . .	190
E.24 Braço Robótico - Modo avançado: importar todos os motores . . . . .	190
E.25 Braço Robótico - Modo avançado: ativar todos os motores . . . . .	191
E.26 Braço Robótico - Modo avançado: movimentos automáticos . . . . .	191
E.27 Braço Robótico - Modo avançado: definir portas no arduino . . . . .	192
E.28 Thymio II - Alvo: definir ângulos, para deslocar o carrinho para um determinado lugar . . . . .	192
E.29 Thymio II - Obstáculos: Inicializar robô . . . . .	193
E.30 Thymio II - Obstáculos: definir constantes . . . . .	193
E.31 Thymio II - Obstáculos: obter motores da roda esquerda e da roda direita . . . . .	193
E.32 Thymio II - Obstáculos: desabilitar módulo de controle . . . . .	193

---

E.33 Thymio II - Obstáculos: definir ângulos, para deslocar o carrinho para um determinado lugar . . . . .	194
E.34 Thymio II - Obstáculos: obter sensores de distância frontais . . . . .	194
E.35 Thymio II - Obstáculos: habilitar sensores de distância . . . . .	194
E.36 Thymio II - Obstáculos: definir velocidade ideal dos motores das rodas	194
E.37 Thymio II - Obstáculos: definir velocidade inicial dos motores das rodas	194
E.38 Thymio II - Obstáculos: curvas à direita se o sensor frontal for acionado	195
E.39 Pioneer 3D-X Seguidor de parede: Iniciar o robô . . . . .	195
E.40 Pioneer 3D-X Seguidor de parede: obter ponteiros para os motores das rodas do robô . . . . .	195
E.41 Pioneer 3D-X Seguidor de parede: Obtenha o ponteiro para o robô . . . . .	195
E.42 Pioneer 3D-X Seguidor de parede: definir a posição alvo para o infinito	196
E.43 Pioneer 3D-X Seguidor de parede: obter a ativar os sensores de distância	196
E.44 Pioneer 3D-X Seguidor de parede: avançar até estarmos a N cm da parede . . . . .	196
E.45 Pioneer 3D-X Seguidor de parede: gire no sentido horário até que a parede esteja a nossa esquerda . . . . .	196
E.46 Pioneer 3D-X Seguidor de parede: laço principal onde a lógica é definida	197
E.47 Pioneer 3D-X Seguidor de parede: pare quando o robô terminar . . . . .	197

# Lista de Tabelas

3.1	Expressão de busca utilizada para encontrar as publicações . . . . .	40
3.2	Fichamento para catalogação . . . . .	44
3.3	Publicações selecionadas após o 2º Filtro . . . . .	46
3.4	Autores dos trabalhos selecionados após o 2º filtro . . . . .	48
3.5	Informações coletadas para responder a QP1 . . . . .	51
3.6	Panorama geral das ferramentas encontradas para responder a QP2 .	54
4.1	Comparação da Proposta com os Trabalhos Correlatos . . . . .	62
6.1	Significado de cada estado do autômato . . . . .	83
6.2	Significado de cada símbolo do alfabeto . . . . .	84
7.1	Lista de passos utilizados no experimento . . . . .	104
7.2	Porcentagens de respostas relacionadas ao experimento com o semáforo	115
7.3	Porcentagem das respostas relacionados ao experimento com o braço robótico . . . . .	117
7.4	Porcentagem das respostas relacionados ao experimento com o Thymio II	119
7.5	Porcentagem das respostas relacionados à percepção dos alunos diante das funcionalidades do UpRobotics . . . . .	121

# Lista de Abreviaturas e Siglas

<b>AFD</b>	<i>Autômato Finito Determinístico</i>
<b>AFN</b>	<i>Autômato Finito Não-Determinístico</i>
<b>CD</b>	<i>Computação Desplugada</i>
<b>CP</b>	<i>Computação Plugada</i>
<b>CD</b>	<i>Computação Desplugada</i>
<b>IDE</b>	<i>Integrated Development Environment</i>
<b>LVBB</b>	<i>Linguagem Visual Baseada em Blocos</i>
<b>MVC</b>	<i>Model-View-Controller</i>
<b>NASA</b>	<i>National Aeronautics and Space Administration</i>
<b>PC</b>	<i>Pensamento Computacional</i>
<b>RD</b>	<i>Robótica Desplugada</i>
<b>RE</b>	<i>Robótica Educacional</i>
<b>RP</b>	<i>Robótica Plugada</i>
<b>UI</b>	<i>User Interface</i>
<b>UML</b>	<i>Unified Modeling Language</i>
<b>UX</b>	<i>User Experience</i>

# Lista de Símbolos

$\Sigma$	Alfabeto
$ \mathbf{X} $	X é conjunto de símbolos pertencentes a $\Sigma$
$L$	Linguagem pertencente a $\Sigma$
$p$	Palavra gerada por L
$\Rightarrow$	Implicação
$\leftarrow$	Atribuição
$\cup$	União de conjuntos
$\in$	Pertence
$\cdot$	Produto
$\forall$	Para todos
$ $	Tal que
$\leq$	Menor ou igual
$\geq$	Maior ou igual

# Sumário

<b>1</b>	<b>Introdução</b>	<b>19</b>
1.1	Contexto . . . . .	19
1.2	Definição do Problema . . . . .	21
1.3	Motivação . . . . .	22
1.4	Objetivos . . . . .	24
1.5	Metodologia de Pesquisa . . . . .	24
1.5.1	Fase de Concepção da Tecnologia . . . . .	25
1.5.2	Fase de Avaliação da Tecnologia . . . . .	27
1.6	Organização da dissertação . . . . .	28
<b>2</b>	<b>Fundamentação Teórica</b>	<b>29</b>
2.1	Pensamento Computacional . . . . .	29
2.2	A Robótica . . . . .	30
2.3	Robótica Educacional . . . . .	32
2.4	Abordagens Desplugadas . . . . .	33
2.4.1	Computação Desplugada . . . . .	33
2.4.2	Robótica Desplugada . . . . .	34
2.5	Algumas Ferramentas Educacionais que Utilizam Linguagens Visuais	35

---

2.6	Resumo do Capítulo . . . . .	36
<b>3</b>	<b>Mapeamento Sistemático</b>	<b>37</b>
3.1	Planejamento . . . . .	37
3.1.1	Objetivo . . . . .	37
3.1.2	Definição do Protocolo . . . . .	38
3.2	Critérios de Seleção Preliminar . . . . .	41
3.3	Condução . . . . .	42
3.3.1	Seleção de Estudos Primários . . . . .	42
3.3.2	Extração de Dados . . . . .	43
3.4	Resultados e Discussões . . . . .	49
3.5	Resumo do Capítulo . . . . .	56
<b>4</b>	<b>Trabalhos Correlatos</b>	<b>57</b>
4.1	Trabalhos relacionados a Atividades Desplugadas . . . . .	57
4.2	Trabalhos Relacionados a Linguagens Visuais e Robótica Educacional	58
4.3	Comparativo da Proposta com os Trabalhos Correlatos . . . . .	62
4.4	Resumo do Capítulo . . . . .	64
<b>5</b>	<b>Método Utilizado</b>	<b>65</b>
5.1	Comunicação . . . . .	67
5.2	Planejamento . . . . .	67
5.3	Modelagem . . . . .	69
5.4	Construção . . . . .	72
5.5	Emprego . . . . .	73
5.6	Geração Automática de Código . . . . .	74
5.6.1	Funções . . . . .	74

---

5.6.2	Arduino . . . . .	76
5.6.3	Controle . . . . .	77
5.6.4	Movimentos . . . . .	77
5.7	Resumo do Capítulo . . . . .	79
<b>6</b>	<b>A Ferramenta UpRobotics</b>	<b>81</b>
6.1	A Linguagem Visual . . . . .	81
6.2	A Ferramenta UpRobotics . . . . .	87
6.3	Escolha das Aplicações a Serem Adicionadas na Ferramenta . . . . .	89
6.4	Escolha dos Componentes Robóticos a Serem Manipulados pela Ferramenta . . . . .	92
6.4.1	O Semáforo . . . . .	92
6.4.2	O Braço Robótico (meArm) . . . . .	92
6.4.3	Robô Carrinho Thymio II . . . . .	93
6.4.4	Robô Carrinho Pioneer 3D-X . . . . .	98
6.5	Resumo do Capítulo . . . . .	100
<b>7</b>	<b>Experimentos e Resultados Obtidos</b>	<b>101</b>
7.1	O Estudo de Viabilidade . . . . .	101
7.1.1	O Nivelamento Teórico . . . . .	102
7.1.2	A Demonstração da Ferramenta . . . . .	103
7.1.3	O Experimento Prático . . . . .	104
7.2	Resultados e Discussões do Estudo de Viabilidade . . . . .	105
7.3	O Experimento Prático em Sala de Aula . . . . .	106
7.3.1	Coleta de dados e Avaliação . . . . .	109
7.4	Resultados e Discussões . . . . .	111

---

7.5	Resumo do Capítulo . . . . .	123
<b>8</b>	<b>Conclusão</b>	<b>125</b>
8.1	Considerações Finais . . . . .	125
8.2	Publicações . . . . .	126
8.3	Limitações . . . . .	127
8.4	Trabalhos Futuros . . . . .	128
	<b>Referências Bibliográficas</b>	<b>130</b>
<b>A</b>	<b>Plano de Ensino</b>	<b>138</b>
<b>B</b>	<b>Planos de Aula</b>	<b>141</b>
<b>C</b>	<b>Questionários de Avaliação</b>	<b>152</b>
<b>D</b>	<b>Avaliação de Usabilidade e Funcionalidade - Resultados mostrados em gráficos</b>	<b>158</b>
<b>E</b>	<b>UpRobotics: Blocos e seus códigos-fonte equivalentes</b>	<b>184</b>
E.1	Semáforo . . . . .	185
E.2	Braço Robótico - Modo básico . . . . .	187
E.3	Braço Robótico - Modo avançado . . . . .	189
E.4	Thymio II - Alvo . . . . .	192
E.5	Thymio II - Obstáculos . . . . .	193
E.6	Pioneer 3D-X - Seguidor de parede . . . . .	195

# Capítulo 1

## Introdução

Este capítulo apresenta a parte introdutória da dissertação, fazendo uma breve contextualização, mostrando os desafios, o que motivou a desenvolver este trabalho, bem como seus objetivos, além de mostrar a metodologia de pesquisa adotada. Essa pesquisa é voltada para a área de Robótica Educacional, envolvendo alguns temas como Pensamento Computacional, Linguagens Visuais Baseadas em Blocos e Robótica Desplugada.

### 1.1 Contexto

A tecnologia tem um forte impacto na força de trabalho, no desenvolvimento e na prosperidade socioeconômica. Deste modo, a computação torna-se uma prioridade educacional desde o ensino básico. É necessário equipar a força de trabalho do futuro por intermédio de conhecimentos e habilidades de ciência da computação, engenharias e robótica. As pessoas vivem na era digital e não apenas aprendem a falar, escrever ou desenvolver habilidades específicas. No que se refere às pessoas como

alunos, elas precisam ter sucesso em seu contexto e uma maneira possível de conseguir isso é desenvolvendo o pensamento computacional, isto é, um dos instrumentos mais importantes da educação moderna, pois é o processo de elaborar problemas e soluções de uma forma que um computador possa efetivamente entender e realizar. Nos últimos anos, existem várias iniciativas para promover o pensamento computacional e definir abordagens e métodos para apoiá-lo. Um deles é a Robótica educacional (RE), isto é, uso interligado de recursos tanto de software, quanto de hardware, com o intuito de estimular estudantes a planejar, projetar e criar soluções em diversos contextos educacionais, possibilitando a efetiva participação e a autonomia. A RE tem forte embasamento teórico na teoria construtivista de Piaget e Merlone (1976), que considera a manipulação de objetos um método chave para que a criança, por si só, seja capaz de construir o seu próprio conhecimento. Com isso, o ensino torna-se mais efetivo quando a criança se esforça, de forma consciente, na construção de algo que ela possa manipular. O objetivo da RE não é apenas permitir a manipulação de robôs, mas sim disponibilizar um ambiente que permita o aprendizado do aluno sobre conceitos lógicos, estimulando o processo de raciocínio e aprendizagem. Desta forma, esse método não se limita apenas à área de computação, pois ele pode ser aplicado para o ensino de diversas áreas de conhecimento técnicos ou básicos. Dentro da RE, encontra-se a Robótica Desplugada (RD), no qual os alunos desenvolvem habilidades de robótica sem usar robôs físicos.

A RE possui uma universalidade de aplicações em diversas áreas de ensino das ciências, em todos os graus de ensino, mas esse trabalho está focado no ensino básico. O grande problema é que a RE ainda não é muito utilizada em escolas onde há baixo nível socioeconômico.

## 1.2 Definição do Problema

Segundo Resende *et al.* (2013) as principais dificuldades percebidas no processo ensino-aprendizagem de ciências exatas estão relacionadas à dificuldade em relação à linguagem usual dos alunos e a linguagem matemática, o que dificulta a interpretação dos textos matemáticos e a proposição de questionamentos. A falta de métodos que minimizem essas barreiras pode afetar o estímulo do raciocínio lógico, trazendo desafios constantes à criatividade, tendo como consequência frustrações aos alunos.

Uma método para sanar esse problema é a utilização de atividades que exercitem o pensamento computacional através da RE (Queiroz *et al.* 2017). No entanto, é necessária a utilização da linguagem de programação para controlar robôs educacionais, mas uma linguagem textual é composta por algoritmos um tanto complexos, no qual um único sinal incorreto pode impedir a realização de certa funcionalidade e isso também pode trazer resultados frustrantes. Deste modo, utilizar RE requer treinamento a profissionais da educação, além da utilização de uma linguagem mais facilitadora tanto para os professores, quanto para os alunos.

Além do mais, Miller *et al.* (2018) afirmam que dispositivos robóticos não estão acessíveis a todas as escolas, principalmente as de baixo nível socioeconômicos, o que dificulta a utilização de RE nessas instituições. Isso é um problema que envolve questões discutidas na sociologia, como distinção entre classes mais ou menos privilegiadas. Deste modo, há necessidade de métodos ou ferramentas que promovam a inclusão de crianças de baixa renda ajudando-as a ter a mesma oportunidade de uma criança com maior nível socioeconômico, para aprender, desenvolver ou manipular as mesmas tecnologias.

### 1.3 Motivação

A construção de uma ferramenta que promova o pensamento computacional por intermédio de RE pode ser uma solução viável para minimizar futuramente ou até mesmo extinguir grande parte da problemática, promovendo a igualdade, minimizando a segregação e até mesmo a viabilizando a equidade entre alunos em diferentes níveis socioeconômicos. Assim sendo, as linguagens e a ferramenta podem gerar o desenvolvimento de habilidades cognitivas importantes a todos.

Trabalhos com contribuições sociais são quase sempre bem recebidos pela comunidade. Deste modo, este trabalho traz avanços que podem ser bastante relevantes à ciência no que tange a linguagens visuais e RE. Desta forma, tendo em vista que toda interação entre os alunos e a ferramenta para a produção dos movimentos de um robô educacional é um ato de comunicação que se dará por meio da linguagem visual baseada em blocos (LVBB), como toda linguagem, a deste trabalho deverá ser projetada levando em consideração elementos léxicos, sintáticos e semânticos, além da complexidade da solução proposta em transformar a linguagem de alto nível para baixo nível, com o intuito de validá-la.

Deste modo, construir uma ferramenta desse porte requer conhecimentos básicos com enfoque em Interação Humano-Computador tendo em vista bons resultados de Experiência do Usuário (User Experience) e Interface com o Usuário (User Interface); Robótica para saber como ela é aplicada na educação e Fundamentos Teóricos da Computação para formalizar a linguagem, para que a especificação da linguagem seja construída de forma satisfatória.

Essa solução se diferencia das outras (mostradas no Capítulo 3 e 4) porque é exclusivamente voltada para o ensino da RE e não de programação de computadores, além de possuir características que podem ser usadas tanto em Robótica Plugada,

quando em Robótica Desplugada. Mesmo sem o auxílio de um computador real, o aluno poderá programar movimentos de um robô apenas arrastando e conectando blocos, sem saber que literalmente ele está produzindo linhas de código-fonte em uma linguagem de programação robótica, e principalmente, sem se preocupar com a sintaxe ou com a intimidação de um cursor piscando na linha de comando. Se ele quiser testar em um robô real (físico), ele pode fazer isso por meio do código-fonte que é gerado automaticamente pela ferramenta.

No que tange ao campo científico, transformar automaticamente uma linguagem de alto nível para uma linguagem de baixo nível não é uma tarefa trivial na computação. Assim, são criados campos de estudo na engenharia de software e ciência da computação, com o intuito de promover a construção de código a partir de modelos de software. Com pré-requisito de quaisquer experimentação utilizando uma linguagem visual de programação criada e baseada em blocos interconectados arrastáveis para a geração automática de código fonte que possa ser embarcado em um dispositivo, é necessário embasamento teórico na literaturas científicas afim de demonstrar a viabilidade e eficiência desse método na área de robótica. Esse processo é uma das características que mostram o cuidado e o rigor científico desse trabalho perante a comunidade.

Esse projeto de pesquisa, em princípio, tem aplicabilidade no conhecimento de matemática básica nas escolas, além de alguns conceitos de robótica. No entanto, futuramente, não é descartada a possibilidade dele servir de apoio ao ensino-aprendizagem de outras ciências da escola básica para o ensino de algum assunto que envolva RE por metodologia de ensino-aprendizagem.

## 1.4 Objetivos

O objetivo principal deste trabalho é aplicar linguagens visuais baseadas em blocos no contexto de RE tendo em vista exercitar o pensamento computacional por meio de atividades plugadas e desplugadas.

Para que esse objetivo seja alcançado, faz-se necessário o cumprimento dos seguintes objetivos específicos:

- Construir linguagens visuais baseadas em blocos, de fácil aprendizagem, para manipular um ou mais robôs educacionais utilizando conceitos de matemática básica, lógica de programação e internet das coisas.
- Traduzir linguagens visuais em blocos gráficos arrastáveis interconectáveis (alto nível) para código-fonte utilizável (baixo nível) de forma automática.
- Verificar e validar a solução proposta, em termos de usabilidade e funcionalidade, em uma escola da região.

## 1.5 Metodologia de Pesquisa

O método de pesquisa adotado neste trabalho está fundamentado nos princípios da Engenharia de Software Experimental que se baseia na condução de estudos primários e secundários em fases diferentes da investigação (Spínola et al., 2008). Ela se divide em duas fases: concepção e avaliação da abordagem proposta, conforme descrito na Figura 1.1

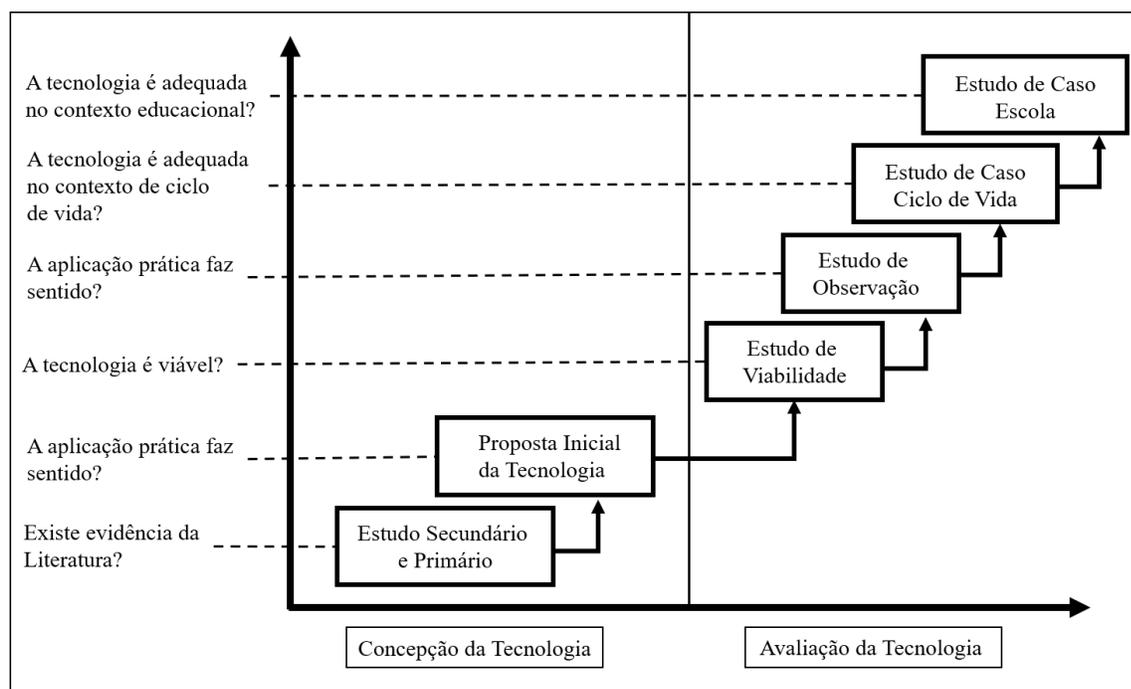


Figura 1.1: Metodologia de pesquisa adotada (adaptado de Spínola et al., 2008)

### 1.5.1 Fase de Concepção da Tecnologia

A fase de concepção da tecnologia envolve alguns passos e a execução de um estudo secundários com o objetivo de se obter uma proposta inicial da ferramenta proposta, como mostrado na Figura 1.2.

- **Revisão Inicial da Literatura (RIL):** a estratégia bibliográfica tem como objetivo fornecer subsídios teóricos acerca dos assuntos envolvidos e identificar os conceitos básicos para apoiar a definição de um protocolo do mapeamento sistemático da literatura mais preciso e abrangente. Esta etapa foi executada no período de março a junho de 2019 com o objetivo de realizar uma revisão ad hoc da literatura sobre Linguagens Visuais, Geração Automática de Códigos e RE com uma tecnologia de apoio ao exercício do pensamento computacional

nas escolas. Esta pesquisa incluiu a busca sobre linguagens e tecnologias que poderiam ser utilizadas do contexto desse trabalho trabalho.

- **Mapeamento Sistemático (MS):** fornece uma visão geral de uma área de pesquisa, identificando a quantidade, os tipos de pesquisas realizadas, os resultados disponíveis, além das frequências de publicações ao longo do tempo para identificar tendências (Petersen et al., 2008). O MS (estudo secundário) foi executado no período de julho a dezembro de 2019 e teve como objetivo compreender o estado da arte, analisando estratégias de ensino do pensamento computacional utilizando robótica no ensino básico, identificando metodologias, abordagens, técnicas ou ferramentas.
- **Corpo de Conhecimento:** corpo de conhecimento é caracterizado pelas metodologias e ferramentas, que utilizam as RE, no que diz respeito a linguagens visuais e pensamento computacional.

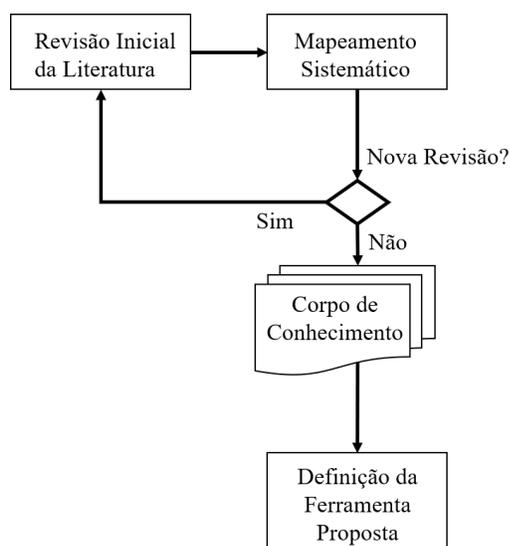


Figura 1.2: Metodologia de pesquisa Fase de Concepção (Spínola et al., 2008)

- **Definição da Ferramenta Proposta:** nesta etapa a definição da proposta da dissertação é consolidada através do corpo de conhecimento adquirido com o estudo secundário. A definição da proposta da dissertação foi executada no período de janeiro a setembro de 2020 e teve por objetivo a construção de uma ferramenta que possibilite a transformação de uma LVBB em código executável de modo a ser utilizada em um microcontrolador que possa movimentar robôs educacionais. A arquitetura da ferramenta e seus componentes constituintes são projetados de modo a embasar detalhes mais técnicos do seu desenvolvimento.

### 1.5.2 Fase de Avaliação da Tecnologia

Após da fase de concepção, foi elaborado um estudo de viabilidade, descrito de forma detalhada na Seção 7.1, cujos resultados serviram para algumas tomadas de decisão e melhorias a serem implementadas, o objetivo desse estudo foi avaliar sob o ponto de vista operacional, técnico, e organizacional se o projeto era viável. Em seguida, foram implementados melhorias e novas funcionalidades na ferramenta, e após experimentos práticos em sala, a etapa de avaliação da ferramenta foi realizada, através de uma pesquisa qualitativa que utiliza o pesquisador como instrumento de observação intensiva no ambiente que está sendo estudado, explorando sua interpretação dos fenômenos ocorridos (Godoy, 1995) e de uma pesquisa quantitativa que classifica e analisa dados obtidos através de opiniões e informações, utilizando técnicas estatísticas (Silva e Menezes, 2005). Para isso, foram realizadas entrevistas (pesquisa qualitativa) e aplicação de questionários (pesquisa quantitativa) em na escola, por meio do público alvo. O objetivo desta fase é avaliar a ferramenta, em um contexto real, de forma a verificar sua usabilidade e funcionalidade.

## 1.6 Organização da dissertação

No **Capítulo 1** é apresentada introdução deste trabalho através da contextualização, definição do problema, metodologia e os objetivos desta dissertação.

O **Capítulo 2** enfatiza a fundamentação teórica, explorando assuntos acerca de robótica, atividades desplugadas e algumas ferramentas baseadas em blocos, tal como o artefato dessa dissertação.

Os **Capítulos 3 e 4** descrevem, respectivamente, o mapeamento sistemático e a discussão dos trabalhos correlatos referentes ao objeto de pesquisa nesta dissertação. Ambos capítulos possibilitam uma visão geral das técnicas e ferramentas utilizadas para solucionar problemas similares aos apresentados durante a definição do problema.

O método utilizado é detalhado no **Capítulo 5** de tal forma que é possível observar, inicialmente, uma visão geral da arquitetura e, em um segundo momento, as particularidades do funcionamento individual e colaborativo dos elementos em função da tecnologia utilizada.

O **Capítulo 6** apresenta as características principais da ferramenta, tais como foi concebida uma linguagem e quais aprimoramentos e tecnologias foram adicionados na ferramenta ao longo do tempo.

O **Capítulo 7** apresenta os principais resultados obtidos e analisados ao longo de estudo de viabilidade e experimentos em ambiente escolar, além de mostrar como foi elaborada e aplicada as avaliações necessárias para obter resultados.

No **Capítulo 8** está a conclusão do trabalho onde estão as considerações finais abordando a cerca de cenários de aplicações, publicações, limitações do método utilizado e trabalhos futuros.

# Capítulo 2

## Fundamentação Teórica

Para melhor compreensão dos elementos que compõem este trabalho, faz-se necessário o entendimento prévio de alguns conceitos, e como eles corroboram para a realização do método utilizado. Neste capítulo é introduzido conceitos sobre Pensamento Computacional (Seção 2.1), a Robótica (Seção 2.2), Robótica Educacional (Seção 2.3), Abordagens Desplugadas (Seção 2.4), incluindo Computação Desplugada (Seção 2.4.1) e Robótica Desplugada (Seção 2.4.2) e as ferramentas utilizadas no ensino e aprendizagem de programação que utilizam linguagens visuais baseadas em blocos gráficos arrastáveis (Seção 2.5).

### 2.1 Pensamento Computacional

Wing (2006) afirma que o Pensamento Computacional (PC) é caracterizado pela capacidade de reformular problemas que, embora sejam difíceis, seja possível transformá-los em problemas mais simples de modo que possamos resolvê-los, prevenindo erros e estando preparado para corrigi-los, trabalhando de forma segura, sem necessitar

conhecer todas as complexidades existentes.

Para Barr e Stephenson (2011), o objetivo principal do PC é permitir que as pessoas resolvam problemas tentando racionar como os computadores pensam, nos mais variados tipos de problemas que enfrentem em todas as áreas, mas não necessariamente pensar como os cientistas da computação pensam.

Em linhas gerais, o Pensamento computacional (PC) é um dos instrumentos mais importantes da educação moderna, pois é o processo de elaborar problemas e soluções de uma forma que um computador possa efetivamente entender e realizar (Nakamura e Kawasaki 2019). Ele é requisitado por ter a capacidade de desenvolver habilidades cognitivas em todas as áreas da matriz curricular. Swaid e Suid (2019) afirmam que o PC é uma habilidade que toda criança do século XXI deve ter, nesse sentido a comunidade científica trabalha para desenvolver e expandir esse conhecimento para que todas as pessoas tenham a oportunidade de desenvolver suas habilidades cognitivas.

## 2.2 A Robótica

De acordo com Queiroz *et al.* (2017) e Kloc *et al.* (2009) um robô é "qualquer estrutura mecânica autômata multifuncional reprogramável operada por meio de circuitos integrados, controles integrados, controles eletro-hidráulicos e engrenagens, projetada para movimentar, de diversas formas, uma série de materiais ou dispositivos especializados". A robótica é o ramo da ciência que se concentra em construir essas estruturas e manipulá-las por meio de lógica de programação (Souza e Duarte 2015).

Segundo May (2017) a robótica é a ciência responsável pelo estudo dos robôs, sendo estes, máquinas que podem ser usadas para realizar trabalhos específicos. Alguns robôs podem trabalhar sozinhos, outros precisam ser controlados o tempo todo

por seres humanos para que o trabalho seja realizado. Deste modo, os robôs se tornaram instrumentos importantes para o avanço da ciência e humanidade em diversos cenários, por exemplo, no ramo aeroespacial, a Agência Nacional Norte Americana (NASA, *National Aeronautics and Space Administration*) utiliza robôs de diversas formas, uma delas é a utilização de braços robóticos acoplados em naves espaciais responsáveis por mover objetos muito pesados no espaço, por exemplo há o braço robótico da estação espacial internacional chamado Canadam2, mostrado na Figura 2.1, que é usado para atividades como liberar ou controlar satélites, agarrar telescópios, mover astronautas ou até mesmo construir estações espaciais. Algumas dessas naves com braço robótico viajam para outros planetas e exploram superfícies. Alguns braços são programados para executarem determinados movimentos, outros são controlados por um astronauta ou pessoas designadas na missão através de controladores semelhantes a *joysticks* usados em videogames.



Figura 2.1: Braço Robótico Canadam2 na Estação Espacial Internacional. (Fonte: NASA)

É importante frisar também a importância da robótica e dos braços robóticos nas áreas da ciência no que tange à computação aliada à medicina e no setor de fábrica, como a indústria automotiva.

## 2.3 Robótica Educacional

Segundo Sousa *et al.* (2018) a Robótica Educacional (RE) é o uso interligado de recursos tanto de software, quanto de hardware, com o intuito de estimular estudantes a planejar, projetar e criar soluções em diversos contextos educacionais, possibilitando a efetiva participação e a autonomia. Neto *et al.* (2018) afirmam que os principais instrumentos desse método são robôs, que permitem um certo estímulo ao aprendizado de alunos ao programarem e construírem sistemas robóticos. Castilho (2002) afirma que o objetivo desse método de ensino não é apenas permitir a manipulação de robôs, mas sim disponibilizar um ambiente que permita o aprendizado do aluno sobre conceitos lógicos, estimulando o processo de raciocínio e aprendizagem. Desta forma, esse método não se limita apenas à área de computação, pois ele pode ser aplicado para o ensino de diversas áreas de conhecimento técnicos ou básicos. Por exemplo, Santos (2016) afirma que a RE é um recurso com potencial para contribuir significativamente para o processo de aprendizagem de conceitos matemáticos, além disso apresenta perspectivas interessantes para outras áreas, porém que ainda não são muito exploradas. Sousa *et al.* (2018) afirmam que a RE revelou vários benefícios no contexto educacional, não apenas ajudando o ensino de disciplinas, mas também possibilitando o desenvolvimento de várias habilidades, como trabalho em equipe, resolução de problemas e criatividade. Isso corrobora com a afirmação de Kim e Lee (2017) quando explanam que nos últimos anos houve um interesse significativo no uso da robótica em diversas atividades educacionais. Fracasso *et al.* (2018) afirmam que o uso da robótica como ferramenta educacional busca o ensino por meio da integração de conhecimentos técnicos e propedêuticos. Nesse sentido, a RE possui uma universalidade de aplicações em diversas áreas de ensino, como a matemática, a biologia, química, geografia, física, artes, entre outras. Embora tenha como principais

instrumentos os robôs, como por exemplo o kit Lego utilizado no trabalho de Sousa *et al.* (2018), foram criadas algumas abordagens, técnicas e ferramentas para facilitar o uso desse método em diversas escolas, em vários níveis de ensino e classes socioeconômicas. Uma abordagem muito utilizada e conhecida pelos educadores envolve atividades desplugadas, isto é, que não utilizam robôs.

## 2.4 Abordagens Desplugadas

Segundo Conde *et al.* (2017) as abordagens desplugadas são um conjunto de métodos nos quais os alunos desenvolvem habilidades e pensamento computacional sem usar tecnologia. Essas abordagens são utilizadas para incentivar alunos do ensino fundamental, médio e superior a promoverem o seu aprendizado em computação, de maneira simples, prática e didática. Assim sendo, há dois termos bastante mencionados na literatura no que se refere as essas práticas de ensino: Computação Desplugada e Robótica Desplugada.

### 2.4.1 Computação Desplugada

Para Bell *et al.* (2009) a Computação Desplugada (CD) é um método de ensino de Ciência da Computação, cujo o principal objetivo é expor aos estudantes conceitos computacionais e a forma como os cientistas da computação pensam, sem a utilização de um computador.

Segundo Vieira *et al.* (2013) a essa técnica se caracteriza por ensinar computação de forma simples, sem se preocupar com detalhes técnicos, pois ela se concentra em atividades lúdicas (por exemplo, peças teatrais) para o ensino real de computação, onde qualquer pessoa pode participar, mesmo que não entenda muito da área,

tendo como benefícios o desenvolvimento de habilidades de comunicação, criatividade, elementos cognitivos, sem muita complexidade e principalmente, sem o uso de qualquer computador real.

Cunha e Nascimento (2018) afirmam que esse método de ensino já é comum em diversos países como Estados Unidos, Israel e Canadá, pois eles têm consenso em suas políticas educacionais que o pensamento computacional no ensino básico é primordial para o futuro do país.

### 2.4.2 Robótica Desplugada

A literatura nos mostra que a RE possibilita uma diversidade de ganhos pedagógicos, mas nem todas as escolas tem recursos para utilizá-la. Assim sendo, a Robótica Desplugada (RD) é considerada uma boa alternativa, pois tem um conceito semelhante à Computação Desplugada, pois é um método de ensino de conceitos e práticas de robótica em sem necessidade de um robô físico (Conde *et al.* 2017). Nessa estratégia, geralmente são utilizadas dinâmicas coletivas que estimulam o pensamento computacional, abstração de conhecimento, raciocínio lógico, na maioria das vezes com materiais recicláveis ou até mesmo o próprio corpo.

Há diversos trabalhos na literatura, como os trabalhos de Krishnamoorthy e Kapila (2016) e Miller *et al.* (2018), por exemplo, que apresentam pesquisas e estudos de caso que aplicam esse método em salas de aula. A maioria dos resultados são satisfatórios, pois essa prática além de não necessitar que alunos tenham uma alta curva de aprendizado em robótica e não necessita de altos custo monetários. Assim sendo, são bastante eficientes justamente por permitir que sejam trabalhados temas abstratos de forma concreta, o que estimulam muitos alunos à área da tecnologia.

## 2.5 Algumas Ferramentas Educacionais que Utilizam Linguagens Visuais

Segundo Grover e Basu (2017), programação em ambientes criados em blocos é um elemento chave para currículos introdutórios de ciência da computação no ensino infantil, fundamental e médio. Ambientes de programação baseados em blocos, como Scratch, Alice, Snap!, App Inventor são veículos populares para programação introdutória. A Figura 2.2 mostra o exemplo da interface principal do Scratch, que utiliza a LVBB para gerar movimentos em objetos escolhidos pelo usuário.

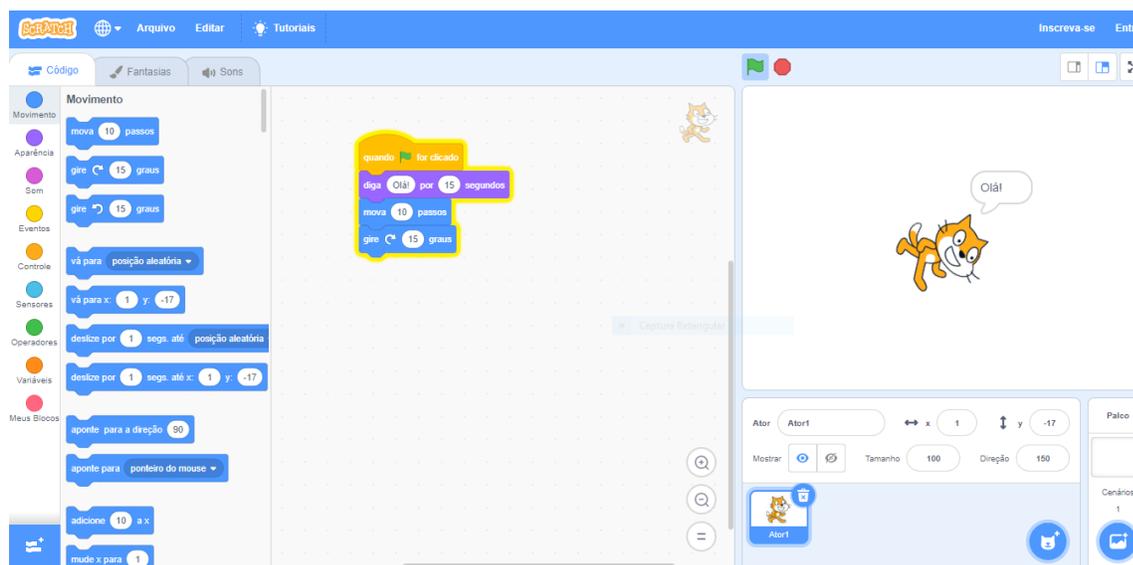


Figura 2.2: Scratch, ferramenta desenvolvida pelo MIT

Segundo Weintrop e Wilensky (2019), as linguagens de programação personalizadas em blocos estão cada vez mais comuns em salas de aula de ciências. Essas ferramentas que utilizam recursos de arrastar e soltar pequenos blocos gráficos facilitam o ensino e o trabalho de educadores porque não precisam que o aluno tenha uma alta curva de aprendizado em computação ou robótica. Essas ferramentas ba-

seadas em blocos são amplamente utilizadas em métodos de computação desplugada e robótica desplugada. Em RE, por exemplo, além do kit Lego, a ferramenta Lab-View é usada para programar pequenos robôs físicos montados. Por serem fáceis de utilizar, elas podem ser aplicadas em outras áreas como, por exemplo, na área musical. O trabalho de Baratè *et al.* (2017) obteve resultados satisfatórios quando promoveu o pensamento computacional de alunos do ensino básico utilizando uma ferramenta baseada em blocos arrastáveis para o ensino de música. O experimento consistiu nos alunos criarem música arrastando blocos que produziam notas musicais. Assim sendo, é possível observar que LVBB é aplicável em diversas áreas da ciência. Portanto, essas ferramentas são utilizadas com o intuito de facilitar o ensino e aprendizagem de diversas disciplinas, muito além de computação e robótica. Elas se popularizaram devido a facilidade de sua utilização, um dos pilares bastante prezados na área de Engenharia de Software.

## 2.6 Resumo do Capítulo

Neste Capítulo foram descritos todos os conceitos e definições que fundamentam o desenvolvimento da solução apresentada nessa dissertação. Foram apresentadas as informações que a literatura diz a respeito de pensamento computacional, características da RE e abordagens desplugadas, além de apresentar alguns exemplos de ferramentas que utilizam linguagens visuais, o que é de suma importância para o respaldo técnico desse trabalho.

# Capítulo 3

## Mapeamento Sistemático

Como base para definição de tecnologias ou abordagens do método utilizado (Capítulo 5), realizou-se um mapeamento sistemático cujo objeto da pesquisa era ter uma visão geral sobre os tipos de metodologias utilizadas para promover o Pensamento Computacional (PC), quais técnicas ou ferramentas utilizam robótica educacional e linguagens visuais.

### 3.1 Planejamento

Este planejamento foi elaborado conforme especificado em Biolchini *et al.* (2005), Mafra e Travassos (2006), e Kitchenham (2004).

#### 3.1.1 Objetivo

O objetivo deste estudo será esquematizado a partir do paradigma GQM (goal, question, and metric) descrito em Basili *et al.* (2002):

Analisar	Estratégias que envolvam PC e robótica no ensino básico
Com o propósito de	Identificar metodologias, abordagens, técnicas e ferramentas
No que diz respeito a	Linguagens visuais e atividades plugadas e desplugadas
Do ponto de vista do	Pesquisador
No contexto	Educacional

### 3.1.2 Definição do Protocolo

#### Questões de Pesquisa

**Questão Principal:** Quais são as principais estratégias e ferramentas que utilizam conceitos de computação ou robótica aplicados ao contexto educacional?

**Questão 1:** Quais as principais técnicas, métodos e abordagens presentes na literatura que utilizam Linguagens Visuais para promover o Pensamento Computacional no Ensino Básico?

**Questão 2:** Quais as principais ferramentas encontradas na literatura que utilizam linguagens visuais para o ensino da computação e/ou robótica educacional?

#### Critérios de Seleção de Fontes

Para as fontes de pesquisa, foram analisados os seguintes critérios:

- Artigos catalogados na biblioteca digital SCOPUS.
- Disponibilidade de consulta na Web.
- Presença de mecanismo de busca por meio de palavras-chaves e, principalmente, que viabilizem a utilização de string de busca.
- Artigos escritos em Língua Inglesa.

## Estratégias de Busca

As fontes serão acessadas via web, através de máquinas de busca que:

- Permitam busca no texto completo ou em campos específicos das publicações;
- Permita inserção manual desde que seja avaliada pelos critérios de inclusão;

## Fontes de Pesquisa

Para este trabalho, a biblioteca SCOPUS ([www.scopus.com](http://www.scopus.com)) será utilizada, devido a sua grande cobertura sob obras científicas publicadas em diversas conferências, revistas e periódicos de sociedades importantes da computação, como por exemplo, a ACM e a IEEE.

## String de Busca

As strings de busca foram definidas a partir das questões de pesquisa e do padrão PICO (population, intervention, comparison, outcomes) conceituado por Kitchenham e Charters (2007), conforme a Tabela 3.1.

A língua universal para a escrita científica é o inglês, além do mais, o estado da arte da área de computação está diretamente relacionada com trabalhos e pesquisas a nível internacional.

## String de busca no padrão da SCOPUS:

Esta String de busca retornou 173 publicações na Scopus.

```
TITLE-ABS-KEY ( ( "educational robotic*"OR "robotics teaching"OR "robotics learning"OR "robotics activit*"OR "educational robotics activit*"OR "computational thinking"OR "computational learn"OR "computational learning"OR "ele-
```

Tabela 3.1: Expressão de busca utilizada para encontrar as publicações

<b>População:</b>	robótica educacional ou computação educacional no ensino básico (e sinônimos) - <b>Palavras-chaves:</b> (“educational robotic*” OR “robotics teaching” OR “robotics learning” OR “robotics activit*” OR “educational robotics activit*” OR “computational thinking” OR “computational learn*” OR “elementary school” OR “elementary classroom” OR “basic education” OR “K-12” OR “K-12 students” )
<b>Intervenção:</b>	publicações que fazem referência a linguagens visuais + computação ou robótica desplugada (e sinônimos) - <b>Palavras-Chave:</b> (“visual language*” OR “visual language modeling” OR “block-based” OR “based on a block” OR “block-based coding” OR “domain-specific modeling language*” OR “domain-specific language*” OR “computer science unplugged” OR “computing disconnected” OR “unplugged robotic*” OR “unplugged schedule” OR “unplugged activit*” OR “unplugged method*”)
<b>Resultados:</b>	metodologias utilizadas (e sinônimos): - <b>Palavras-Chave:</b> “technique*” OR “approach*” OR “method*” OR “process*” OR “strateg*” OR “practice*” OR “platform*” OR “tool*”

mentary school"OR "elementary classroom"OR "basic education"OR "K-12"OR "K-12 students") AND ( "visual language\*"OR "visual language modeling"OR "block-based"OR "based on a block"OR "block-based coding"OR "domain-specific modeling language\*"OR "domain-specific language\*"OR "computer science unplugged"OR "computing disconnected"OR "unplugged robotic\*"OR "unplugged schedule"OR "unplugged activit\*"OR "unplugged method\*") AND ( "technique\*"OR "approach\*"OR "method\*"OR "process\*"OR "strateg\*"OR "practice\*"OR "platform\*"OR "tool\*") )

## 3.2 Critérios de Seleção Preliminar

### Critérios de Inclusão (CI):

- **CI1:** Utiliza Computação Educacional no ensino básico.
- **CI2:** Utiliza Robótica Educacional no ensino básico.
- **CI3:** Utiliza ferramenta baseada em linguagens visuais auxiliar a Computação Educacional no ensino básico.
- **CI4:** Utiliza ferramenta baseada em linguagens visuais para auxiliar a Robótica Educacional no ensino básico.
- **CI5:** Utiliza linguagem visual para o ensino da Computação Educacional e/ou Robótica Educacional no ensino básico.
- **CI6:** Utiliza linguagem visual apenas para Robótica Educacional. critérios de inclusão.
- **CI7:** Utilizam linguagem visuais que são transformadas em código-fonte aplicável exclusivamente à robótica educacional.
- **CI8:** Apresentam discussões a respeito de computação/robótica desplugada ou linguagens visuais para o ensino de computação/robótica no contexto educacional.

### Critérios de Exclusão (CE):

- **CE1:** Não conter informações úteis que contribuam de alguma maneira para que o objetivo do trabalho seja atingido.

- **CE2:** Publicação não disponível para a leitura.
- **CE3:** Trabalho não aplicável a nenhum critério de inclusão.
- **CE4:** Contém informações muito semelhantes a de outras publicações analisadas.
- **CE5:** Artigo contém resultados publicados há mais de 3 anos.

### Critérios de Qualidade

Para os artigos selecionados para extração de dados serão considerados os números de citações e o seu fator de impacto, caso estejam disponíveis.

## 3.3 Condução

### 3.3.1 Seleção de Estudos Primários

1. A seleção preliminar das publicações foi feita a partir da aplicação da expressão de busca às fontes selecionadas. Cada publicação foi catalogada em um repositório para análise posterior.
2. Cada artigo deverá ser analisado conforme o critério de inclusão e exclusão para poder ser selecionado ou rejeitado. Em princípio, os campos analisados no artigo serão: título, resumo e palavras-chave.
3. Os artigos incluídos ou excluídos são documentados no “Formulário de Seleção de Estudo”.

### Avaliação da Qualidade de Estudos Primários

A string de busca aplicada na Scopus retornou 173 publicações, das quais 99 foram aceitas no Filtro 1, com base nos critérios de inclusão e exclusão. O ano de publicação e a fonte foram primordiais nesse primeiro filtro.

### 3.3.2 Extração de Dados

A seleção preliminar nas publicações foi feita a partir da expressão de busca na SCOPUS, mas isso não garante que todo o material seja útil no contexto da pesquisa. Para o **Filtro 01**, foram analisados e considerados os seguintes itens: Título, Resumo e Palavras-chave.

No Filtro 01, apresentado na Figura 3.1, foram aplicados os critérios de seleção com base no contexto desses itens analisados nessa fase. Em seguida, para o **Filtro 02**, apresentado na Figura 3.2 foram verificados se de fato as publicações identificadas contextualizam o objetivo da pesquisa. As publicações selecionadas nesse segundo filtro foram catalogadas em fichamentos com determinados campos de coleta de dados, conforme mostra a Tabela 3.2.

Tabela 3.2: Fichamento para catalogação

<b>ID</b>	identificador do artigo
<b>Ano:</b>	ano de publicação do artigo
<b>Citações:</b>	quantidade de citações do artigo
<b>Fator de Impacto:</b>	fator de impacto de acordo com o JCR
<b>Título:</b>	título do artigo
<b>Autor(es):</b>	nome(s) do(s) autor(es)
<b>Palavras-chave:</b>	palavras chaves da publicação
<b>Fonte:</b>	local de publicação (conferência; journal)
<b>Resumo:</b>	breve descrição do estudo
<b>Objetivo:</b>	objetivo alcançado ou contribuições do estudo
<b>Método(s):</b>	abordagens, técnicas ou práticas
<b>Ferramenta(s):</b>	ferramentas ou plataformas utilizadas
<b>Comentários</b>	comentários a respeito desse estudo
<b>Trabalhos futuros:</b>	questão de pesquisa sugerida para trabalhos futuros

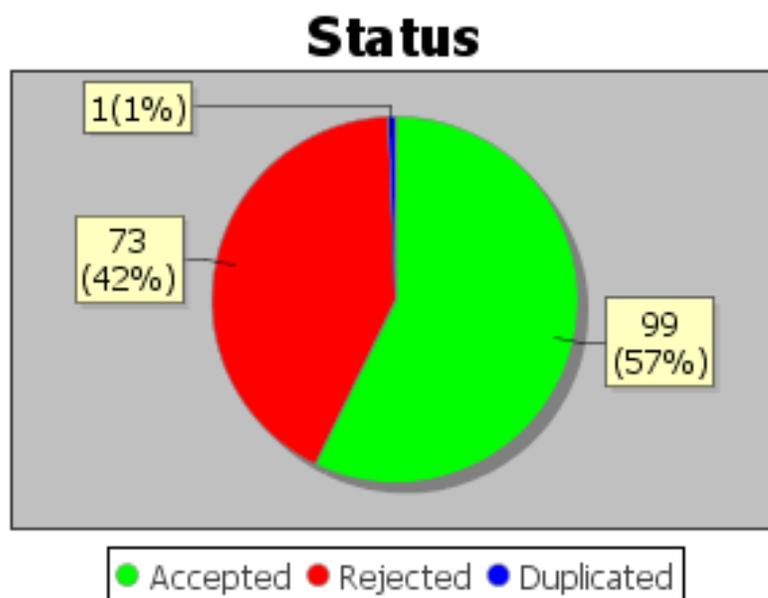


Figura 3.1: Seleção no 1º Filtro

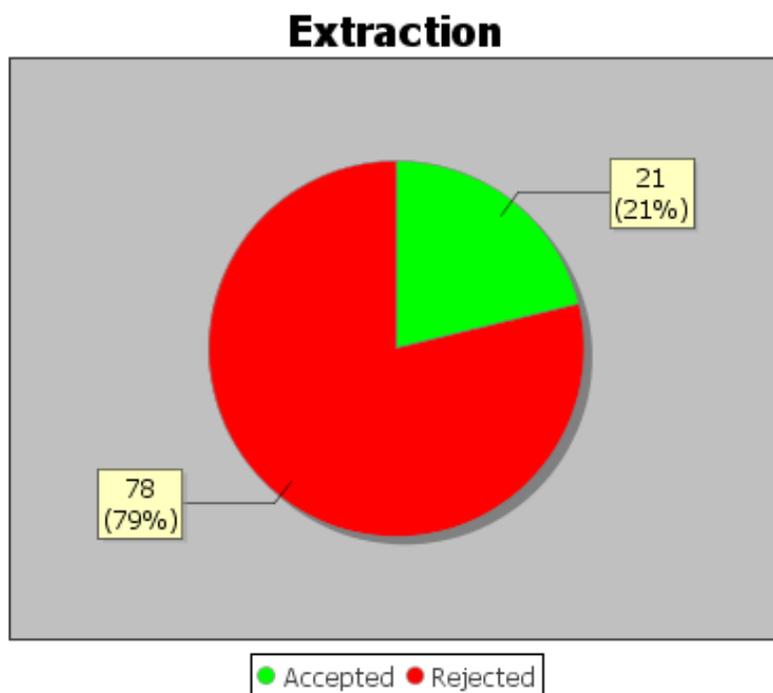


Figura 3.2: Seleção no 2º Filtro

As Figuras 3.3 e 3.4 mostram uma visão geral da inclusão e exclusão dos trabalhos durante o decorrer do Filtro 02. A Tabela 3.3 mostra as publicações selecionadas e fichadas após o Filtro 02 e a Tabela 3.4 mostra seus respectivos autores:

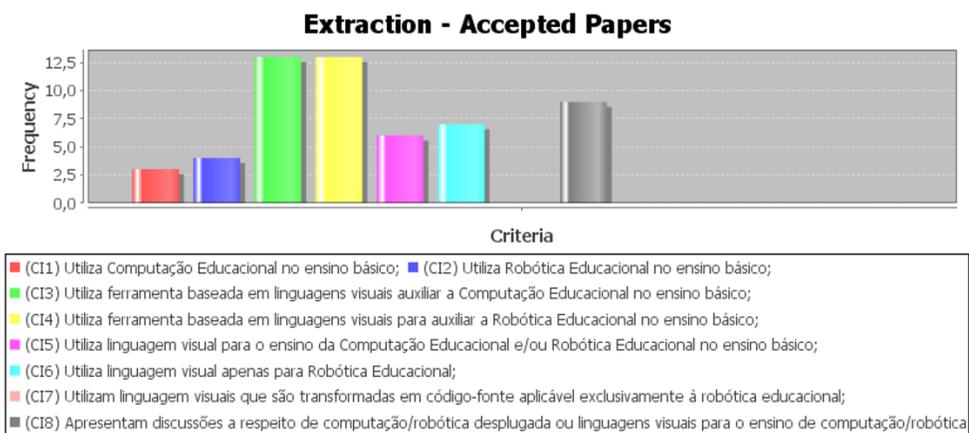


Figura 3.3: Panorama: critérios de aceitação na extração de dados

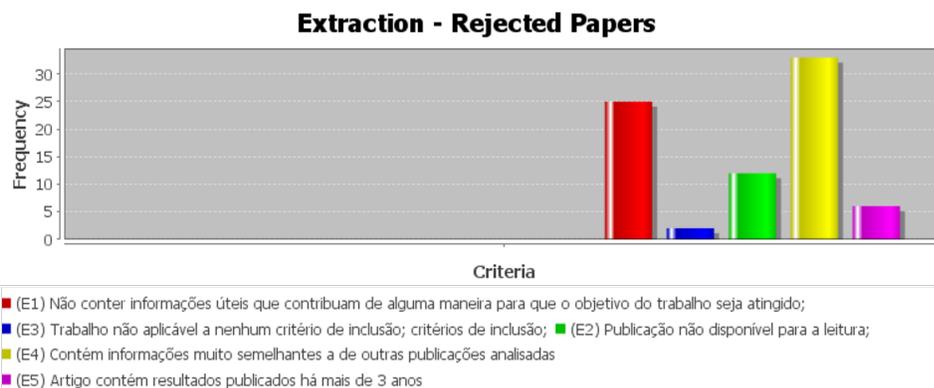


Figura 3.4: Panorama: critérios de exclusão na extração de dados

Tabela 3.3: Publicações selecionadas após o 2º Filtro

ID	Título	Fonte	Ano	Categoria
[P01]	Transitioning from introductory block-based and text-based environments to professional programming languages in high school computer science classrooms	Computers and Education	2019	Journal
[P02]	A cross-analysis of block-based and visual programming apps with computer science student-teachers	Journal Education Sciences	2019	Journal
[P03]	Exploring Different Unplugged Game-like Activities for Teaching Computational Thinking	Proceedings - Frontiers in Education Conference, FIE 2018-October, 8659077	2018	Conferência
[P04]	A systematic review on the use of LEGO robotics in education	Proceedings - Frontiers in Education Conference, FIE 2018-October, 8658751	2019	Conferência
[P05]	Computational thinking education through creative unplugged activities	Olympiads in Informatics 13, pp. 171-192	2019	Conferência

Tabela 3.3 Publicações selecionadas após o 2º Filtro

ID	Título	Fonte	Ano	Categoria
[P06]	Easy prototyping of multimedia interactive educational tools for language learning based on block programming	Conferência CSEDU 2019 - Proceedings of the 11th International Conference on Computer Supported Education2, pp. 140-153	2019	Conferência
[P07]	CT-Blocks Analyser: Analysing CT-Blocks Projects	International Conference on Computing, Communication and Networking Technologies, ICCCNT 2018 8493970	2018	Conferência
[P08]	Map-Blocks: Playing with Online Data and Infuse to Think in a Computational Way	Conferência 9th International Conference on Computing, Communication and Networking Technologies, ICCCNT 2018	2018	Conferência
[P09]	Digital Storytelling Using Scratch: Engaging Children Towards Digital Storytelling	Conferência 9th International Conference on Computing, Communication and Networking Technologies, ICCCNT 2018	2018	Conferência
[P10]	Teaching and learning computer science for primary school teachers: An argentine experience	Latin American Conference on Learning Technologies, LACLO 20188783523, pp. 349-355	2018	Conferência
[P11]	Starting from scratch: Outcomes of early computer science learning experiences implications for what comes next	ICER 2018 - Proceedings of the 2018 ACM Conference on International Computing Education Research pp. 142-150	2018	Conferência
[P12]	A review of intuitive robot programming environments for educational purposes	Journal Adaptive Behavior27(3), pp. 183-196	2018	Journal
[P13]	Pirate plunder: Game-based computational thinking using scratch blocks	Conference on Human Factors in Computing Systems - Proceedings 3312871	2018	Conferência
[P14]	Programming competitions in high school classrooms: RITA en RED	2017 43rd Latin American Computer Conference, CLEI 2017	2017	Conferência

Tabela 3.3 Publicações selecionadas após o 2º Filtro

ID	Título	Fonte	Ano	Categoria
[P15]	Promoting Computational Thinking in K-12 students by applying unplugged methods and robotics	Conferência TEEM 2017, October 2017, Cádiz, Spain	2017	Conferência
[P16]	A study of educational method using app inventor for elementary computing education	Journal of Theoretical and Applied Information Technology	2017	Journal
[P17]	A blocks-based visual environment to teach robot-programming to K-12 students	Computers in Education Journal 8(2), pp. 24-32	2017	Journal
[P18]	Measuring student learning in introductory block-based programming: Examining misconceptions of loops, variables, and Boolean logic	Proceedings of the Conference on Integrating Technology into Computer Science Education, ITiCSE pp. 267-272	2017	Conferência
[P19]	Visual computational thinking using Patch	Conferência 16th International Conference on Web-based learning, ICWL 2017	2017	Conferência
[P20]	Improving teaching and learning computer programming in schools through educational software	International Journal on Interactive Design and Manufacturing 13(4), pp. 1401-1422	2019	Journal
[P21]	Using a visual programming environment and custom robots to learn C programming and K-12 STEM concepts	ACM International Conference Proceeding Series p. 41-48	2016	Conferência

Tabela 3.4: Autores dos trabalhos selecionados após o 2º filtro

ID	Autores
[P01]	Weintrop e Wilensky (2019)
[P02]	João <i>et al.</i> (2019)
[P03]	Jagušt <i>et al.</i> (2018)

**Tabela3.4 Autores dos trabalhos selecionados após o 2º filtro**

ID	Autores
[P04]	Souza <i>et al.</i> (2018)
[P05]	Weigend <i>et al.</i> (2019)
[P06]	Federici <i>et al.</i> (2019)
[P07]	Vinayakumar <i>et al.</i> (2018a)
[P08]	Vinayakumar <i>et al.</i> (2018c)
[P09]	Vinayakumar <i>et al.</i> (2018b)
[P10]	Casali <i>et al.</i> (2018)
[P11]	Weintrop <i>et al.</i> (2018)
[P12]	Bravo <i>et al.</i> (2017)
[P13]	Rose <i>et al.</i> (2018)
[P14]	Rosales <i>et al.</i> (2017)
[P15]	Conde <i>et al.</i> (2017)
[P16]	Kim e Lee (2017)
[P17]	Goud Yadagiri <i>et al.</i> (2017)
[P18]	Grover e Basu (2017)
[P19]	Baratè <i>et al.</i> (2017)
[P20]	Jancheski (2017)
[P21]	Krishnamoorthy e Kapila (2016)

### 3.4 Resultados e Discussões

Com relação a última etapa do mapeamento sistemático foram relatados os resultados e discussões conforme as duas questões de pesquisas definidas.

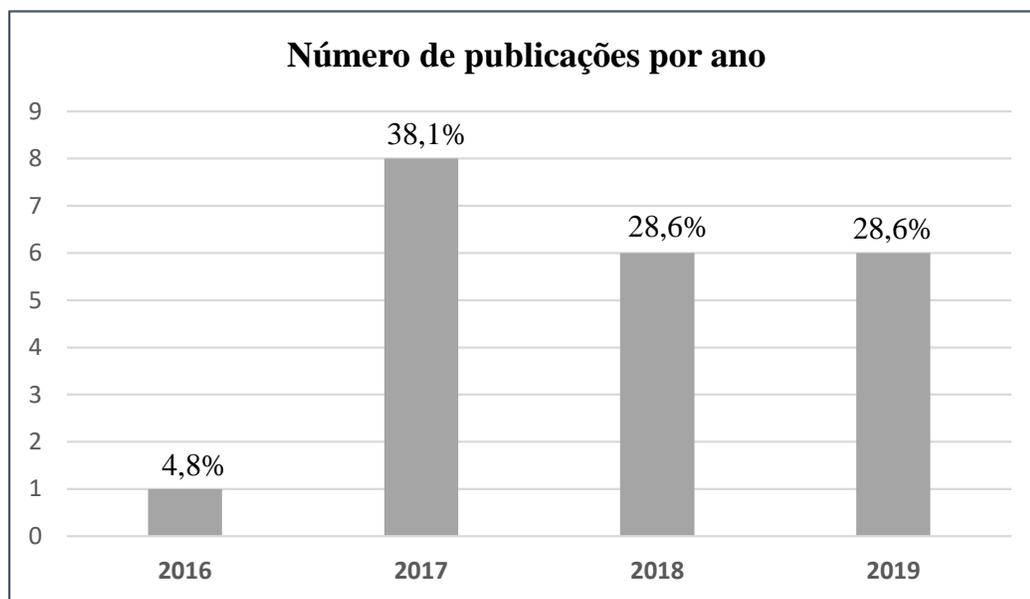


Figura 3.5: Artigos Selecionados por ano

### Análise e Discussão da QP1

A Figura 3.5 mostra o número de publicações selecionados por ano. Com relação à primeira questão de pesquisa “Quais as principais técnicas, métodos e abordagens presentes na literatura que utilizam Linguagens Visuais para promover o Pensamento Computacional no Ensino Básico?”, as respostas foram divididas em duas categorias: Computação Educacional e Robótica Educacional, sendo estes as principais estratégias de ensino que possuem as seguintes técnicas ou características descritas conforme observado na Tabela 3.5 e na Figura 3.6:

- **CP:** Computação Plugada (Connected Computing);

- **RP**: Robótica Plugada (Connected Robotics);
- **CD**: Computação Desplugada (Unplugged Computing);
- **RD**: Robótica Desplugada (Unplugged Robotics);
- **LVBB**: Linguagens Visuais Baseadas em Blocos (Block-Based Visual Languages);
- **GAME**: Jogos educacionais (Gamification);

Tabela 3.5: Informações coletadas para responder a QP1

ID	Métodos Utilizados	Computação Educacional	Robótica Educacional
[P01]	CD; LVBB	X	
[P02]	CP; LVBB	X	
[P03]	CD; GAME; LVBB	X	
[P04]	RD; RC; LVBB		X
[P05]	CD; GAME; LVBB	X	
[P06]	CD; LVBB	X	
[P07]	CD; CP; LVBB	X	
[P08]	CD; LVBB	X	
[P09]	CD; CP; LVBB	X	
[P10]	CP; CD; LVBB	X	
[P11]	RD; CD; LVBB		X
[P12]	RD; RC; LVBB	X	X
[P13]	CP; CD; GAME; LVBB	X	

**Tabela 3.5** Informações coletadas para responder a QP1

ID	Métodos Utilizados	Computação Educacional	Robótica Educacional
[P14]	RC; RD; LVBB		X
[P15]	RD;LVBB		X
[P16]	CP; LVBB	X	
[P17]	RP; LVBB		X
[P18]	CP; LVBB	X	
[P19]	CD; CP; LVBB	X	
[P20]	CD; CP; LVBB	X	
[P21]	RC; RD; LVBB		X

A grande maioria das publicações focaram em metodologias e ferramentas para o ensino de computação educacional, e isso equivale a 76,1%. Apenas 23,9% focaram no ensino da robótica, isso demonstra que ainda há carência de metodologias nessa área.

É importante frisar que 100% dos trabalhos utilizaram alguma linguagem baseada em blocos arrastáveis definidos de acordo com cada problema. Para fundamentar e justificar isso, muitos dos autores utilizaram pesquisas que afirmam que esse tipo de linguagem não requer que os alunos tenham alta curva de aprendizagem para aprender coisas novas por meio delas. É importante frisar que 19% também utilizam linguagens textuais e 28,6% utilizam gameificação. No que se referem a Computação Desplugada e Robótica Desplugada, 28,6% utilizam Robótica Desplugada, 52,4% utilizaram Computação Desplugada.

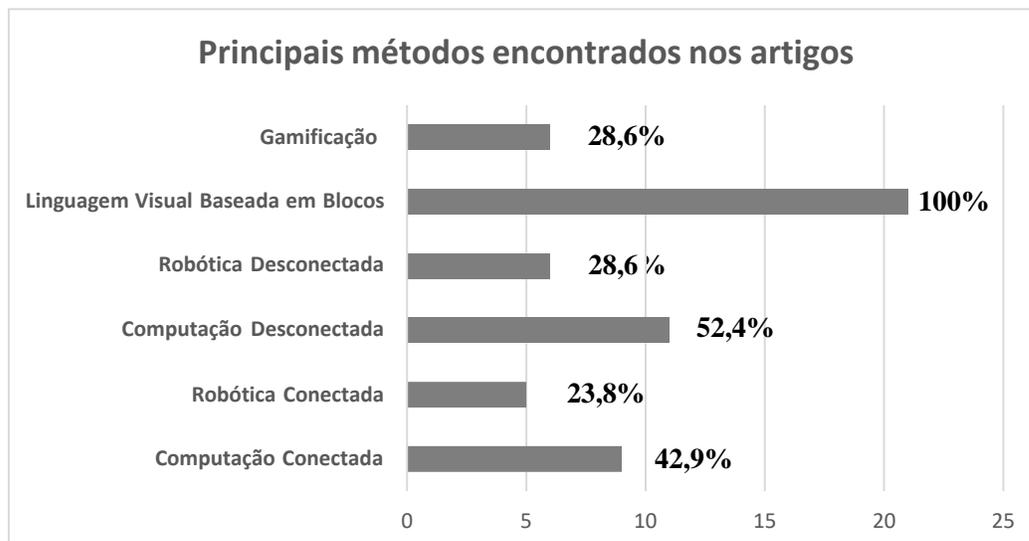


Figura 3.6: Panorama geral sobre os métodos encontrados para responder a QP1

## Análise e Discussão da QP2

Sobre a segunda questão “Quais as principais ferramentas encontradas na literatura que utilizam linguagens visuais para o ensino da computação e/ou robótica educacional?”, apresentadas na Tabela 3.6 e na Figura 3.7, há uma ferramenta predominante e mais citada na maioria dos artigos: o Scratch.

Tabela 3.6: Panorama geral das ferramentas encontradas para responder a QP2

ID	Ferramenta	Linguagem Visual	Linguagem Textual	Computação	Robótica
[P01]	Pencil.cc	X			
[P02]	Scratch, Alice, App Inventor e Snap	X	X	X	
[P03]	CSUnplugged	X			
[P04]	LEGO Robotics (LabView)	X			X
[P05]	Nome a definir (Google Blockly)	X		X	
[P06]	BlockLang	X		X	
[P07]	CT - Blocks	X		X	
[P08]	Map-Blocks	X		X	
[P09]	Scratch	X		X	
[P10]	Nome a definir (Google Blockly)	X	X	X	
[P11]	Scratch e Alice	X		X	
[P12]	ABA	X		X	X
[P13]	Pirate plunder, Scratch	X	X		X
[P14]	RITA en RED	X	X		X

**Tabela 3.6** Panorama geral das ferramentas encontradas para responder a QP2

ID	Ferramenta	Linguagem Visual	Linguagem Textual	Computação	Robótica
[P15]	Nome a definir (Google Blockly)	X			X
[P16]	App Inventor	X		X	
[P17]	Nome a definir (Google Blockly)	X			X
[P18]	Scratch	X		X	
[P19]	Nome a definir (Google Blockly)	X		X	
[P20]	Scratch, Logo TooKid	X		X	
[P21]	Blockly VPE	X			X

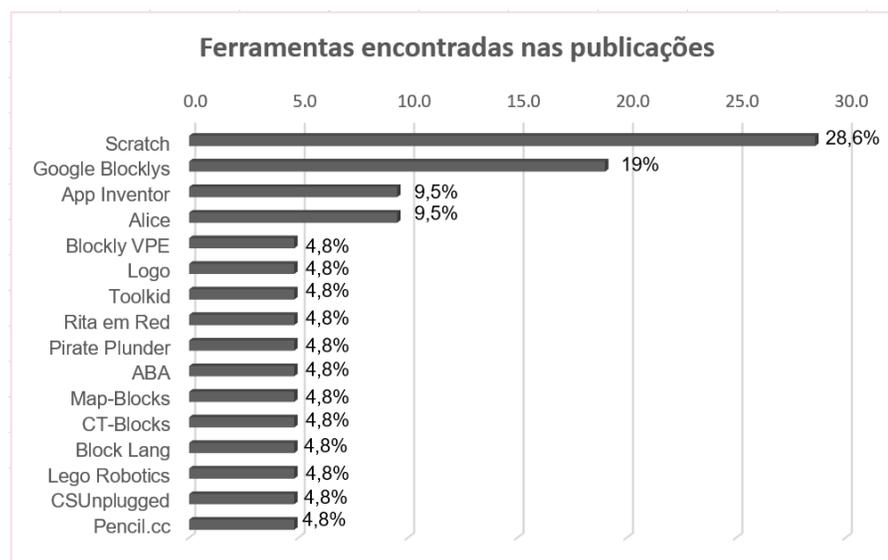


Figura 3.7: Panorama: Principais ferramentas encontradas na literatura

Conforme mostra a Tabela 5, cerca de 28,6% utilizam a ferramenta Scratch como principal ferramenta para o ensino da Computação Educacional, e essa foi a ferramenta mais citada nos trabalhos catalogados. As outras ferramentas mais citadas foram o AppInventor e Alice com em 9,5% dos artigos cada uma. O restante das publicações citaram Google Blockly como referência para a criação de suas ferramentas, isso equivale a 19% das publicações. Assim sendo, foi possível perceber que as ferramentas construídas utilizando Google Blockly oferecem mais suporte à robótica educacional, enquanto que o Scratch é mais usado na computação educacional, porém ambas são bastante úteis para promover o pensamento computacional nas escolas. A biblioteca Google Blockly é a referência quando se pretende criar uma ferramenta de acordo com cada necessidade por viabilizar a construção de uma linguagem própria.

### 3.5 Resumo do Capítulo

O Capítulo 3 apresentou os resultados um mapeamento sistemático que visou identificar as principais metodologias de ensino e ferramentas que auxiliam na aplicação de robótica ou computação educacional. Nesse caso, buscou-se saber se esses métodos usam robótica desplugada ou computação desplugada e linguagens visuais. Como resultado, foi possível perceber que a maioria dos educadores utilizam o Scratch como principal ferramenta educacional por utilizar linguagem visual baseada em blocos gráficos arrastáveis como principal método de ensino. São utilizadas conceitos de robótica e computação desplugada, pois são vistas como uma boa forma de ensino, mas o enfoque principal está no uso das ferramentas.

# Capítulo 4

## Trabalhos Correlatos

Este capítulo apresenta alguns trabalhos obtidos durante o mapeamento sistemático, descrito no Capítulo 3, que possibilitou a avaliação de estudos cujos autores fizeram uso de ferramentas ou métodos utilizados para resolver de alguma forma o problema tratado neste trabalho. Outros trabalhos que serão apresentados nesse capítulo foram obtidos na Revisão Inicial da Literatura, descrita na Seção 1.5. A Seção 4.3 mostra o comparativo dessa proposta com os trabalhos correlatos.

### 4.1 Trabalhos relacionados a Atividades Desplu- das

O trabalho de Krishnamoorthy e Kapila (2016) apresenta um experimento para promover o pensamento computacional usando métodos desplugados e empregando robôs como um fator de envolvimento dos alunos. Durante o experimento, eles foram distribuídos em dois grupos. O primeiro grupo realizou atividades desplugadas para desenvolver o pensamento computacional, enquanto o segundo grupo utilizou

atividades plugadas. O artigo decreve que os resultados são melhores para os alunos que concluíram atividades desplugadas e que existem diferenças dependendo da idade. O trabalho de Krishnamoorthy e Kapila (2016) serviu de base para incluir características inspiradas na robótica desplugada na ferramenta proposta, devido ser uma estratégia bem sucedida.

O trabalho de Miller *et al.* (2018) teve o intuito de aumentar o interesse dos alunos do ensino fundamental em atividades de ciência, tecnologia, engenharia e matemática. Para isso os autores utilizaram robótica desplugada com interface de usuário magnética tangível, isto é, blocos conectáveis de movimentos cortados em uma máquina a laser, conforme mostrado na Figura 4.1. Como resultado, os autores conseguiram que alunos de baixo nível socioeconômico alcancem interesse e o mesmo nível de aprendizagem semelhantes a seus pares de alto nível socioeconômico. O trabalho de Miller *et al.* (2018) traz contribuições de suma importância na elaboração de ideias de como os blocos devem ser desenhados nessa proposta, em termos de design e obedecendo regras de sintaxe. O Capítulo 7 mostra como os blocos gráficos foram parcialmente inspirados e adaptados desse trabalho.

## 4.2 Trabalhos Relacionados a Linguagens Visuais e Robótica Educacional

O trabalho de Bravo *et al.* (2017) examina as principais estratégias de programação de robôs e como elas são usadas no contexto educacional. Os autores analisaram as vantagens e desvantagens das estratégias de programação mais comuns em RE, como programação baseada em texto, visual e tangível. Além disso, a pesquisa explorou a abordagem da programação por demonstrações PbD (Pesquisa Baseada em Design)

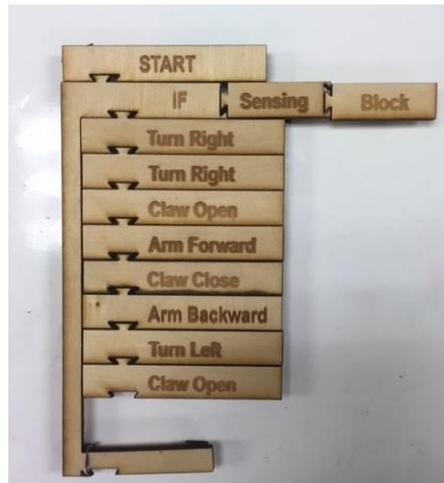


Figura 4.1: A interface de usuário magnética tangível utilizada no trabalho de Miller et. al. (2018).

em um contexto educacional. Eles identificaram as principais técnicas de PbD, como demonstrações através da manipulação do corpo do robô aprendiz ou um modelo do robô, demonstrações através dos movimentos corporais de um robô, demonstrações através do comportamento de um objeto, entre outras. O documento também identifica as técnicas de programação que permitem que não programadores criem um programa de robô facilmente. Além disso, apresentam oportunidades de pesquisa de programação intuitiva de robôs em um contexto educacional. Atualmente, o escopo e o impacto das atividades baseadas em robôs se concentram principalmente no ensino e aprendizagem de disciplinas técnicas. Deste modo, os autores discutem algumas características que os ambientes de programação de robôs devem ter para facilitar a integração da RE nos processos de aprendizado e ensino de disciplinas escolares não técnicas. É importante frisar que o trabalho de Bravo *et al.* (2017) traz ideias interessantes que foram reaproveitadas para elaboração do estudo de viabilidade que será apresentado no Capítulo 7.

Baião *et al.* (2017) apresentaram uma proposta para o ensino de corrente elétrica para o terceiro ano do ensino médio, com o auxílio do Scratch para Arduino. O objetivo foi implementar a interação da linguagem de programação Scratch para Arduino e suas *shields* (placas de circuito que podem ser conectadas ao Arduino, ampliando suas funcionalidades) para trabalhar de forma prática o desenvolvimento dos conhecimentos e habilidades de alunos no ensino de física. Como resultados, foi observado que Scratch para Arduino têm características para a incorporação de instrumentos tecnológicos modernos em sala de aula com facilidade de uso, baixo custo e por focar no desenvolvimento de competências como criatividade, criticidade, pensamento sistêmico e trabalho em equipe. O trabalho de Baião *et al.* (2017) serviu de base para o esboço de algumas funcionalidades futuras na ferramenta proposta, no que tange à construção dos blocos personalizados para fazer atividades semelhantes gerando código de forma automática.

O trabalho de Goud Yadagiri *et al.* (2017) considera o uso de um ambiente visual baseado em blocos para demonstrar e ensinar programação de robôs para alunos do ensino fundamental e médio. Um ambiente de programação visual é criado usando a biblioteca Blockly de código aberto, baseada em JavaScript, desenvolvida pelo Google. Para fins ilustrativos, eles empregaram um computador de placa única de baixo custo, como o Raspberry Pi, com microcontroladores incorporados, como o Brick Pi para LEGO ou o Arduino UNO. Duas formas de robôs móveis são criadas para experimentação, um robô móvel de rodas e um robô móvel de duas pernas. Para comandar e controlar cada robô móvel, a ferramenta visual desenvolvida emprega blocos correspondentes a construções básicas de programação, como loops, instruções condicionais, variáveis e procedimentos. Para demonstrar o uso dessa abordagem eles propuseram um jogo educacional baseado em labirinto. Especificamente, o jogo exige que o usuário programe um robô por meio da ferramenta visual deles para navegar

pelo labirinto e marcar pontos distribuídos por todo o labirinto. O trabalho de Goud Yadagiri *et al.* (2017) serviu de base para a escolha de quais tecnologias seriam utilizadas nessa proposta, para a construção da linguagem baseada em blocos.

Sales *et al.* (2017) relataram a experiência de uso do Arduino programado com Scratch para o desenvolvimento de conteúdos de matemática. O estudo visou compreender se a programação com Scratch poderia potencializar o interesse dos alunos pelas aulas de geometria. Como metodologia, eles propuseram uma sequência didática na qual alunos da educação básica com faixa etária entre 11 e 16 anos, de uma escola de rede pública municipal, criaram e programaram cubos de LED nas aulas de matemática, desenvolveram aprendizagens relacionadas ao pensamento computacional e a geometria. Como resultado, ficou evidenciado que os alunos foram capazes de interagir e aprender matemática com êxito, em um ambiente de aprendizagem motivador, utilizando linguagens de programação. O trabalho de Sales *et al.* (2017) serviu como base para a elaboração do estudo de viabilidade na primeira versão da ferramenta proposta nesse trabalho.

O trabalho de Queiroz *et al.* (2017) apresentou uma ferramenta de programação visual voltada para a RE com a utilização de componentes de baixo custo como o Arduino e tecnologias gratuitas. O objetivo dos pesquisadores foi apoiar o ensino-aprendizagem de lógica de programação para promover o pensamento computacional fazendo experimentos em escolas públicas do Rio de Janeiro. Os resultados preliminares foram promissores, pois despertou o interesse de alunos e professores pela ferramenta proposta, principalmente pela RE. O trabalho de Queiroz *et al.* (2017) servirá de base para elaboração das atividades de verificação e validação da ferramenta, além de ter ajudado a definir quais artefatos e tecnologias serão utilizados ao longo desse trabalho.

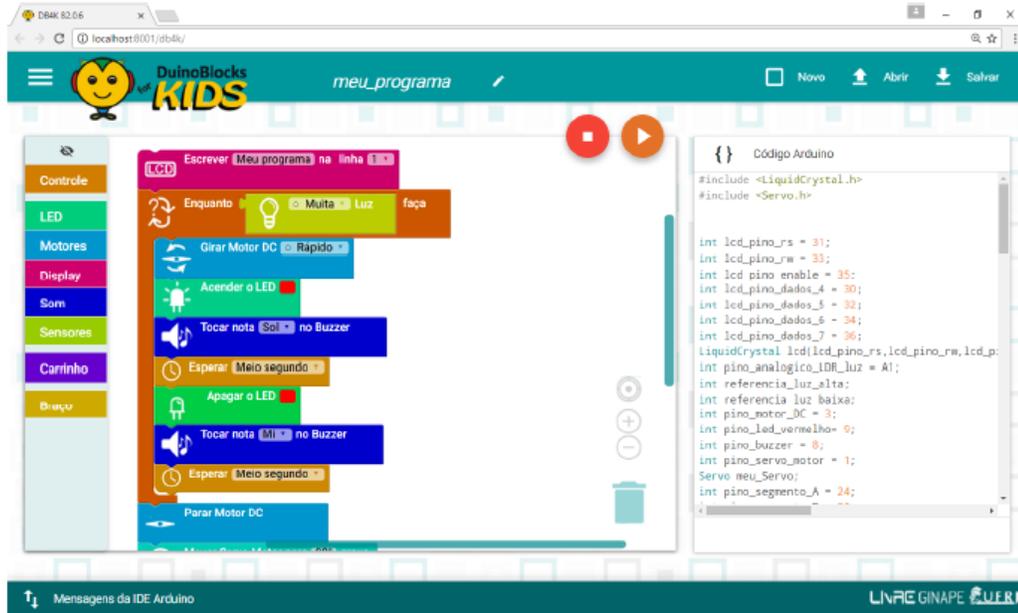


Figura 4.2: DuinoBlocks

### 4.3 Comparativo da Proposta com os Trabalhos Correlatos

A Tabela 4.1 mostra as principais características dos trabalhos correlatos com o intuito de mostrar um comparativo com a proposta desse trabalho.

Tabela 4.1: Comparação da Proposta com os Trabalhos Correlatos

ID	Trabalhos Correlatos	Comparação da Proposta com os Trabalhos Correlatos					
		Ferramenta Educacional Voltada para Robótica	Linguagem Visual Baseada em Blocos	Robótica Plugada	Robótica Desplugada	Robótica Educacional	Geração Automática de Código
1	Krishnamoorthy e Kapila (2016)		X	X	X		
2	Miller et al. (2018)		X		X	X	

3	Bravo et al. (2017)	X	X	X		X	
4	Baião et al. (2017)	X	X	X		X	
5	Goud Yadagiri et al. (2017)	X	X	X			
6	Sales et al. (2017)	X	X			X	
7	Queiroz et al. (2017)	X	X	X		X	X
8	Proposta	X	X	X	X	X	X

Percebe-se observando a Tabela 4.1 que o grande diferencial do método utilizado está em duas características: Robótica Desplugada e Geração Automática de Código. Cada trabalho correlato contribui para implementação de todas as características que estarão presentes na ferramenta: (i) Ferramenta Educacional voltada para a robótica, porque o foco não é ensino de conceitos de ciência de computação ou programação, somente RE, que pode auxiliar o ensino-aprendizagem de outras ciências na educação básica; (ii) Linguagem Visual Baseada em Blocos, porque isso mostrou ser uma estratégia bem sucedida em aplicações convencionais, tais como Scratch e AppInventor, também apresentados nos resultados do Capítulo 3 e amplamente aceitos pela comunidade científica; (iii) Robótica Plugada, porque a ferramenta pode ser utilizada para movimentar robôs educacional físicos; (iv) Robótica Desplugada, porque os blocos gráficos serão inspirados nos blocos magnéticos utilizados em robótica desplugada e além disso, haverá uma tela onde um robô virtual fará os movimentos programados pelos blocos, o que exclui totalmente a necessidade de um robô físico; (v) RE, porque é uma ferramenta voltada exclusivamente para um contexto educacional; (vi) Geração Automática de Código, porque a ferramenta vai possibilitar a geração de código-fonte em linguagem C para ser embarcado em um microcontrolador.

## 4.4 Resumo do Capítulo

Esse capítulo apresentou sete trabalho correlatos, mostrando como esses trabalhos contribuíram para o desenvolvimento da proposta, além de fazer um comparativo geral fazendo um *checklist* das principais características envolvidas em cada trabalho em relação à proposta.

# Capítulo 5

## Método Utilizado

Pressman e Maxim (2016) definem processo como "um conjunto de atividades, ações e tarefas realizadas na criação de algum produto de trabalho". No contexto de Engenharia de Software, o processo é uma abordagem adaptável que possibilita à equipe de software realizar o trabalho de selecionar e escolher o produto apropriado de ações e tarefas. Nesse sentido, uma metodologia de processo genérica estabelece cinco atividades: comunicação, planejamento, modelagem, construção e emprego.

Para este trabalho, será utilizado um modelo de processo prescritivo, que tem por objetivo prescrever um conjunto de elementos (ações de engenharia de software, produtos de trabalho, artefatos) e a maneira como esses elementos se inter-relacionam. Os modelos de processo de software utilizam as cinco atividades metodológicas genéricas, podem cada um dá ênfase diferente a essas atividades. Assim sendo, dentre os modelos de processo prescritivos existentes, foi selecionado o Modelo Espiral, por ser um processo adequado para softwares que precisam passar por inúmeras evoluções na medida que o desenvolvimento acontece. Esse processo foi criado por Boehm (1988) e é uma melhoria do modelo incremental, que entrega as

partes prontas uma por vez. O modelo espiral permite fazer sucessivos refinamentos, quantas vezes forem necessárias, conforme é possível observar na Figura 5.1 .

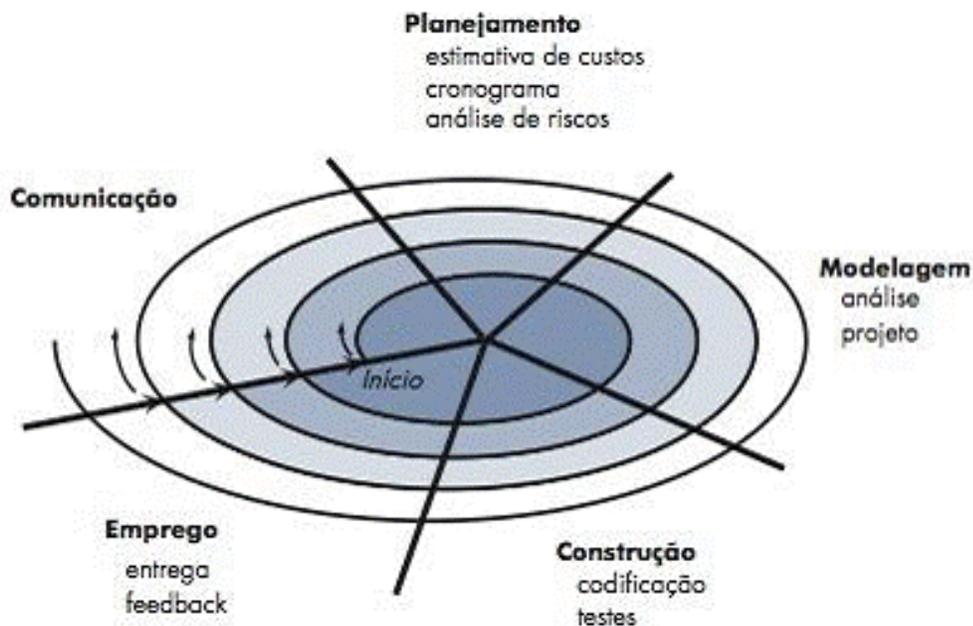


Figura 5.1: Modelo Espiral descrito por Pressman e Maxim (2016)

Uma grande vantagem desse modelo, segundo Sommerville (2011) é a combinação entre prevenção e tolerância a mudanças, assumindo que mudanças são um resultado de risco de projeto. Outra vantagem é que os riscos técnicos são levados em consideração em todos estágios de evolução, o que pode reduzir riscos antes de se tornarem problemas.

Deste modo, este capítulo está dividido em cinco seções, que descrevem as atividades do modelo espiral envolvidas para a execução do método utilizado.

## 5.1 Comunicação

Nessa fase, a intenção é compreender os objetivos de todas as pessoas interessadas no projeto, além do levantamento das necessidades que ajudarão a definir as funções e características do software (Pressman e Maxim 2016). Deste modelo, essa fase está descrita na Subseção 1.5.1 (Fase de concepção da Tecnologia), que mostra como foi feita a Revisão Inicial da Literatura e o Mapeamento Sistemático (descrito no Capítulo 3).

## 5.2 Planejamento

Essa fase descreve as tarefas técnicas a serem realizadas, os riscos prováveis, os recursos que deverão ser utilizados, os artefatos e cronograma (Pressman e Maxim 2016). Assim sendo, percebeu-se que a maioria de ferramentas educacionais baseadas em linguagens visuais que são encontradas na literatura utiliza blocos gráficos arrastáveis interconectados como os principais artefatos no desenvolvimento de suas atividades relacionadas à computação, tornando-se uma estratégia diferenciada em virtude do alto nível de abstração na etapa de implementação. Desse modo, no contexto de RE os alunos precisam interagir com uma interface, com características de um ambiente que possibilite a criação de comandos para controlar um robô apenas arrastando blocos e conectando-os, para que desta forma o código-fonte que será embarcado no robô seja gerado automaticamente, e principalmente de forma organizada. Deste modo, foram definidos os materiais (software/hardware) envolvidos no projeto:

- Biblioteca Google Blockly: para a construção dos blocos personalizados e aproveitamento de esboços de scripts para geração automática de códigos.

- **Arduíno:** segundo Monk (2013) Arduíno é uma pequena placa de microcontrolador contendo um plugue de conexão USB que permite a conexão com um computador. Essa placa possui um microcontrolador, isto é, um pequeno computador dentro de um chip, que contém tudo o que haviam nos primeiros computadores domésticos (um processador, um ou dois quilobytes de memória RAM para guardar dados e pinos de entrada e saída). Através do Arduíno, é possível conectar circuitos eletrônicos aos terminais de modo que ele possa controlar outros dispositivos, por isso ele é um recurso que pode ser utilizada na área de robótica. Nesse trabalho, TinkerCAD foi utilizado para simular o funcionamento de componentes robóticos em Arduíno (conforme será mostrado no Capítulo 6).
- **Robô Educacional:** esses robôs podem ser carrinhos, braços robóticos ou outros robôs utilizados na RE. Como um primeiro estudo de viabilidade, foi utilizado um braço robótico. Nos experimentos finais, foram utilizados além desse braço robótico, um robô Thymio II em ambiente 3D chamado robotbenchmark, uma versão web da ferramenta Webots. O Webots foi utilizado para estudos dos robôs 3D e validação de código gerado.
- **Tecnologias de Desenvolvimento Web:** XML, NextJS e Bootstrap 5.1.3 (ferramenta de design que engloba HTML e CSS) para organização e a estilização da ferramenta (front-end), Javascript para as regras envolvidas na codificação da lógica do sistema.

Além disso, foram utilizados outros softwares para auxiliar no design da ferramenta (Figma) e modelagem da arquitetura (Whimsical), além de componentes de hardware para a conexão dos circuitos eletrônicos entre Arduino e robô.

## 5.3 Modelagem

Nessa fase é feito um esboço do projeto, de modo que possa ser a ideia do todo. O intuito é compreender melhor o problema e como resolvê-lo. Entender as necessidades do software e o projeto que irá atender a essas necessidades (Pressman e Maxim 2016). Assim sendo, o padrão arquitetural definido nesse projeto é o MVC (do inglês, *Model, View e Controller*), um modelo de três camadas físicas que se tornou a arquitetura padrão para softwares baseados na web.

Segundo Rocha (2018), a Camada *Model* serve para a manipulação, leitura e escrita de dados, além de suas validações, enquanto que a Camada *View* é a camada de interação com o usuário. Ela apenas faz a exibição dos dados, por meio de HTML ou CSS. A Camada *Controller* é responsável por receber todas as requisições do usuário, organizá-las e enviar para a camada *Model*. Posteriormente, a camada *Controller* envia os resultados para a camada *View*.

Desta forma, conforme mostra a Figura 5.2, o MVC será utilizado por ser bastante vantajoso em termos de reaproveitamento de código e regras, facilidade de manutenção, facilidade na implementação de camadas de segurança, facilidade de atualização da interface da aplicação.

- **Controller**: Essa camada possui os scripts responsáveis por organizar os blocos conectados e enviar para a camada Model para ser processados. Em outras palavras, é nessa camada que as requisições são feitas pelos usuários. Por exemplo, quando o usuário arrastar um bloco “gire o braço 90º a direita” e em seguida conectar com um bloco denominado “abrir garra”, o Controller pegará esses dados, organizará em pequenos métodos e enviará para camada *Model* para serem processados. Após processados, esses dados são retornados ao Controller de forma "bruta", são "lapidados" e enviados para a camada View para

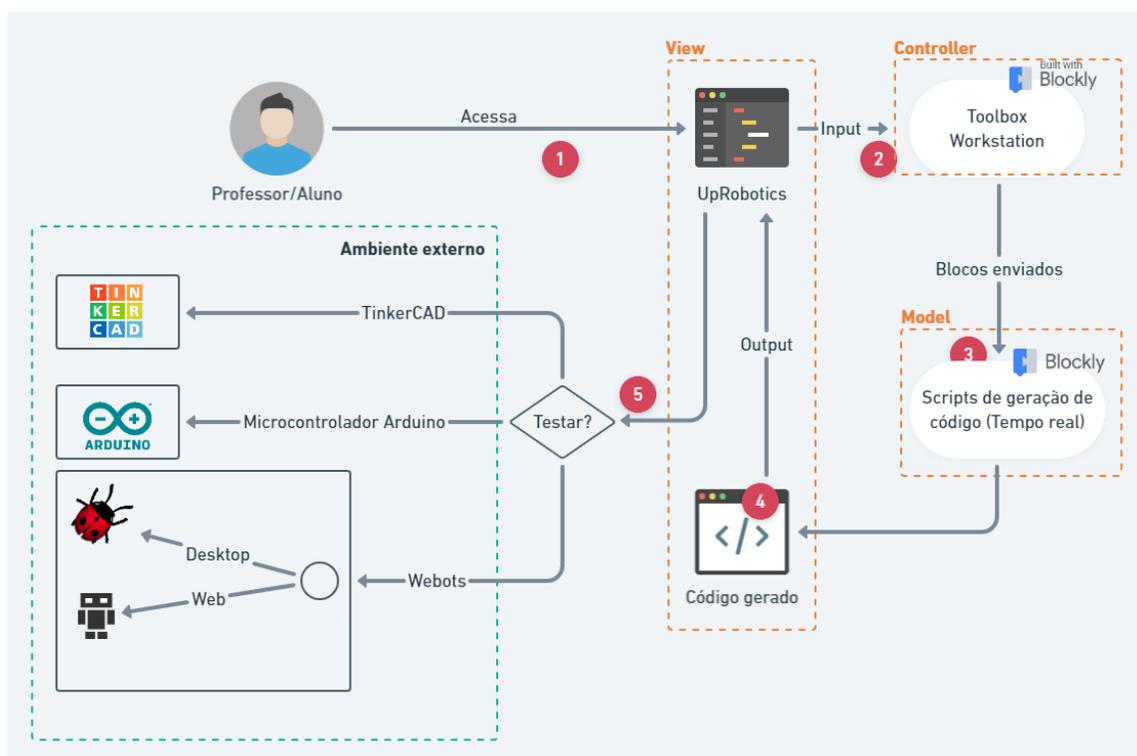


Figura 5.2: Arquitetura do UpRobotics.

serem exibidos na interface.

- **Model:** É onde estão os scripts responsáveis pela geração de código, a biblioteca Google Blockly disponibiliza esses scripts para que códigos sejam gerados na linguagem desejada. Assim sendo, nessa camada serão processados os dados enviados pelo Controller, geralmente na forma de arquivos XML e Javascript devidamente identificados. Dependendo da lógica definida pelo usuário na utilização desses blocos, o código será gerado a nível de software embarcado. Após gerar os códigos, eles serão mandados de volta para a camada Controller para que a mesma reorganize e mande os resultados para serem exibidos na camada *View*.

- **View:** É a camada de interação com o usuário, ela será responsável por exibir os resultados dos códigos gerados. Nela haverá uma caixa de ferramentas, isto é, o menu de blocos disponíveis para serem arrastados conforme a necessidade dos alunos. Haverá também uma área de trabalho, isto é, local para onde os blocos serão arrastados e interconectados.

Deste modo, o roteiro geral descrito na Figura 5.2 é definida pelos seguintes a seguir:

- Passo 1: O usuário acessa a interface de interação do UpRobotics.
- Passo 2: O usuário programa o robô utilizando os blocos.
- Passo 3: Os blocos já conectados e com valores definidos são enviados para a camada de modelo para serem processados. Os códigos são gerados em tempo real pelos scripts de geração de código.
- Passo 4: O código gerado está pronto, mas precisa ser organizado e indentado no mesmo formato de um código-fonte de uma IDE do Arduino ou editor do Webots. O código gerado e organizado é exibido na mesma interface onde os blocos estão conectados, ao lado da área de trabalho da ferramenta.
- Passo 5: O código gerado é copiado e, dependendo do tipo de robô escolhido, para testar o algoritomo, o usuário pode embarcar o Arduino por meio da IDE do Arduino ou testar no Tinkercad. Se for um robô 3D, código será testado pelo webots.

A Figura 5.3 mostra o diagrama de atividades, onde é possível observar o processo da utilização da ferramenta, desde quando o usuário seleciona o robô

até quando ele testa o mesmo, podendo selecionar um novo robô e criar um outro algoritmo.

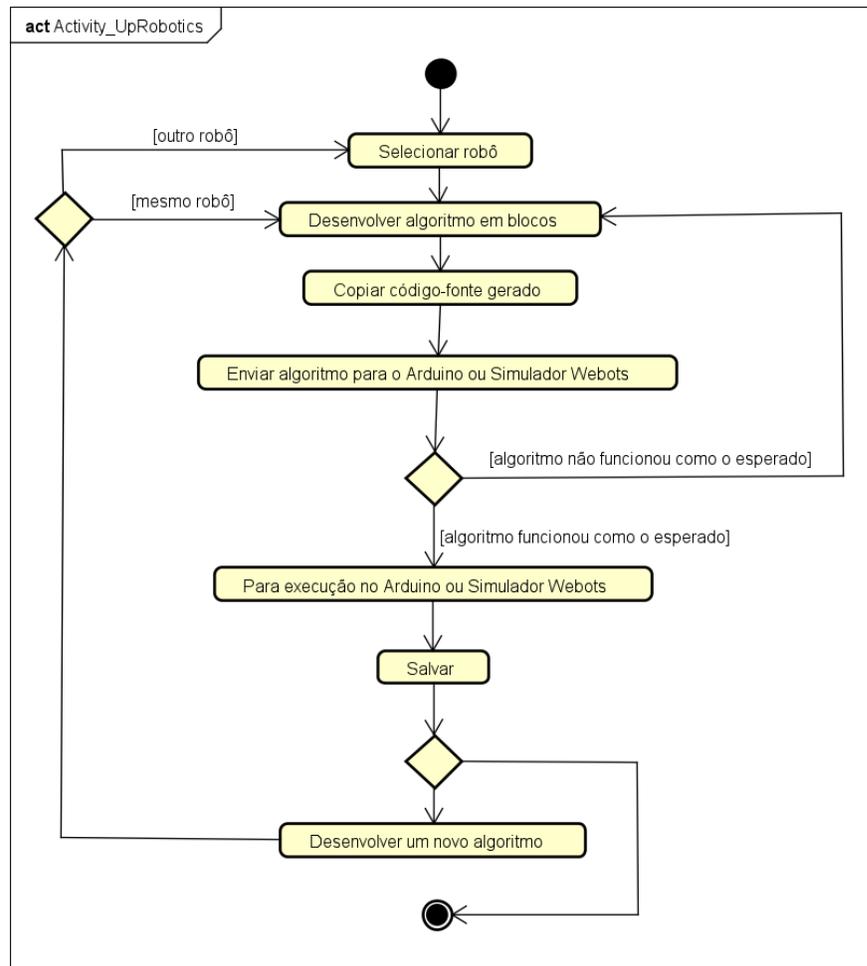


Figura 5.3: Diagrama de atividades da ferramenta

## 5.4 Construção

Essa atividade combina geração de código e testes necessários para revelar erros na codificação (Pressman e Maxim 2016). Deste modo, tendo como público alvo

desenvolvedores de software, o Google Blockly é uma biblioteca que adiciona um editor de código visual a aplicativos da Web e móveis. O editor Blockly usa blocos gráficos interligados para representar conceitos de código como variáveis, expressões lógicas, loops e muito mais. Ele permite que os usuários apliquem princípios de programação sem ter que se preocupar com sintaxe ou com a intimidação de um cursor piscando na linha de comando. O Blockly também pode ser definido como uma interface de usuário pronta para criar uma linguagem visual que emite código gerado pelo usuário sintaticamente correto, nas linguagens Javascript, Python, PHP, Lua e Dart. O Google Blockly possui diversas ferramentas que serão utilizadas nesse projeto, como por exemplo o Google Developer Tools que permite a criação de blocos personalizados usando Block Factory e Block Exporter, e Workspace Factory (Pasternak *et al.* 2017).

No que se refere aos testes de software, complementando o que está descrito na Seção 1.5.2 (Fase de Avaliação da Tecnologia), serão aplicadas técnicas de Verificação e Validação (V & V) que, segundo Sommerville (2011) é a denominação dada a processo de verificação e análise, que ocorrem em cada estágio do processo de software. A verificação foca em verificar se o software está de acordo com as especificações (estamos construindo o produto corretamente?). A validação visa assegurar que o sistema atende as expectativas do cliente (estamos construindo o produto correto?).

## 5.5 Emprego

É a entrega ao cliente que avalia o produto e fornece feedback, baseado na avaliação (Pressman e Maxim 2016). Para facilitar o acesso, o produto final será disponibilizado na internet. Visando obter feedback dos usuários, terão duas opções: "Relatar um

erro" e "Sugerir melhorias". Baseado na avaliação dos usuários, serão feitos ajustes e melhorias contínuas, empregando práticas de Engenharia de Software descritas na literatura.

## 5.6 Geração Automática de Código

A geração automática de código é uma funcionalidade muito importante nesse trabalho, pois ela se dá através dos blocos gráficos. Assim sendo, a seguir segue um exemplo de como funciona alguns blocos e seus respectivos códigos gerados na primeira versão do UpRobotics. Nesse exemplo, são blocos específicos para o braço robótico. Os blocos atuais da ferramenta estão disponíveis no Apêndice E.

### 5.6.1 Funções



Figura 5.4: Bloco importar motores

```
1
2 #include <Servo.h>
3 Servo base, angulo, altura, garra;
4 int pos, atual;
5 int valorGarraInicial, valorAlturaInicial, valorAnguloInicial,
   valorBaseInicial;
6 int valorGarraFinal, valorAlturaFinal, valorAnguloFinal,
   valorBaseFinal;
```

Listagem 5.1: Código gerado a partir do bloco importar motores

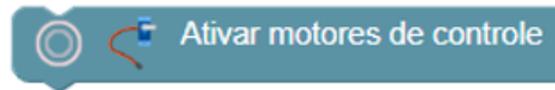


Figura 5.5: Bloco ativar motores

```
1
2 int igualarPosInitFinal(int PosInit, int PosFinal){
3     PosInit = PosFinal;
4     return PosInit;
5 }
6
7 int inicializarBraco(int bas, int ang, int alt, int gar){
8     valorBaseInicial = bas;
9     valorAnguloInicial = ang;
10    valorAlturaInicial = alt;
11    valorGarraInicial = gar;
12    valorBaseFinal = valorBaseInicial;
13    valorAnguloFinal = valorAnguloInicial;
14    valorAlturaFinal = valorAlturaInicial;
15    valorGarraFinal = valorGarraInicial;
16
17    base.write(valorBaseInicial);
18    angulo.write(valorAnguloInicial);
19    altura.write(valorAlturaInicial);
20    garra.write(valorGarraInicial);
21    delay(300);
22 }
23 int moverServomotor(int a, int b, Servo motor){
```

```
24 int pos;
25 if(a<b){
26     for(pos = a; pos <= b; pos +=1)
27     {
28         motor.write(pos);
29         delay(15);
30     }
31 }
32 if(a>=b){
33     for(pos = a; pos >= b; pos -=1)
34     {
35         motor.write(pos);
36         delay(15);
37     }
38 }
39 }
```

Listagem 5.2: Código gerado a partir do bloco ativar motores

## 5.6.2 Arduino

Definir portas no Arduino

base: 11 , ângulo: 9 , altura: 10 e garra: 6

Figura 5.6: Bloco definir portas

```
1 void setup()
2 {
3     Serial.begin(9600);
```

```
4 base.attach(11);
5 angulo.attach(9);
6 altura.attach(10);
7 garra.attach(6);
8 }
```

Listagem 5.3: Código gerado a partir do bloco definir portas no Arduino

### 5.6.3 Controle



Figura 5.7: Bloco controle de movimentos

```
1 void loop()
2 {
3
4 }
```

Listagem 5.4: Código gerado a partir do bloco controle de movimentos

### 5.6.4 Movimentos



Figura 5.8: Bloco inicializar braço

```
1 inicializarBraco(90,90,90,25);
```



Figura 5.9: Blocos abrir garra e fechar garra

```
1 valorGarraInicial=igualarPosInitFinal(valorGarraInicial ,  
    valorGarraFinal);  
2 valorGarraFinal = 145;  
3 atual = igualarPosInitFinal(valorGarraInicial,valorGarraFinal);  
4 moverServomotor(valorGarraInicial, atual, garra);
```

Listagem 5.5: Código gerado a partir dos blocos abrir garra e fechar garra



Figura 5.10: Blocos girar braço para direita ou para a esquerda

```
1     valorBaseInicial = igualarPosInitFinal(valorBaseInicial ,  
        valorBaseFinal);  
2 valorBaseFinal = 160;  
3 atual = igualarPosInitFinal(valorBaseInicial,valorBaseFinal);  
4 moverServomotor(valorBaseInicial, atual, base);
```

Listagem 5.6: Código gerado a partir dos blocos girar braço para direita ou para a esquerda

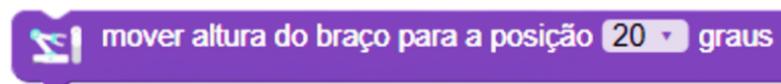


Figura 5.11: Blocos mover altura do braço

```
1     valorAlturaInicial = igualarPosInitFinal(  
        valorAlturaInicial, valorAlturaFinal);  
2 valorAlturaFinal = 20;  
3 atual = igualarPosInitFinal(valorAlturaInicial, valorAlturaFinal  
    );  
4 moverServomotor(valorAlturaInicial, atual, altura);
```

Listagem 5.7: Código gerado a partir do bloco mover altura do braço

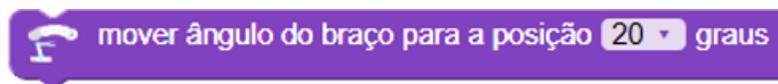


Figura 5.12: Bloco mover ângulo do braço

```
1 valorAnguloInicial = igualarPosInitFinal(valorAnguloInicial,  
        valorAnguloFinal);  
2 valorAnguloFinal = 20;  
3 atual = igualarPosInitFinal(valorAnguloInicial, valorAnguloFinal  
    );  
4 moverServomotor(valorAnguloInicial, atual, angulo);
```

Listagem 5.8: Código gerado a partir do mover ângulo do braço

## 5.7 Resumo do Capítulo

Esse capítulo apresentou o método do pesquisa que utilizou o modelo espiral, que é um modelo de processo prescritivo de Engenharia de Software que possui cinco etapas, desde a comunicação, que é a fase de concepção da tecnologia; planejamento, uma etapa de definição de tarefas, artefatos e ferramentas; modelagem, uma etapa

de esboço do problema e solução, construção, etapa de a codificação e testes inclusos; finalizando com a emprego, que é a disponibilização do produto final.

# Capítulo 6

## A Ferramenta UpRobotics

Este capítulo apresenta o processo de concepção de uma linguagem visual, através da modelagem formalizada e representada por um autômato finito determinístico, e uma primeira versão da ferramenta com os blocos e respectivos códigos associados, essa primeira versão foi utilizada em um estudo de caso com um robô educacionaldo tipo braço robótico, onde foram obtidos resultados e discussões a respeito do estudo de caso apresentado na Seção 7.1. A ferramenta foi nomeada UpRobotics devido a suas características serem inspiradas no conceito de Unplugged Robotics (Robótica Desplugada). Além disso, são apresentados os aprimoramentos da ferramenta como, por exemplo, a inclusão de mais alguns robôs, linguagens visuais e simuladores com ambiente 3D.

### 6.1 A Linguagem Visual

Foram realizados testes em laboratório para observar os movimentos possíveis em um braço robótico com a utilização de dois joysticks e chegou-se a um padrão de

instruções necessárias para realizar todos os movimentos conhecidos, sem a necessidade de joysticks, apenas utilizando a lógica. Visando representar a linguagem criada com sintaxe bem definida e semântica precisa, foi elaborada uma linguagem formal baseada em um Autômato Finito Determinístico (AFD) apresentado na Figura 6.1. Esse autômato é um formalismo particularmente adequado ao reconhecimento dessa linguagem. Assim sendo, as características principais desse AFD são: o autômato recebe uma palavra  $p$  de entrada, composta por uma sequência arbitrária de símbolos pertencentes ao alfabeto  $\Sigma$ . A leitura de cada símbolo em  $p$  implica na mudança de estado no autômato (Hopcroft *et al.* 2002).

A Linguagem Visual de Programação Baseada em Blocos construída para o UpRobotics é definida da seguinte maneira:  $L$  é uma linguagem pertencente ao alfabeto  $\Sigma$  de modo que  $\Sigma = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$  tal que  $L = \{ p \in \{ \{1^k 2^k 3^k 4^k\} \cup \{1^k 3^k 2^k 4^k\} \} 5^k \{6, 7, 8, 9\}^n \mid k = 1, n \geq 1 \}$  onde  $p$  é uma palavra gerada pela linguagem  $L$ .

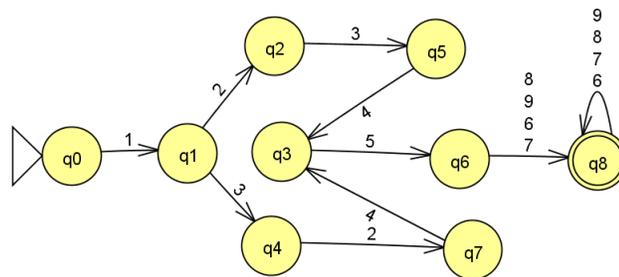


Figura 6.1: Autômato Finito da Linguagem Baseada em Bloco.

A Tabela 6.1 descreve os significados de cada estado no AFD e a Tabela 6.2 descreve o significado de cada símbolo do alfabeto. A Figura 6.2 mostra um pequeno exemplo de blocos necessários para fazer o braço apontar um objeto a frente dele e logo em seguida voltar a posição anterior.

Tabela 6.1: Significado de cada estado do autômato

Estados	Significado
{q0}	Nenhum bloco foi adicionado
{q1}	Os motores foram importados
{q2,q7}	Os motores foram ativados
{q4,q5}	As portas seriais foram definidas
{q3}	O bloco para inicializar todos os movimentos foi adicionado
{q6}	A posição inicial do braço foi definida
{q8}	A garra foi aberta ou fechada; Ou o ângulo do braço foi movido da posição atual até a nova posição informada; Ou a altura do braço foi movida da posição A até a posição B; Ou a base girou o braço para a direita ou esquerda;

Na Figura 6.2 é possível notar alguns aspectos semelhantes à didática bem sucedida abordada no trabalho de Miller *et al.* (2018), visto que os blocos gráficos do UpRobotics possuem algumas características inspiradas nos blocos conectáveis de movimentos mostrados na Figura 4.1 (Seção 4.2). O diferencial é que os blocos da UpRobotics geram código-fonte em linguagem C, que é compilado e embarcado em um Arduino por meio de uma IDE movimentando o braço robótico.

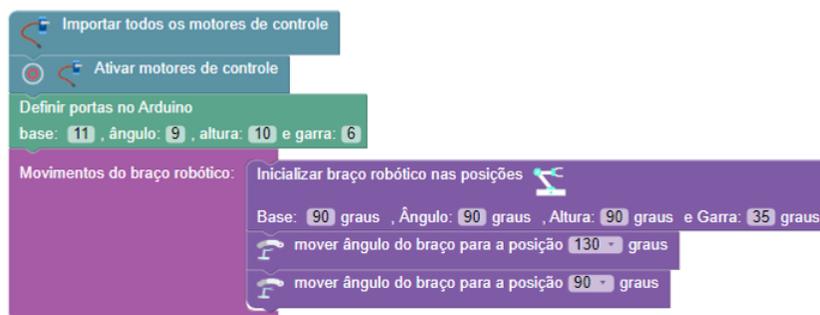


Figura 6.2: Exemplo de programação em blocos com o intuito de fazer o braço robótico apontar um objeto a frente dele e logo em seguida voltar a posição anterior.

Tabela 6.2: Significado de cada símbolo do alfabeto

Alfabeto ( $\Sigma$ )	Significado
1	Importar motores
2	Ativar motores
3	Definir as portas no Arduino
4	Inicializar todos os movimentos
5	Inicializar o robô a partir de um posição inicial padrão
6	Movimentar fechamento/abertura da garra robótica
7	Movimentar o ângulo do braço
8	Movimentar a altura do braço
9	Girar o braço para a direita ou para a esquerda

A Figura 6.3 mostra a tela inicial home e a Figura 6.4 mostra a parte principal da ferramenta UpRobotics, cuja mesma é dividida em 3 partes:

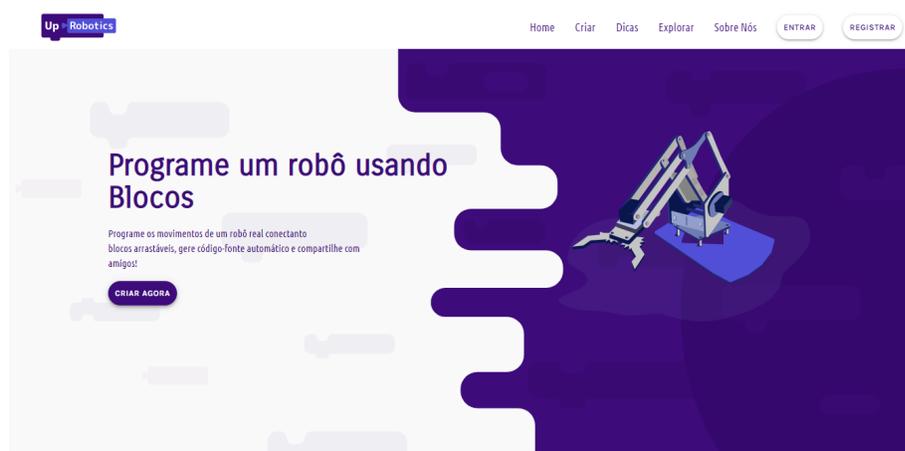


Figura 6.3: Página inicial da ferramenta criada.

1. Caixa de ferramentas: local dividido em 4 categorias, onde cada categoria possui um conjunto de blocos. As categorias são: (i) Movimentos, que possui todos os blocos principais que controlam as articulações do braço robótico,

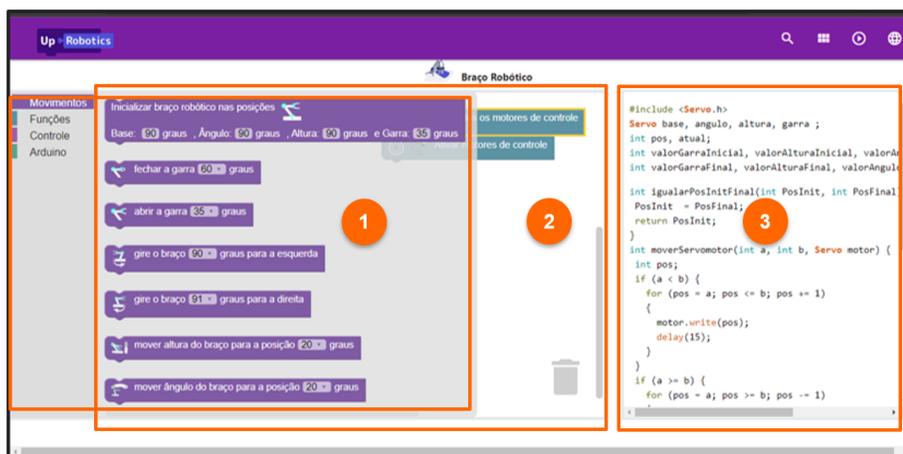


Figura 6.4: Parte principal da ferramenta criada.

isto é, os quatro Servo motores (garra, base, altura e ângulo); (ii) Funções, onde estão todas as os códigos responsáveis por importações de bibliotecas do Arduino necessários, além de conter funções responsáveis por manipular a velocidade de movimentação do braço e controle das posições de um ponto A ao ponto B; (iii) Controle, onde está um bloco responsável por inicializar todos os movimentos ou ações que devem ser executadas; (iv) Arduino, contém um único bloco para definir as portas seriais no Arduino onde serão conectados os Servo motores.

2. Área de Trabalho: local onde os blocos devem ser arrastados, montados, excluídos, preenchidos e manipulados.
3. Tela de saída do código gerado: uma tela onde os códigos gerados são mostrados de acordo com a montagem dos blocos na área de trabalho.

A Figura 6.5 mostra um exemplo dos blocos interligados na área de trabalho da ferramenta, onde é possível observar algumas características dos blocos, como por exemplo, a facilidade de perceber onde os blocos se conectam, visto que estes são

criados para serem encaixados no lugar correto. A Figura 6.6 mostra o código que é gerado a partir dos blocos interligados mostrado na Figura 6.5.

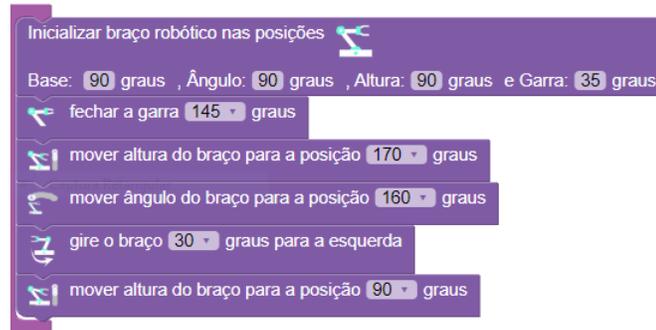


Figura 6.5: Exemplo de blocos interligados.

```
void loop()
{
  inicializarBraco(90, 90, 90, 35);

  valorGarraInicial = igualarPosInitFinal(valorGarraInicial, valorGarraFinal);
  valorGarraFinal = 145;
  atual = igualarPosInitFinal(valorGarraInicial, valorGarraFinal);
  moverServomotor(valorGarraInicial, atual, garra);

  valorAlturaInicial = igualarPosInitFinal(valorAlturaInicial, valorAlturaFinal);
  valorAlturaFinal = 170;
  atual = igualarPosInitFinal(valorAlturaInicial, valorAlturaFinal);
  moverServomotor(valorAlturaInicial, atual, altura);

  valorAnguloInicial = igualarPosInitFinal(valorAnguloInicial, valorAnguloFinal);
  valorAnguloFinal = 160;
  atual = igualarPosInitFinal(valorAnguloInicial, valorAnguloFinal);
  moverServomotor(valorAnguloInicial, atual, angulo);

  valorBaseInicial = igualarPosInitFinal(valorBaseInicial, valorBaseFinal);
  valorBaseFinal = 30;
  atual = igualarPosInitFinal(valorBaseInicial, valorBaseFinal);
  moverServomotor(valorBaseInicial, atual, base);

  valorAlturaInicial = igualarPosInitFinal(valorAlturaInicial, valorAlturaFinal);
  valorAlturaFinal = 90;
  atual = igualarPosInitFinal(valorAlturaInicial, valorAlturaFinal);
  moverServomotor(valorAlturaInicial, atual, altura);
}
```

Figura 6.6: Código gerado a partir dos blocos da Figura 6.5

## 6.2 A Ferramenta UpRobotics

O "Up" vem do termo Unplugged Robotics ("Robótica desplugada"), isto é, uma estratégia de ensino de robótica bem sucedida, que não necessita da utilização de um robô físico. O UpRobotics, ferramenta web (disponível em <https://uprobotics-ten.vercel.app/>), é um artefato de inspiração em pesquisas científicas sobre educação, estratégias e softwares educacionais. Com o UpRobotics, o usuário pode programar seus próprios robôs, testá-los, aprimorar seus algoritmos, sem ter a necessidade de compreender a sintaxe das linguagens de programação textuais como Python e C, por exemplo. A linguagem visual é convertida para linguagens que o robô irá entender. UpRobotics foi feito para o exercício do pensamento computacional, além de ajudar os jovens a aprender a pensar de forma criativa, raciocinar sistematicamente e trabalhar de forma colaborativa, habilidades essenciais no mundo moderno. A capacidade de codificar programas de computador é uma parte importante da alfabetização na sociedade atual. Quando as pessoas aprendem a codificar com o Up, aprendem estratégias importantes para resolver problemas, projetar projetos e comunicar ideias. A Figura 6.7 mostra a tela principal do UpRobotics e seus principais itens numerados de 1 a 11.

- **1:** O usuário pode selecionar qualquer robô disponível no UpRobotics.
- **2:** É a imagem do robô que está selecionado atualmente.
- **3:** É o nome do robô selecionado atualmente.
- **4:** É o nome do ambiente do robô selecionado. Por exemplo, "Alvo", "Obstáculo", "Seguidor de parede".

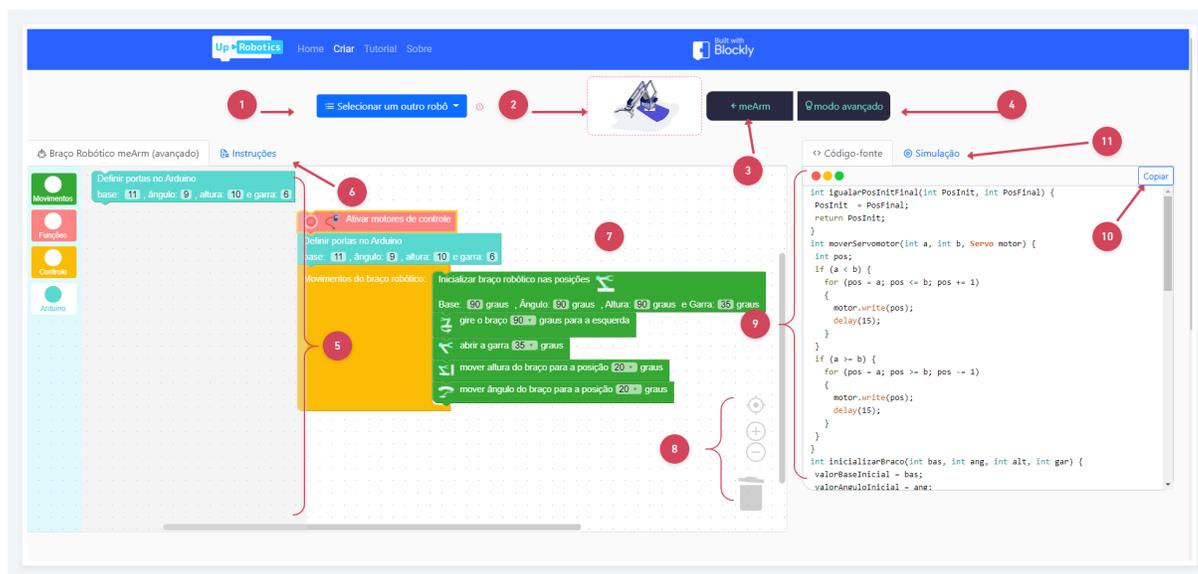


Figura 6.7: Tela principal do UpRobotics

- **5:** É a Toolbox, isto é, a caixa de ferramentas onde os blocos estão disponíveis para esse robô.
- **6:** É onde estão as instruções para que o usuário saiba qual é a sua missão ou objetivo com esse robô.
- **7:** É a workspace, isto é, a área de trabalho para onde os blocos são arrastados e conectados.
- **8:** É um recurso que permite controlar o Zoom dos blocos, ou deletar determinado bloco (basta arrastar o bloco para a lixeira).
- **9:** É onde é exibido o código que é gerado em tempo real.
- **10:** Botão que permite copiar todo o código gerado.

- **11:** É onde o usuário poderá testar o seu código, seja no simulador 3D robot-benchmark ou TinkerCAD.

A Figura 6.8 mostra como selecionar um robô. Além disso, no UpRobotics contém outras telas, como a Home (Figura 6.9), Tutorias (Figura 6.10) e Sobre (Figura 6.11).

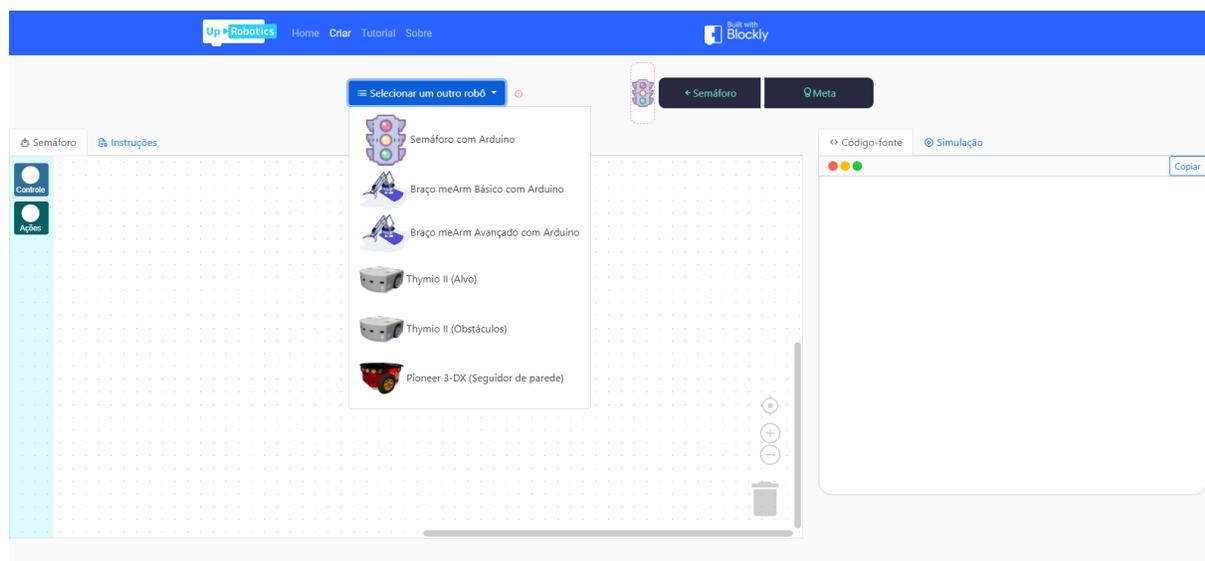


Figura 6.8: Selecionar robô no UpRobotics

### 6.3 Escolha das Aplicações a Serem Adicionadas na Ferramenta

- **Webots:** é um aplicativo de desktop de código aberto e multiplataforma usado para simular robôs. Ele fornece um ambiente de desenvolvimento completo para modelar, programar e simular robôs. Ele foi projetado para um uso profissional e é amplamente utilizado na indústria, educação e pesquisa. Cyberbotics

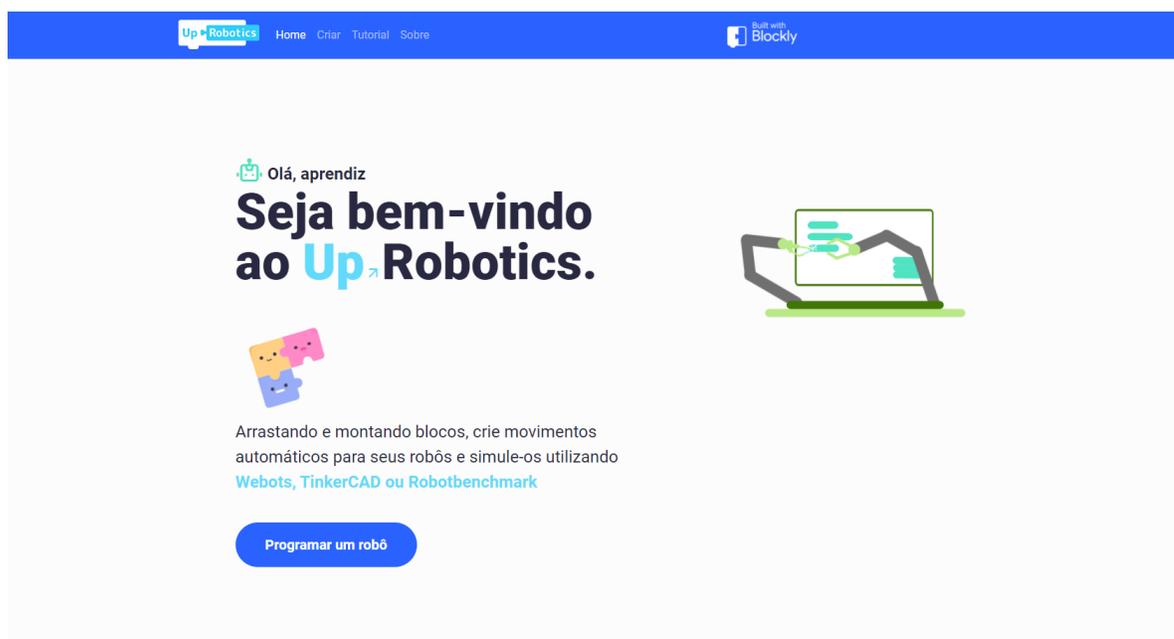


Figura 6.9: Tela home do UpRobotics

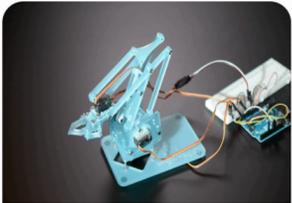
Ltd. mantém Webots como seu produto principal continuamente desde 1998. O Webots é a base do Robotbenchmark.

- Robotbenchmark: é uma aplicação online baseada em Webots. Ele oferece uma série de desafios de programação de robôs que abordam vários tópicos em uma ampla gama de níveis de dificuldade. Esses benchmarks são fornecidos gratuitamente como simulações online, com base em uma pilha de software de código aberto 100% gratuito. O desempenho alcançado pelos usuários é registrado e exibido online. NO UpRobotics, ele é utilizado para simular os carrinhos robôs: Thymio II e Pioneer 3-DX.
- TinkerCAD: é um aplicativo Web gratuito e fácil de usar que fornece a projetistas e engenheiros a próxima geração de habilidades fundamentais para inovação: projeto 3D, eletrônica e codificação. No UpRobotics, ele é utilizado

**UpRobotics** Home Criar Tutorial Sobre Built with Blockly

## Guia de atividades

No UpRobotics você pode realizar atividades com robótica desplugada, isto é, sem um robô real, para isso existem simuladores, aqui no Up utilizamos o Webots para desplugada e Arduino caso você tenha um robô. Vamos ver alguns exemplos de como trabalhar no UpRobotics?



**meArm**

O meArm é um kit de braço robótico pequeno e barato que utiliza Servo Motores e acrílico cortado a laser. Para movimentá-lo, você manipulará 4 servomotores que controlarão a base, a altura, a direção e a garra. Esses movimentos podem ser feitos utilizando 2 joysticks, mas no UpRobotics você poderá programá-lo para que o robô possa fazer movimentos automáticos para um Arduino Uno.

[Ver tutorial](#)



**Thymio II**

Thymio II é um robô educacional acessível. Ele oferece três características principais: um grande número de sensores e atuadores, uma interatividade educativa baseada em luz e toque, um ambiente de programação com programação gráfica e de texto.

[Ver tutorial](#)



**Pioneer 3DX**

Veículo terrestre não tripulado (VTNT) da empresa Adept MobileRobots o NERO possui 3 exemplares de dimensão 52x38x21,5cm. São equipados com sonares e encoders, possibilitando, por exemplo, o mapeamento de um ambiente desconhecido.

[Ver tutorial](#)

Figura 6.10: Tela de tutoriais do UpRobotics

**UpRobotics** Home Criar Tutorial Sobre Built with Blockly

## Sobre o UpRobotics

O "Up" vem do termo Unplugged Robotics ("Robótica desplugada"), isto é, uma estratégia de ensino de robótica bem sucedida, que não necessita da utilização de um robô físico. O UpRobotics é um artefato de inspiração em pesquisas científicas sobre educação, estratégias e softwares educacionais.

Com o UpRobotics, você pode programar seus próprios robôs, testá-los, aprimorar seus algoritmos, sem ter a necessidade de compreender a sintaxe das linguagens de programação textuais como Python e C, por exemplo. A linguagem visual é convertida para linguagens que o robô irá entender.

UpRobotics foi feito para o exercício do pensamento computacional, além de ajudar os jovens a aprender a pensar de forma criativa, raciocinar sistematicamente e trabalhar de forma colaborativa, habilidades essenciais no mundo moderno.

A capacidade de codificar programas de computador é uma parte importante da alfabetização na sociedade atual. Quando as pessoas aprendem a codificar com o Up, aprendem estratégias importantes para resolver problemas, projetar projetos e comunicar ideias.

Aula 1 - Conhecendo o Google Bloc...

Assistir ma... Comparilh... 1/7

Blockly

Aula 1 - Conhecendo a Biblioteca Blockly

Assistir no YouTube

Projeto Publicações Palestra Softwares utilizados

O UpRobotics é fruto da dissertação de Mestrado do Aluno Francisco Gabriel Teixeira Marinho, do Programa de Pós-Graduação em Informática (PPGI) no

Figura 6.11: Tela sobre o UpRobotics

para simular o circuito da Semáforo e do Braço Robótico.

## 6.4 Escolha dos Componentes Robóticos a Serem Manipulados pela Ferramenta

### 6.4.1 O Semáforo

O semáforo é um instrumento utilizado para controlar o tráfego de veículos e de pedestres nas grandes cidades. Utiliza uma linguagem simples e, por isso, de fácil assimilação. É composto geralmente por três círculos de luzes coloridas: vermelho, que significa "pare", verde que significa "avance" e amarelo que significa "atenção". A Figura 6.12 representa um circuito simples no Arduino UNO que simula o comportamento de um semáforo, utilizando três pequenos LEDs que representam cada cor. A Figura 6.13 representa um exemplo de algoritmo feito no Up. A Figura 6.14 representa um exemplo de código gerado para o semáforo.

### 6.4.2 O Braço Robótico (meArm)

O Braço Robótico meArm mostrado na Figura 6.15 trata-se de um braço mecânico desenvolvido para aplicação em projetos que trabalham com automação de robôs, proporcionando maiores possibilidades de movimentos e desenvolvimento de tarefas. A Figura 6.16 representa um no Arduino UNO que simula os movimentos dos servomotores que controlam o braço. A Figura 6.17 representa um exemplo de programação do braço robótico.

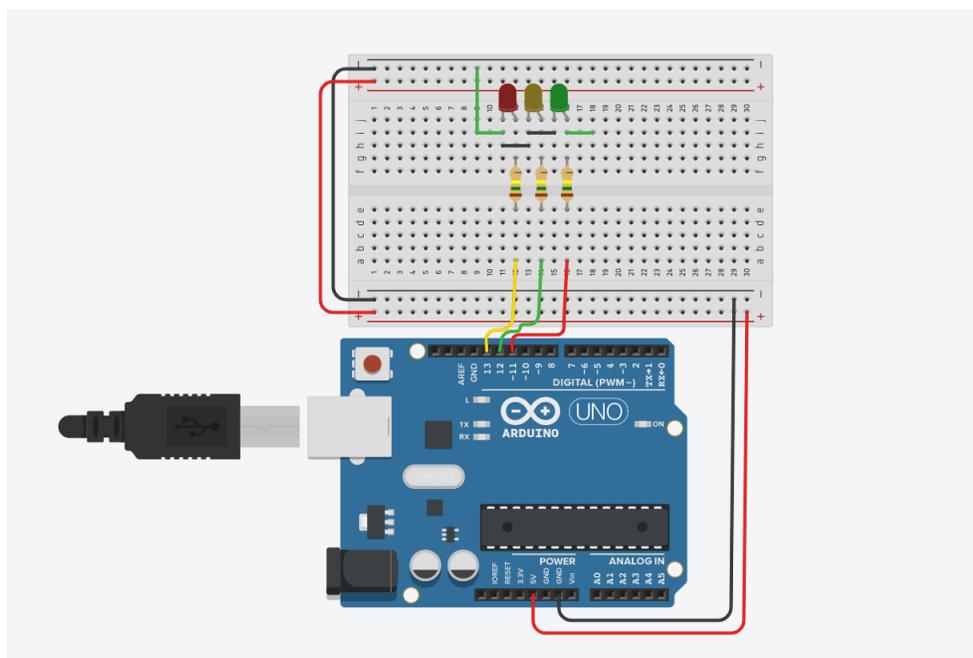


Figura 6.12: Circuito no TinkerCAD que representa um semáforo simples programável no UpRobotics

### 6.4.3 Robô Carrinho Thymio II

O Thymio é um robô educacional de código aberto projetado por pesquisadores da EPFL, em colaboração com a ECAL, e produzido pela Mobsya, uma associação sem fins lucrativos cuja missão é oferecer jornadas STEAM abrangentes e envolventes para alunos de todas as idades. No UpRobotics, é possível programá-lo e testar no simulador do robotbenchmark na aba "Simulação", conforme mostrado na Figura 6.18. No UpRobotics, existem dois cenários para esse robô, o "Alvo" como mostrado na Figura 6.18 e o "Obstáculo" mostrado na Figura 6.19. A Figura 6.21 e Figura 6.21 mostram respectivamente um exemplo de programação e um exemplo de como o Thymio é exibido no simulador.



Figura 6.13: Exemplo de algoritmo montado no UpRobotics para o semáforo

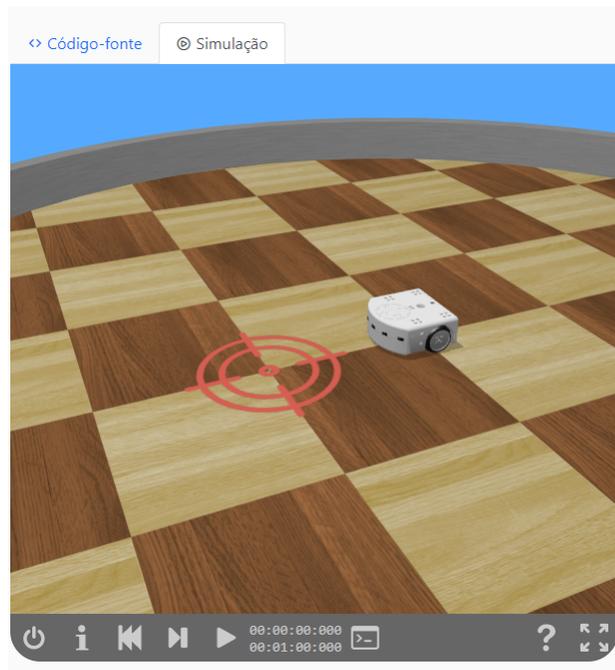
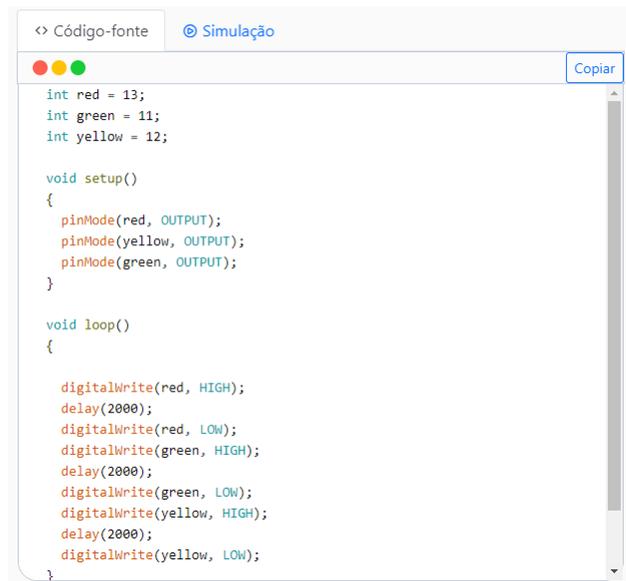


Figura 6.18: Thymio II no cenário "Alvo"



```
<> Código-fonte  @ Simulação Copiar  
  
int red = 13;  
int green = 11;  
int yellow = 12;  
  
void setup()  
{  
  pinMode(red, OUTPUT);  
  pinMode(yellow, OUTPUT);  
  pinMode(green, OUTPUT);  
}  
  
void loop()  
{  
  
  digitalWrite(red, HIGH);  
  delay(2000);  
  digitalWrite(red, LOW);  
  digitalWrite(green, HIGH);  
  delay(2000);  
  digitalWrite(green, LOW);  
  digitalWrite(yellow, HIGH);  
  delay(2000);  
  digitalWrite(yellow, LOW);  
}
```

Figura 6.14: Exemplo de código gerado no UpRobotics para o semáforo



Figura 6.19: Thymio II no cenário "Obstáculos"

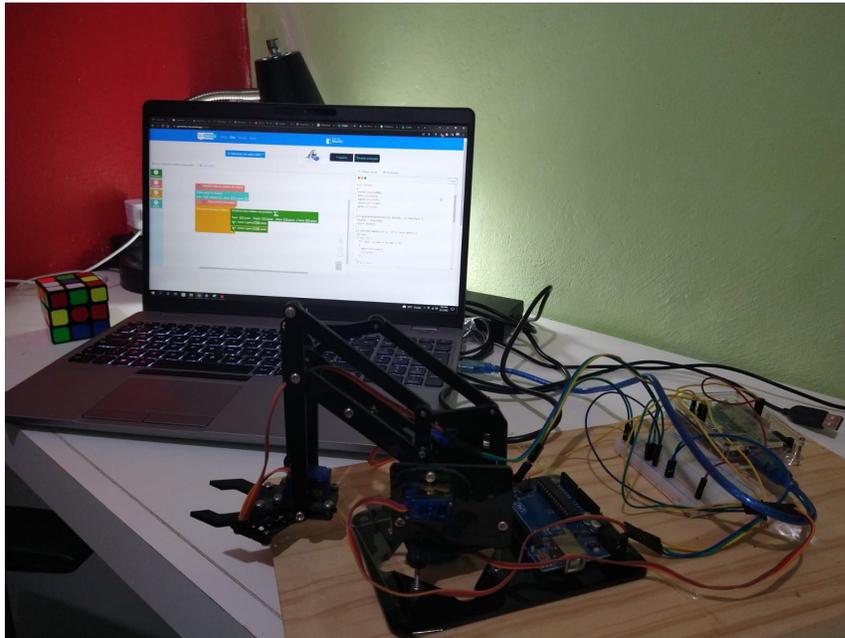


Figura 6.15: Braço robótico utilizado no experimento

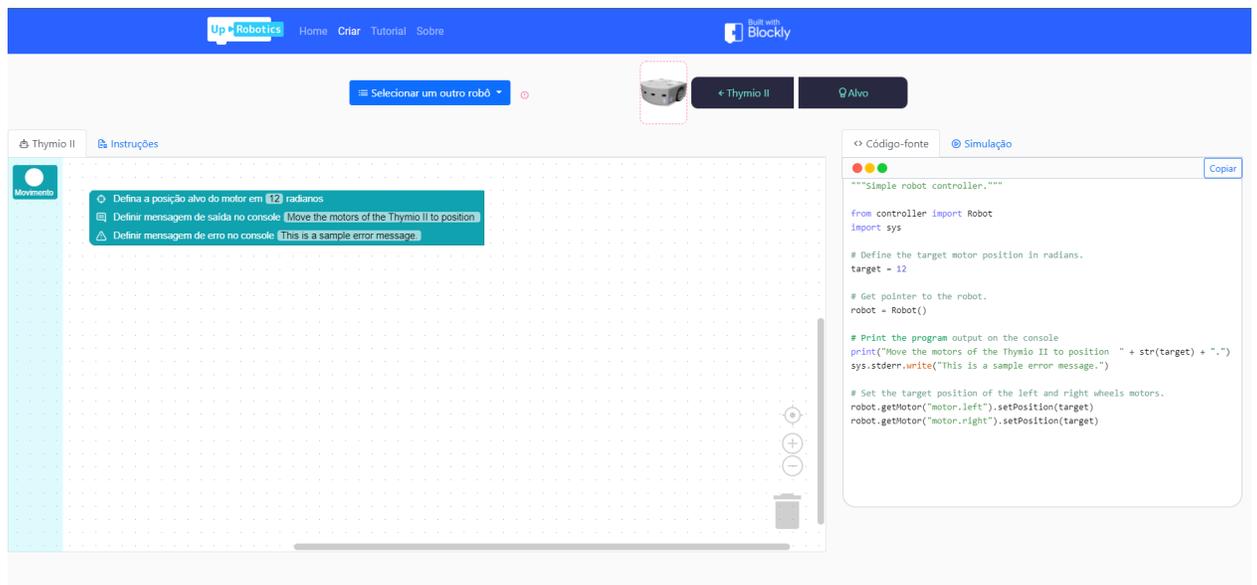


Figura 6.20: Exemplo de programação do ThymioII

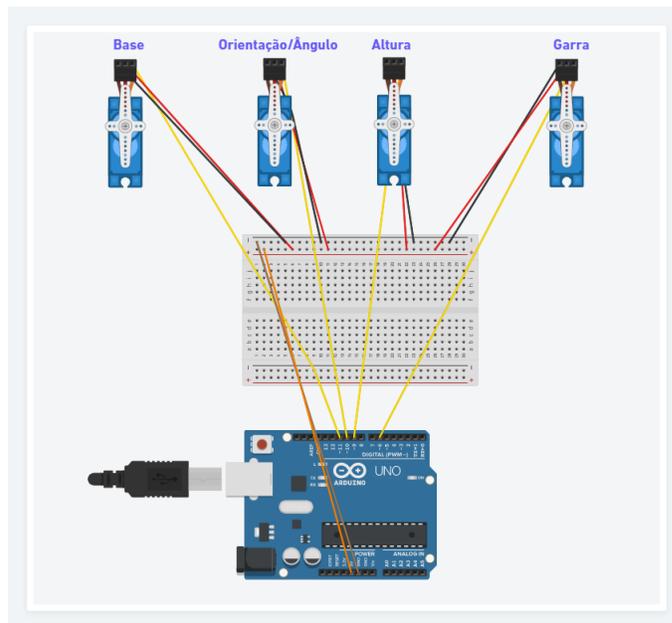


Figura 6.16: Circuito no TinkerCad que representa os servomotores que movimentam o braço robótico

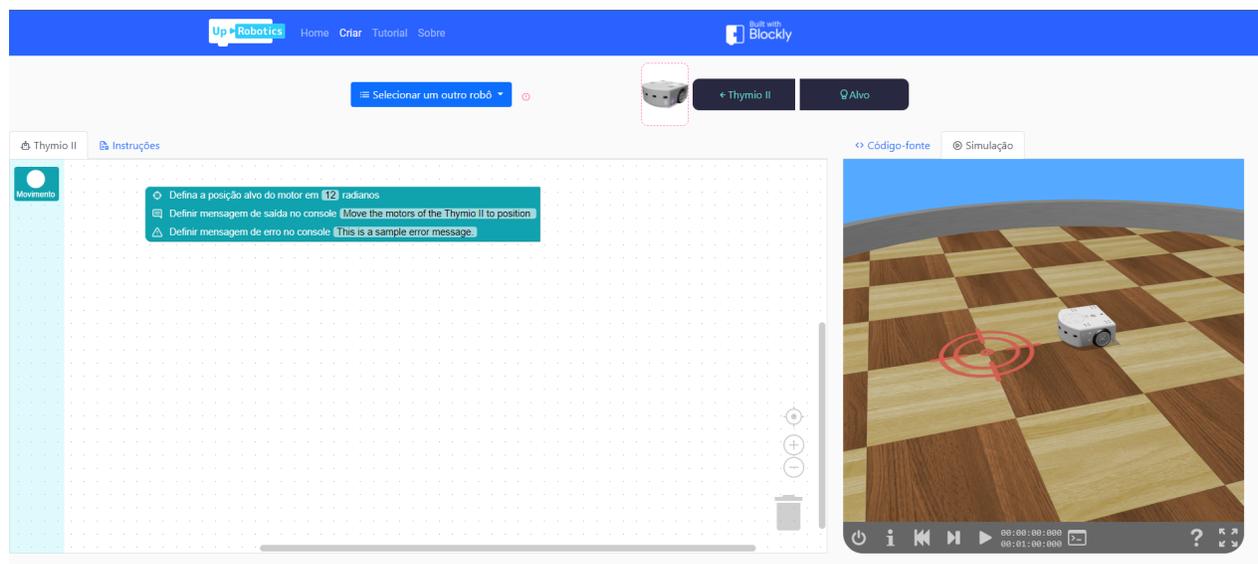


Figura 6.21: Exemplo do ThymioII na aba simulador do UpRobotics

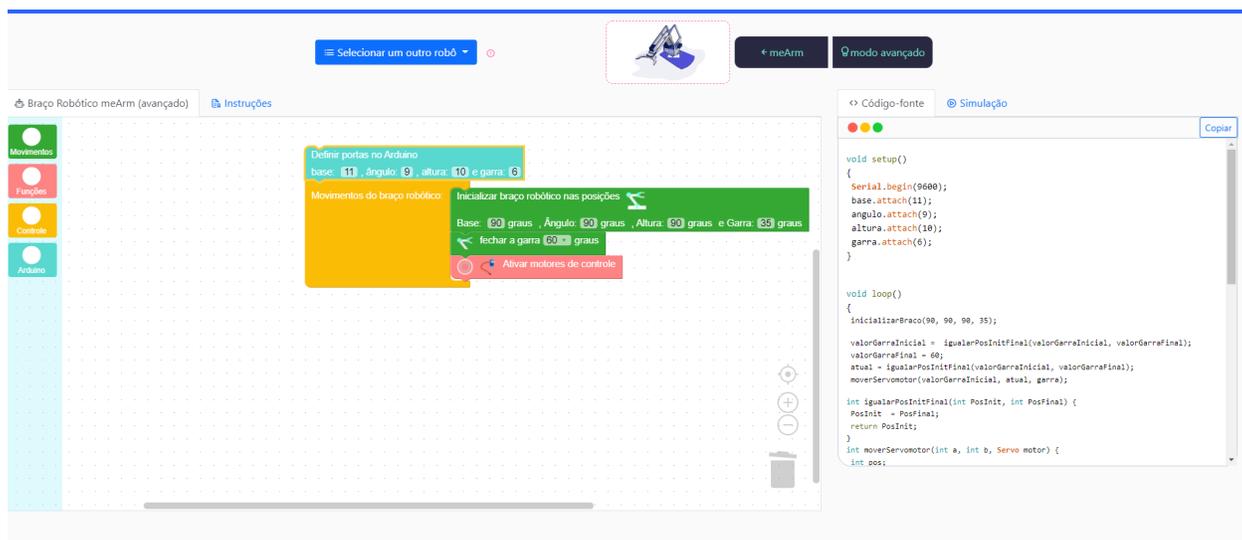


Figura 6.17: Exemplo de programação do braço no UpRobotics

#### 6.4.4 Robô Carrinho Pioneer 3D-X

O robô Pioneer 3-DX é uma base multifuncional, de duas rodas, utilizada para pesquisas e aplicações envolvendo mapeamento, teleoperação, localização, monitoramento, reconhecimento e outros comportamentos. Desta forma, esse carrinho robô está presente no UpRobotics e o objetivo do usuário é desenvolver um programa que controla esse robô para seguir uma parede, o segredo está em construir a sua lógica utilizando alguns de seus sensores. A Figura 6.22 mostra como esse robô é mostrado no simulador. A Figura 6.23 mostra um exemplo de programação desse robô no UpRobotics.



Figura 6.22: Pioneer 3D-X no cenário "Seguidor de Parede"

The screenshot displays the UpRobotics web interface. At the top, there is a navigation bar with the UpRobotics logo and links for Home, Criar, Tutorial, and Sobre. A 'Selecionar um outro robô' dropdown menu is present. The main workspace is split into two panels: a visual programming area on the left and a code editor on the right. The visual programming area shows a sequence of blocks for initializing the robot, enabling distance sensors, and implementing a wall-following logic with conditional blocks for distance thresholds and velocity adjustments. The code editor on the right shows the corresponding Python code for the wall-following benchmark.

```

Código-fonte
Copiar
"""Sample Webots controller for the wall following benchmark."""

from controller import Robot

def getDistance(sensor):
    """
    Return the distance of an obstacle for a sensor.

    The value returned by the getValue() method of the distance sensors
    corresponds to a physical value (here we have a sonar, so it is the
    strength of the sonar ray). This function makes a conversion to a
    distance value in meters.
    """
    return ((1000 - sensor.getValue()) / 1000) * 5

# Maximum speed for the velocity value of the wheels.
# Don't change this value.
MAX_SPEED = 5.24

# Get and enable the distance sensors.
frontSensor = robot.getDistanceSensor("son3")
  
```

Figura 6.23: Exemplo de programação do Pioneer 3D-X

## 6.5 Resumo do Capítulo

Este capítulo apresentou o processo de concepção de uma linguagem visual, através da modelagem formalizada e representada por um autômato finito determinístico, e como estava primeira versão da ferramenta com os blocos e respectivos códigos associados, essa primeira versão foi utilizada em um estudo de viabilidade com um robô educacional do tipo braço robótico, onde foram obtidos resultados e discussões a respeito do estudo de caso. Além disso, são apresentados os aprimoramentos da ferramenta como, por exemplo, a inclusão de mais alguns robôs, linguagens visuais e simuladores com ambiente 3D.

# Capítulo 7

## Experimentos e Resultados Obtidos

Este capítulo apresenta os resultados e discussões a respeito do estudo de viabilidade da ferramenta quando ainda estava em suas primeiras fases de desenvolvimento. A partir disso, foi possível identificar alguns possíveis aprimoramentos e melhorias que foram acrescentados na ferramenta e, posteriormente foi feito um experimento com uma turma de alunos em uma escola, onde houve a aplicação da avaliação de usabilidade da ferramenta, e diante disso foram obtidos alguns resultados e discussões apresentados nesse capítulo.

### 7.1 O Estudo de Viabilidade

Um estudo de viabilidade foi feito com o intuito validar a primeira versão entregável da ferramenta, verificando se código gerado possibilita a realização dos mesmos movimentos utilizados com o uso de dois *joysticks*. Deste modo, foram convidados dois participantes caracterizados como o público alvo (alunos de 11 anos de idade, do 7º ano), para testar a ferramenta.

Antes da realização desse estudo de viabilidade com os alunos, foi realizado dois experimentos com o braço robótico em laboratório. No primeiro experimento, o braço robótico deveria pegar um objeto de um lugar, girar e colocá-lo em outro lugar. No segundo experimento, ele deveria pegar um objeto de um lugar baixo, isto é, a superfície onde ele estava e colocar em cima de uma pilha de livros. Todos os passos para que ele fizesse as ações foram feitas por blocos interconectados na ferramenta proposta. O resultado foi promissor, pois ele fez exatamente como foi programado, inclusive respeitando a velocidade suave utilizada com *joystick*. Figura 7.2 mostra como funciona o controle do braço por intermédio de dois *joysticks*.



Figura 7.1: Braço robótico deixando um objeto em cima de uma pilha de livros.

### 7.1.1 O Nivelamento Teórico

Visando fornecer um nivelamento técnico para os dois participantes do experimento, foi lecionado uma aula, que durou cerca de 45 minutos, com o intuito de familiarizar os alunos com o conteúdo de Ângulos Matemáticos. Foi utilizado um slide com alguns exercícios simples ao final da apresentação e foram conduzidos alguns exercícios baseados nos livros distribuídos pelo governo exatamente sobre ângulos.

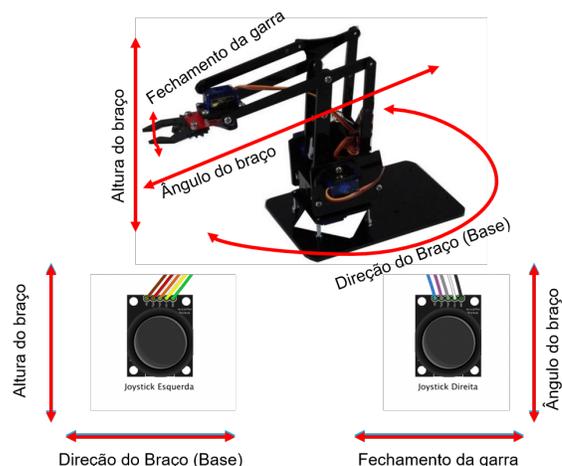


Figura 7.2: Relação entre *Joysticks* controladores e movimentos do braço.

Durante essas aulas foram apresentados alguns exemplos das aplicações práticas desses conhecimentos, como por exemplo, os braços robóticos utilizados pela NASA para explorar o espaço. Além disso, foi explicado como esses conceitos seriam utilizados no experimento para manipular o braço robótico.

### 7.1.2 A Demonstração da Ferramenta

Após a finalização do breve nivelamento, foi mostrado como é o funcionamento do braço quando se utiliza dois *joysticks*. Em seguida, foi apresentada a ferramenta e como os blocos disponíveis poderiam ser utilizados para realizar exatamente os mesmos movimentos, mas sem o auxílio de *joysticks*, apenas utilizando o conceito de ângulos matemáticos.

Tabela 7.1: Lista de passos utilizados no experimento

Passos	Movimentos a serem feitos
1	inicializar robô com a posição padrão definida no bloco
2	abrir garra 35 graus
3	mover altura para a posição 50 graus
4	mover ângulo para a posição 180 graus
5	fechar a garra 160 graus
6	mover ângulo para a posição 90 graus
7	girar para a direita 110 graus
8	abrir garra 35 graus
9	girar o braço esquerda 90 graus
10	mover a altura para a posição 90 graus

### 7.1.3 O Experimento Prático

Foram utilizados dois braços robóticos iguais e de baixo custo no experimento. A diferença é que o primeiro podia ser controlado por dois *joysticks* e o segundo poderia ser programado utilizando a ferramenta UpRobotics, mas sem o auxílio de *joysticks*. Deste modo, foi dado o mesmo desafio para ambos os participantes.

Desafio: Na mesa onde estavam os robôs existia um pequeno objeto, fácil de ser agarrado pela garra robótica. Assim sendo, cada participante teve que realizar um conjunto de movimentos com o braço primeiramente utilizando *joysticks* e o objeto da mesa. Após fazer isso, o participante teria que fazer exatamente os mesmos movimentos no robô sem *joysticks* programando-o com a LVBB do UpRobotics e utilizando o mesmo cenário do experimento com *joysticks*.

Deste modo, cada um dos participantes realizou o experimento sem o auxílio de qualquer pessoa. Assim sendo, como base, foi apresentado a eles uma lista com 10 movimentos necessários para o braço carregar um objeto de uma posição A até uma posição B. A Tabela 7.1 é a lista de movimentos utilizada no experimento.

## 7.2 Resultados e Discussões do Estudo de Viabilidade

A métrica que decide se o experimento obteve resultados nesse cenário positivos ou negativos é baseada em um método de avaliação utilizado na obra de Jurafsky e Martin (2018), no que tange a definição de precisão geral do modelo em Aprendizagem de Máquina. Nesse caso, avaliamos a precisão dos movimentos que usaram apenas a linguagem visual do UpRobotics em relação aos movimentos realizados pelo braço robótico utilizando *joysticks*. A precisão foca nos valores classificados como positivos:

$$prec = \frac{VP}{VP + VN} \quad (7.1)$$

Adaptando essa definição para o nosso problema, temos:

- VP (Verdadeiros Positivos): Movimentos produzidos do braço robótico exatamente como deveria acontecer se estivesse utilizando dois *joysticks*.
- VN (Verdadeiros Negativos): Movimentos produzidos do braço robótico diferente em relação à mesma tarefa utilizando dois *joysticks*.

O primeiro participante obteve 8 VPs e o segundo obteve 9 VPs. Temos então:

$$prec = \frac{17}{17 + 3} \quad (7.2)$$

Deste modo, nesse primeiro momento a precisão de 0.85 nesse cenário realista é considerada ótima. Deste modo, o resultado para esse experimento foi considerado aceitável e então essa primeira versão da ferramenta tem grande potencial para apoiar o ensino-aprendizagem de matemática, no que se refere a Ângulos, utilizando a RE, podendo contribuir com as escolas do ensino básico. De modo geral,

foi possível realizar a maioria das tarefas definidas, obedecendo quase tudo o que foi programado através dos blocos. Foi possível perceber também que os blocos são fáceis de manusear, o que é uma característica bastante aceita por estudantes, comparado às construções de uma linguagem de programação convencional. Algumas características se destacam na ferramenta, como a rapidez na realização de todas as suas funcionalidades e principalmente na geração automática de códigos e correteude comparado com o método tradicional.

Há melhorias que podem ser feitas no bloco de abertura/fechamento da garra robótica, pois os alunos ainda tiveram um pouco de dificuldade em saber quantos graus seriam necessários para a garra encaixar perfeitamente no objeto, sem apertar muito ou afrouxar o fechamento, tal ação é muito mais fácil com a utilização de *joysticks*.

Além do mais, com o decorrer desse projeto, desde as pesquisas na fase de levantamento bibliográfico ao desenvolvimento da ferramenta proposta, foi notório perceber o poder que as linguagens visuais possuem em agregar inovação dentro da educação, ensino e aprendizagem, principalmente no contexto da internet das coisas. Partindo desse mesmo princípio, assim como foi possível gerar código para um microcontrolador Arduino UNO, é possível também programar outros componentes como por exemplo um Raspberry Pi ou ESP8266.

### 7.3 O Experimento Prático em Sala de Aula

O experimento foi realizado na Escola de Robótica, Programação e Games para Crianças e Adolescentes - Manaós Tech, na cidade de Manaus, estado do Amazonas. O público alvo foram alunos do 3<sup>o</sup> ao 6<sup>o</sup> ano, com idade entre 7 a 11 anos. Visando fornecer um nivelamento técnico para foi elaborado um plano de ensino, cujo obje-

tivo foi desenvolver o pensamento computacional dos alunos para que eles pudessem resolver problemas do mundo real, utilizando RE e LVBB, aplicando conceitos de matemática básica na resolução de problemas. A Figura 7.3 mostra o UpRobotics em alguns notebooks e a Figura 7.4 mostra realizando uma atividade com o UpRobotics. Ao todo, participaram da pesquisa 5 alunos e cada aluno ficou com um notebook. A escola cedeu os equipamentos e disponibilizou um profissional de educação para auxiliar durante as aulas. O experimento também contou com a ajuda da coordenadora pedagógica da escola.

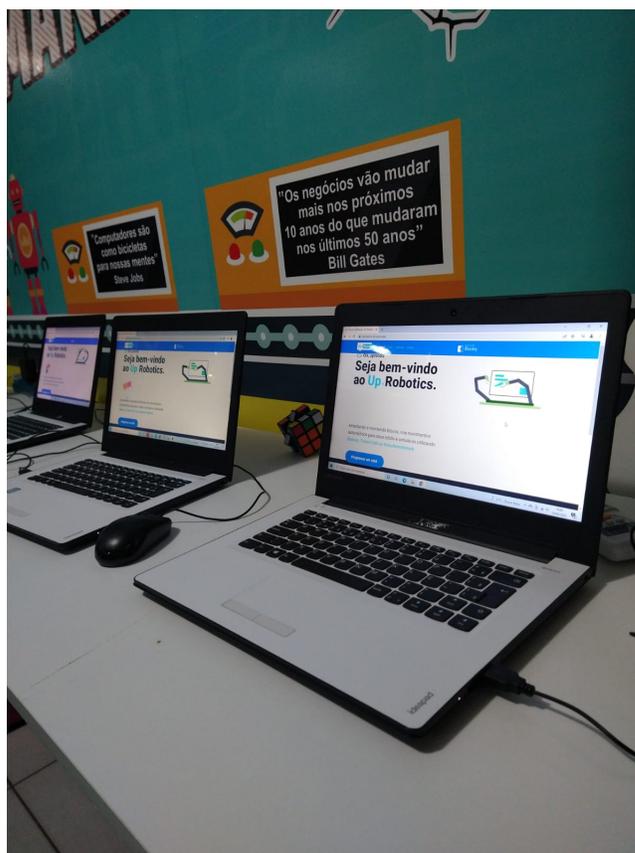


Figura 7.3: Máquinas com acesso ao UpRobotics preparadas para o experimento

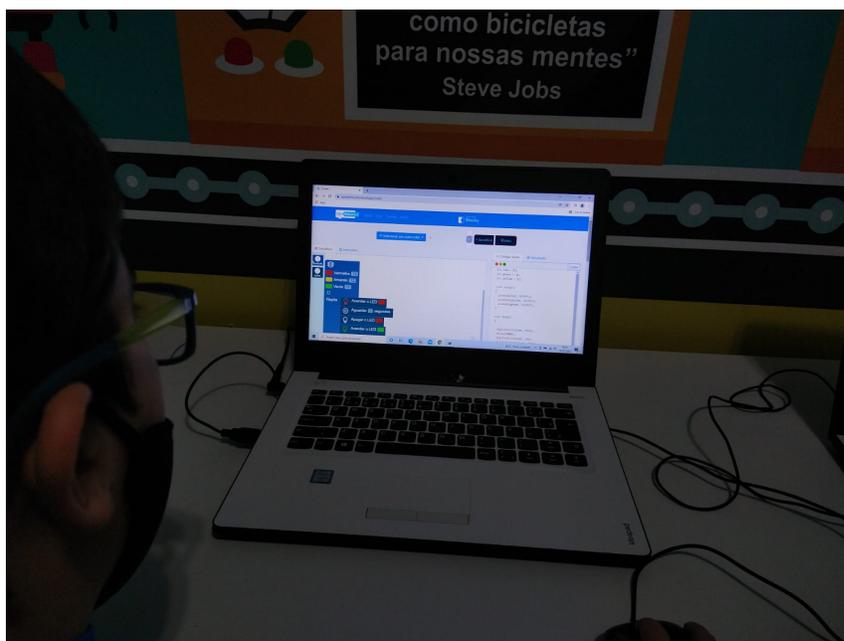


Figura 7.4: Aluno programando um semáforo

Foram elaborados 2 planos de aula, onde a aula 01 teve como principal objetivo permitir que os alunos aplicassem os conceitos de grandezas e medidas em um contexto real com o auxílio de RE, e para essa aula foi utilizado o cenário/ambiente "Semáforo". Já a aula 02 teve o objetivo de permitir que os alunos aplicassem conceitos de ângulos matemáticos em contextos reais com o auxílio de RE, para essa aula foram utilizados os robôs: braço robótico e Thymio II. Durante essas aulas foram levantados algumas discussões e questionamentos, onde foi possível utilizar alguns exemplos das aplicações práticas desses conhecimentos, como por exemplo, os braços robóticos utilizados pela NASA para explorar os planetas, e até sobre como esses conhecimentos podem ser utilizados em robôs da saga Star Wars, como o R2-D2, BB-8 e C3-PO, o que chamou bastante a atenção dos alunos. A Figura 7.5 mostra alguns alunos observando uma explicação sobre o funcionamento do UpRobotics.



Figura 7.5: Alunos observando uma explicação sobre o funcionamento de uma semáforo no UpRobotics

O detalhamento do plano de ensino e dos planos de aula, com as devidas especificações e descrições de todas as atividades realizadas, bem como dos conteúdos abordados, materiais adotados, desenvolvimento metodológico e itens de avaliação de aprendizado utilizados, encontram-se nos apêndices A e B.

### 7.3.1 Coleta de dados e Avaliação

#### Definição da Instrumentação

A avaliação de usabilidade tem o objetivo de verificar o desempenho de um software no que se refere a sua eficiência e eficácia quanto a interface homem-computador, tendo em vista a obtenção de indícios que demonstrem o grau de satisfação do usuário, identificando também possíveis problemas de usabilidade durante a realização de

atividades para que os mesmos sejam posteriormente corrigidos.

Com o intuito de utilizar um método de avaliação de usabilidade mais acessível ao público infantil, o formulário foi inspirado na Fun Toolkit que segundo 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. O Fun Toolkit é formado por um conjunto de quatro ferramentas, dentre as quais está o Smileyometer (Sorrisômetro): um tipo especial de escala Likert na qual os “itens de classificação” são acompanhados de smilyes, conforme a Figura 7.6.

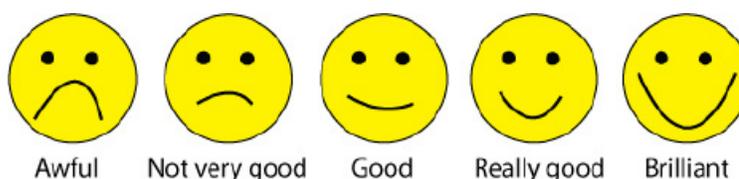


Figura 7.6: Smileyometer

Deste modo, para este trabalho, essa escala foi adaptada da seguinte forma: A Figura 7.7 mostra como fica essa escala para respostas que vão de "Muito fácil" até "Muito difícil". A Figura 7.8 mostra como fica essa escala para respostas que vão de "Muito legal" até "Ruim". A Figura 7.9 mostra como fica essa escala para respostas que vão de "Sim" até "Não".

Foi utilizado um questionário de avaliação foi dividido em duas etapas: (i) Questionário de Caracterização dos Participantes (Figura 7.11), (ii) Questionário de usabilidade e funcionalidade (Apêndice C). Antes da aplicação dos questionários, os pais ou responsáveis dos participantes tiveram que assinar um termo de consentimento livre e esclarecido (Figura 7.10).



Figura 7.7: Fácil ou Difícil

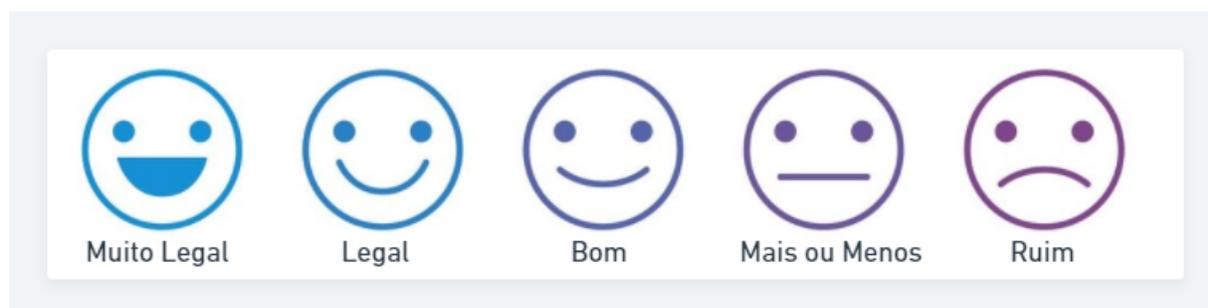


Figura 7.8: Legal ou Ruim



Figura 7.9: Sim ou Não

## 7.4 Resultados e Discussões

A respeito do perfil dos participantes, conforme mostrado na Figura 7.12, 40% tem 11 anos de idade, 20% tem 7 anos, 20% tem 8 anos e 20% tem 9 anos de idade. Esses alunos estão no 3<sup>o</sup>, 4<sup>o</sup>, 5<sup>o</sup> e 6<sup>o</sup> ano. Também, conforme mostrado nos demais resul-



### TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO

Avaliação do Software "UpRobotics: ferramenta que possibilita o uso de robótica desplugada utilizando linguagens visuais baseadas em blocos".

Prezados, pais ou responsáveis, meu nome é Francisco Gabriel Teixeira Marinho, sou aluno da Universidade Federal do Amazonas e curso Mestrado em Informática. Estou realizando uma pesquisa como resultado da minha dissertação "UpRobotics: Robótica Educacional Utilizando Linguagem Visual Baseada em Blocos", sob orientação do professor Dr. Raimundo da Silva Barreto. O objetivo desse estudo é aplicar linguagens visuais baseadas em blocos no contexto de robótica educacional tendo em vista exercitar o pensamento computacional por meio de atividades plugadas e desplugadas.

A participação do(a) aluno(a) envolve utilizar o UpRobotics durante as aulas e ao final responder um questionário para avaliar esse software. O processo de responder o questionário dura cerca de 15 minutos. A participação do aluno é muito importante para a conclusão desse trabalho.

A participação do(a) estudante é voluntária e ele(a) poderá desistir a qualquer momento. Na publicação dos resultados desta pesquisa a identidade dos participantes será mantida no mais rigoroso sigilo. Serão omitidas todas as informações que permitem identificar o(a) estudante. Se você autoriza a participação do(a) aluno(a) pelo(a) qual você é responsável, aceite esse termo na opção abaixo.

( ) Autorizo a participação do estudante nesse estudo

Em: \_\_\_\_/\_\_\_\_/\_\_\_\_ Assinatura: \_\_\_\_\_

Figura 7.10: Termo de Consentimento Livre e Esclarecido

tados presentes no Apêndice D, 80% nunca tinham conhecimento sobre Grandezas e Medidas, 60% não tinham conhecimento sobre ângulos.

Após o lecionamentos das aulas e exercícios práticos, devidamente planejados no planos de aula, os alunos responderam os questionários de usabilidade e funcionalidade (Conforme mostrado na Figura 7.13). Esses dados foram tabulados com o auxílio da ferramenta Google Forms que gera os gráficos de forma automática,

## I. Caracterização do participante

1. Qual é o seu nome?
2. Quantos anos você tem?
3. Em que série você está?
4. Você já ouviu falar sobre Grandezas e Medidas? ( ) Sim ( ) Não
5. Seu nível de conhecimento sobre Grandezas e Medidas? ( ) Sei ( ) Sei um pouco ( ) Não sei
6. Você já ouviu falar sobre Ângulos? ( ) Sim ( ) Não
7. Seu nível de conhecimento sobre Ângulos? ( ) Sei ( ) Sei um pouco ( ) Não sei
8. Você já ouviu falar sobre Pensamento Computacional? ( ) Sim ( ) Não
9. Você já ouviu falar sobre Robótica? ( ) Sim ( ) Não
10. Você já ouviu falar sobre Robótica Educacional? ( ) Sim ( ) Não
11. Você já ouviu falar sobre Robótica Desplugada? ( ) Sim ( ) Não
12. Você já ouviu falar sobre Internet das Coisas? ( ) Sim ( ) Não

Figura 7.11: Questionário de Caracterização dos Participantes

para se ter uma visualização do resultado da pesquisa de forma mais detalhada e representado em gráficos, veja o Apêndice D, os dados tabulados estão disponíveis em forma de gráficos. Devido o grande número de perguntas, aqui nesta seção o panorama geral das respostas serão mostrados em tabelas. Desta forma, foram obtidos os seguintes resultados:

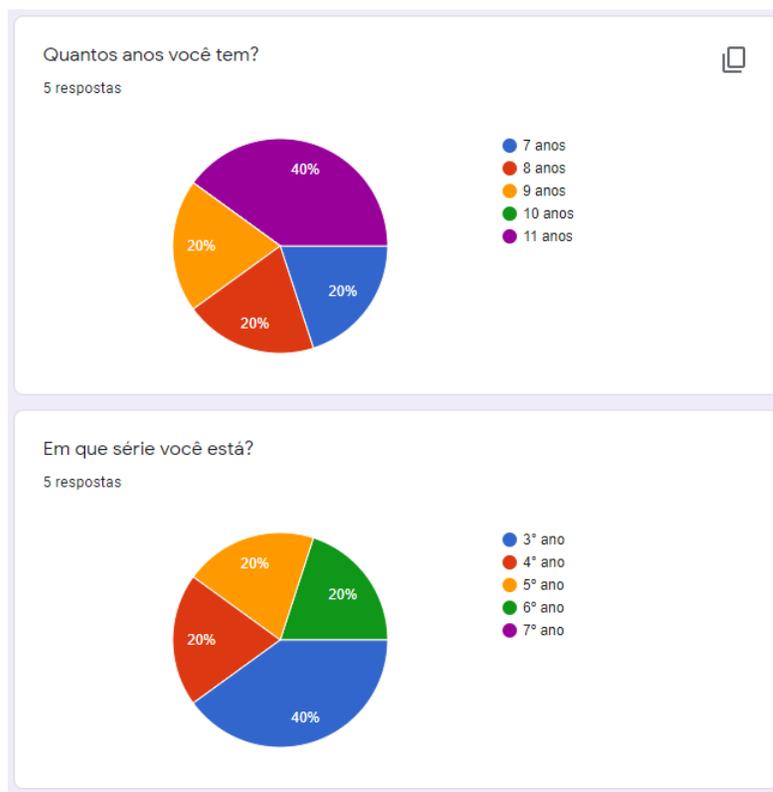


Figura 7.12: Faixa etária e nível escolar dos participantes

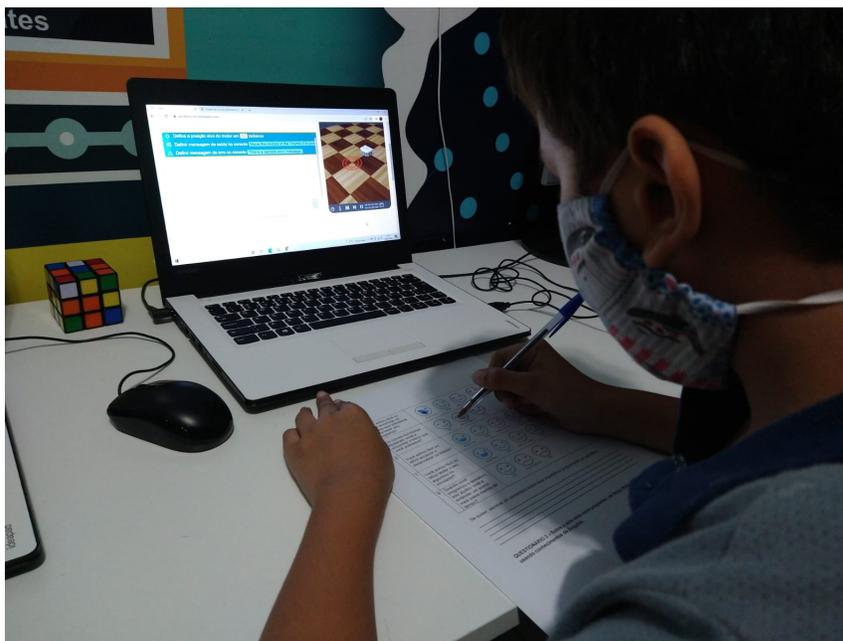


Figura 7.13: Aluno respondendo o questionário

A Tabela 7.2 mostra o resultado obtido do primeiro experimento, o que está relacionado com a aula sobre grandezas e medidas e a aplicação prática num semáforo.

Tabela 7.2: Porcentagens de respostas relacionadas ao experimento com o semáforo

Nº	Pergunta	Resposta				
		Muito legal (80%)	Legal (20%)	Bom (20%)	Mais ou menos	Ruim
1	O que você achou sobre a aula de Grandezas e Medidas?	Muito legal (80%)	Legal (20%)	Bom (20%)	Mais ou menos	Ruim
2	O que você achou de programar um semáforo com o UpRobotics?	Muito legal (80%)	Legal (20%)	Bom	Mais ou menos	Ruim
3	O que você achou de ter aplicado seu conhecimento sobre grandezas e medidas para programar o semáforo?	Muito legal (40%)	Legal (60%)	Bom	Mais ou menos	Ruim
4	Você achou fácil ou difícil encontrar os blocos que precisava para fazer seu programa?	Muito fácil (40%)	Fácil	Mais ou menos (40%)	Difícil (20%)	Muito difícil
5	Os ícones dos blocos ajudaram você a encontrar os blocos que precisava para fazer seu programa?	Sim (60%)	Mais ou menos (40%)	Não		

Tabela 7.2 continuação da página anterior

Nº	Pergunta	Resposta				
		Muito fácil (20%)	Fácil (20%)	Mais ou menos (40%)	Difícil	Muito difícil (20%)
6	Você achou fácil ou difícil encaixar e desencaixar os blocos?	Muito fácil (20%)	Fácil (20%)	Mais ou menos (40%)	Difícil	Muito difícil (20%)
7	Você achou fácil ou difícil testar o seu algoritmo no simulador?	Muito fácil	Fácil (80%)	Mais ou menos	Difícil	Muito difícil (20%)
8	Quando você programou o semáforo, isso ajudou você a entender um pouco mais sobre medida de tempo?	Sim (100%)	Mais ou menos	Não		

Esse resultado da Tabela 7.2 mostra primeiramente que os alunos gostaram das aulas sobre Grandezas e Medidas, e nesse caso, o foco principal foi medidas de tempo, o que lhes chamou atenção para pôr em prática seus conhecimentos para programar um semáforo. A maioria achou muito legal programar um semáforo. Percebe-se também que alguns tiveram dificuldade em encontrar os blocos que precisavam para fazer o programa, todavia a grande maioria achou que os ícones ajudaram a identificar os blocos certos. Sobre o encaixe e desencaixe dos blocos, o resultado foi muito distribuídos, alguns acharam fácil e outros acharam difícil, o que nos leva a pensar em uma solução melhor para trabalhos futuros. A grande maioria achou fácil testar o seu algoritmo no simulador e, todos os alunos afirmam que programar um semáforo ajudou a entender mais sobre a medida de tempo.

A Tabela 7.3 mostra o resultado obtido do segundo experimento, o que está

relacionado com a aula sobre ângulos e a aplicação prática num braço robótico.

Tabela 7.3: Porcentagem das respostas relacionados ao experimento com o braço robótico

Nº	Pergunta	Resposta				
1	O que você achou sobre a aula de Ângulos?	<b>Muito legal (100%)</b>	Legal	Bom	Mais ou menos	Ruim
2	O que você achou de programar um braço robótico com o UpRobotics?	<b>Muito legal (60%)</b>	<b>Legal (20%)</b>	<b>Bom (20%)</b>	Mais ou menos	Ruim
3	O que você achou de ter aplicado seu conhecimento sobre ângulos para programar o braço robótico?	<b>Muito legal (60%)</b>	<b>Legal (20%)</b>	<b>Bom (20%)</b>	Mais ou menos	Ruim
4	Você achou fácil ou difícil encontrar os blocos que precisava para fazer seu programa?	<b>Muito fácil (20%)</b>	<b>Fácil (20%)</b>	<b>Mais ou menos (60%)</b>	Difícil	Muito difícil
5	Os ícones dos blocos ajudaram você a encontrar os blocos que precisava para fazer seu programa?	<b>Sim (80%)</b>	<b>Mais ou menos (20%)</b>	Não		

**Tabela 7.3** continuação da página anterior

Nº	Pergunta	Resposta				
		Muito fácil (40%)	Fácil (40%)	Mais ou menos (20%)		Muito difícil
6	Você achou fácil ou difícil encaixar e desencaixar os blocos?	Muito fácil (40%)	Fácil (40%)	Mais ou menos (20%)		Muito difícil
7	Você achou fácil ou difícil testar o seu algoritmo no simulador?	Muito fácil	Fácil (40%)	Mais ou menos (60%)	Difícil	Muito difícil
8	Quando você programou o braço robótico, isso ajudou você a entender um pouco mais sobre ângulos?	Sim (60%)	Mais ou menos (40%)	Não		

Esse resultado da Tabela 7.3 mostra primeiramente que os alunos gostaram das aulas sobre Ângulos, pois todos responderam "Muito legal". Todos gostaram de programar o braço robótico e também gostaram de ter aplicado o conhecimento de ângulos para resolver o problema de exercício proposto, isto é, fazer um movimento para o ângulo, altura, base e garra, ou seja, para cada servomotor. 60% dos alunos acharam que o encontrar os blocos necessários "mais ou menos", ou seja, nem muito fácil, mas também não muito difícil. Além disso, 80% acham que os ícones ajudam "Mais ou menos" a encontrar os blocos certos, o que nos leva a pensar em uma solução melhor para trabalhos futuros, pois os ícones deveriam facilitar muito a vida do usuário nesse caso. A grande maioria achou nem muito fácil, nem muito difícil testar o seu algoritmo no simulador e, no entanto, todos os alunos disseram que aprenderam

de alguma forma um pouco mais sobre ângulos, o que é um ponto bastante positivo.

A Tabela 7.4 mostra o resultado obtido do ainda no segundo experimento, o que está relacionado com a aula sobre ângulos e a aplicação prática num carrinho Thymio II do simulador 3D virtual.

Tabela 7.4: Porcentagem das respostas relacionados ao experimento com o Thymio II

Nº	Pergunta	Resposta				
		Muito legal (100%)	Legal	Bom	Mais ou menos	Ruim
1	O que você achou sobre a aula de Ângulos?	Muito legal (100%)	Legal	Bom	Mais ou menos	Ruim
2	O que você achou de programar o Thymio II com oUpRobotics?	Muito legal (60%)	Legal (20%)	Bom	Mais ou menos (20%)	Ruim
3	O que você achou de ter aplicado seu conhecimento sobre ângulos para movimentar o Thymio II?	Muito legal (60%)	Legal (20%)	Bom (20%)	Mais ou menos	Ruim
4	Você achou fácil ou difícil encontrar os blocos que precisava para fazer seu programa?	Muito fácil (20%)	Fácil	Mais ou menos (80%)	Difícil	Muito difícil

Tabela 7.4 continuação da página anterior

Nº	Pergunta	Resposta				
5	Os ícones dos blocos ajudaram você a encontrar os blocos que precisava para fazer seu programa?	<b>Sim</b> (40%)	<b>Mais ou menos</b> (60%)	Não		
6	Você achou fácil ou difícil encaixar e desencaixar os blocos?	<b>Muito fácil</b> (60%)	<b>Fácil</b> (20%)	<b>Mais ou menos</b> (20%)		Muito difícil
7	Você achou fácil ou difícil testar o seu algoritmo no simulador?	<b>Muito fácil</b> (40%)	<b>Fácil</b> (20%)	<b>Mais ou menos</b> (40%)	Difícil	Muito difícil
8	Quando você programou o Thymio II, isso ajudou você a entender um pouco mais sobre ângulos?	<b>Sim</b> (60%)	<b>Mais ou menos</b> (40%)	Não		

Esse resultado da Tabela 7.4 mostra primeiramente que os alunos gostaram de aplicar o conhecimento de ângulos para movimentar o robô Thymio II. A maioria gostou de programar o Thymio II e também gostaram de ter aplicado o conhecimento de ângulos para resolver o problema de exercício proposto, e além disso, todos aprenderam de alguma forma um pouco mais sobre ângulos com esse robô. Porém há alguns pontos de melhoria a serem feitos como trabalhos futuros, por exemplo, os

ícones ainda não estão muito claros como deveriam para os alunos, e embora haja apenas um bloco para programar esse robô, percebe-se que alguns alunos ficaram em dúvida de como encontrá-lo ou usá-lo.

A Tabela 7.5 mostra o resultado obtido ao final da aula, que mostra uma percepção do aluno perante a ferramenta UpRobotics.

Tabela 7.5: Porcentagem das respostas relacionados à percepção dos alunos diante das funcionalidades do UpRobotics

Nº	Pergunta	Resposta				
1	O que você achou da ferramenta UpRobotics?	<b>Muito legal</b> (40%)	<b>Legal</b> (60%)	Bom	Mais ou menos	Ruim
2	O UpRobotics é fácil ou difícil de usar?	<b>Muito fácil</b> (20%)	<b>Fácil</b> (20%)	<b>Mais ou menos</b> (60%)	Difícil	Muito difícil
3	Você acha fácil ou difícil encontrar os blocos que precisa para fazer seu programa?	<b>Muito fácil</b> (20%)	<b>Fácil</b> (20%)	<b>Mais ou menos</b> (60%)	Difícil	Muito difícil
4	Você achou fácil ou difícil encontrar os blocos que precisava para fazer seu programa?	<b>Sim</b> (40%)	<b>Mais ou menos</b> (60%)	Não		

Tabela 7.5 continuação da página anterior

Nº	Pergunta	Resposta				
5	Os desenhos que têm nos blocos ajudam você a encontrar o bloco que você está procurando?	<b>Sim</b> (40%)	<b>Mais ou menos</b> (60%)	Não		
6	Você achou fácil ou difícil selecionar o robô desejado?	<b>Muito fácil</b> (60%)	Fácil	<b>Mais ou menos</b> (40%)	Difícil	Muito difícil
7	Você achou fácil ou difícil copiar o seu código?	<b>Muito fácil</b> (60%)	Fácil	<b>Mais ou menos</b> (40%)	Difícil	Muito difícil
8	Você achou fácil ou difícil testar o seu código no simulador?	<b>Muito fácil</b> (40%)	<b>Fácil</b> (20%)	<b>Mais ou menos</b> (40%)	Difícil	Muito difícil
9	Você achou fácil ou difícil acessar os tutoriais no UpRobotics?	<b>Muito fácil</b> (40%)	<b>Fácil</b> (20%)	<b>Mais ou menos</b> (40%)	Difícil	Muito difícil
10	Você achou fácil ou difícil encontrar as instruções para programar os robôs no UpRobotics?	<b>Muito fácil</b> (80%)	<b>Fácil</b> (20%)	Mais ou menos	Difícil	Muito difícil
11	Você gostaria de usar o UpRobotics em outras aulas?	<b>Sim</b> (80%)	Mais ou menos	<b>Não</b> (20%)		

Tabela 7.5 continuação da página anterior

Nº	Pergunta	Resposta				
12	Você gostaria de programar outros robôs no UpRobotics?	<b>Sim</b> <b>(60%)</b>	<b>Mais ou menos</b> <b>(20%)</b>	<b>Não</b> <b>(20%)</b>		

Esses resultados da Tabela 7.5 são bastante promissores, pois todos os alunos gostaram da ferramenta, 80% gostaria de usar em outras aulas, 60% gostaria de programar outro robô na ferramenta. Ninguém achou difícil testar o código ou acessar os tutoriais ou instruções do UpRobotics. Porém há alguns pontos de melhoria a serem feitos como trabalhos futuros, por exemplo, os ícones ainda não estão muito claros como deveriam para os alunos, isso foi perceptível em todos os resultados, ainda é um desafio aprimorar a ferramenta no que tange à simulação 3D com robotbenchmark.

## 7.5 Resumo do Capítulo

Esse Capítulo apresentou os resultados obtidos com estudo de viabilidade, que envolveu etapas que envolvem desde o processo de concepção da linguagem (modelagem formalizada e representada em autômato, mostrada na Seção 6.1), até a análise dos primeiros resultados do estudo. Em seguida, apresentou um novo experimento feito em uma escola, a avaliação de usabilidade e funcionalidade da ferramenta, resultados obtidos e discussões onde foi possível observar, portanto, o grande potencial que essa pesquisa tem para o avanço da fronteira de conhecimento no que se refere a RE, geração automática de código para programação robótica e conceitos baseados na

robótica desplugada que podem ser aprimorados e aplicados na técnica envolvida na ferramenta.

# Capítulo 8

## Conclusão

### 8.1 Considerações Finais

O projeto UpRobotics foi desenvolvido com o intuito de aplicar linguagens visuais baseadas em blocos no contexto de RE tendo em vista exercitar o pensamento computacional por meio de atividades plugadas e desplugadas. Como resultados, nessa última versão da ferramenta, percebeu-se que a lógica de programação se tornou transparente para o usuário, pois a programação passou a ser a ligação adequada dos blocos funcionais.

O apelo visual e o encaixe correto das peças induzem o aluno a percepção das ligações corretas das partes da programação do início ao fim do programa, facilitando a manipulação dos elementos da lógica de programação. Diante dos resultados obtidos e analisados é possível afirmar que o UpRobotics cumpriu o seu objetivo principal, conseguiu servir de auxílio para repassar um conhecimento novo para os alunos. A sua grande maioria de robôs e ambiente podem ser testados de forma virtual, até mesmo o braço robótico pode ser testado no circuito do TinkerCAD. O

mesmo código gerado para o braço robótico e para o semáforo, vai funcionar corretamente no robô/ambiente real se funcionar corretamente no virtual. Isso torna o UpRobotics um artefato muito promissor no que tange a área de RE e pensamento computacional.

Os resultados mostraram ser promissores para a continuação do projeto, visto que esse mínimo produto viável já possibilita a geração automática de código embarcável em um Arduino, que é suficiente para o controle de movimentos do robô educacional. Deste modo, esse trabalho tem potencial para dar subsídios a um estudo mais aprofundado sobre ensino de robótica, dando base para experimentos de campo aplicados a um número maior de alunos.

## 8.2 Publicações

Esta pesquisa gerou as seguintes publicações:

- O artigo Marinho *et al.* (2020) foi publicado nos anais do Brazilian Symposium on Computing Systems Engineering (SBESC'2020), evento com Qualis B1.
- Foi ministrado um minicurso de 2 horas intitulado **Construção de uma Linguagem Visual Baseada em Blocos para Ensino de Robótica** na Escola de Sistemas Embarcados (ESSE) evento conjunto do SBESC'2020.
- Visando a disseminação de conhecimentos a respeito da construção da linguagem visual baseada em blocos, foi disponibilizado em Marinho (2020) uma playlist de vídeos dividida em 7 aulas: (i) Conhecendo o Google Blockly, (ii) Primeiros Passos com Blockly Developer Tools, (iii) Criando um Projeto, (iv) Blocos Personalizados, (v) Exportando Blocos, (vi) Geração Automática de Códigos e (vii) Ferramenta Pronta. Essas aulas mostram um pouco do processo

inicial de criação do UpRobotics em sua versão mais básica. É ressaltado nessas aulas a importância de como criar os excaixes deixando-os mais intuitivos para que o usuário evite cometer erros semânticos durante a programação.

### 8.3 Limitações

As melhorias implementadas na versão atual do UpRobotics resolveram algumas limitações na versão anterior, porém ainda há algumas possíveis melhorias a serem implementadas nas futuras versões. Também há limitações nos experimentos:

- **Componentes Físicos:** No que tange ao braço robótico, embora o UpRobotics utilizasse conceitos de robótica desplugada nas características da linguagem, era possível validar o código gerado apenas por intermédio de componentes reais, ou seja, Arduino e Braço robótico físicos. Para resolver esse problema, foi acrescentado componentes virtuais que simulam os movimentos dos motores no TinkerCAD, de acordo com os blocos montados pelo aluno, porém no TinkerCAD o aluno tem uma visão real de como funcionam os servomotores, mas não de como funciona o braço robótico por completo.
- **Simulador 3D:** A adição do ambiente 3D (robotbenchmark) no UpRobotics foi um ponto bastante positivo no que tange a testar o algoritmo virtualmente dentro do próprio UpRobotics, todavia, o processo de copiar o código e embarcar dentro do robô para iniciá-lo novamente é um pouco custoso para o usuário. A ideia para resolver isso seria criar um único botão que automatizasse todo esse processo.
- **Experimento:** O experimento foi feito numa escola particular voltada para a área de robótica, todavia os resultados poderiam ser diferentes em uma escola

pública, com estudantes da mesma idade, sem nenhum conhecimento de pensamento computacional. Além do mais, o experimento foi limitado a estudantes com idade de 7 a 11 anos, talvez, poderíamos obter resultados diferentes com estudantes com idade de 12 a 17 anos, por exemplo. Deste modo, isso é algo a ser explorado em trabalhos futuros.

- **UpRobotics:** Para se utilizar a ferramenta, é recomendável que o estudante tenha alguns conhecimentos prévios de matemática básica, tendo noções de grandezas e medidas de tempo, ângulos e noções básicas de lógica. Esses são os pré-requisitos para se utilizar a versão atual da ferramenta da melhor forma possível.

## 8.4 Trabalhos Futuros

De acordo com os resultados preliminares, vimos que a robótica plugada está implementada, pois é possível movimentar um robô por meio de código-fonte gerado de acordo com os blocos conectados. No caso do braço robótico e do semáforo, o mesmo código gerado para o simulador TinkerCAD, funciona no ambiente real com um circuito similar. O próximo passo será incluir novos robôs (com movimentos mais complexos), além de aprimoramentos na funcionalidade de testar o algoritmo nos ambientes de simulação 3D. Isso será de suma importância para ampliar o público alvo da ferramenta, no caso, para turmas do ensino médio e ensino superior.

No que tange as LVBBs atuais, o modo de como elas foram construídas ajuda o usuário a não cometer alguns equívocos semânticos ou sintáticos em seu algoritmo, isso acontece devido ao modo de como os encaixes foram projetados respeitando esses fatores, como por exemplo, blocos de "loop"(comportamento automático) só englo-

bam blocos de movimentos, e blocos de movimentos foram projetados para serem inseridos dentro de blocos de "loop". Deste modo, como trabalhos futuros, essas linguagens poderão ser ampliadas e aprimoradas levando em consideração análises de corretude e completude, além de utilização de técnicas dedutivas.

No que tange ao potencial da ferramenta para trabalhos futuros, com a flexibilidade de controlar uma placa com o Arduino usando os blocos funcionais arrastáveis, a ferramenta tem potencial de expandir suas funcionalidades com a inclusão de outros componentes relacionados à matemática, internet das coisas e robótica. Por exemplo, pode haver a inclusão de sensores e atuadores, outros microcontroladores, conceitos de outras áreas da ciência e educação, poderá servir de base para a elaboração de outros experimentos cujos os resultados serão de suma importância para o avanço da fronteira de conhecimento nesse ramo científico.

# Referências Bibliográficas

- Baião, E. R., do Amaral, S. F. e Vizconde, E. (2017), Scratch for arduino e o ensino de corrente elétrica: Uma proposta para o ensino médio.
- Baratè, A., Formica, A., Ludovico, L. A. e Malchiodi, D. (2017), Fostering computational thinking in secondary school through music-an educational experience based on google blockly, *in* International Conference on Computer Supported Education, Vol. 2, SCITEPRESS, pp. 117–124.
- Barr, V. e Stephenson, C. (2011), Bringing computational thinking to k-12: what is involved and what is the role of the computer science education community?, *Acm Inroads* 2(1), 48–54.
- Basili, V. R., Caldiera, G. e Rombach, H. D. (2002), Experience factory, *Encyclopedia of software engineering* .
- Bell, T., Alexander, J., Freeman, I. e Grimley, M. (2009), Computer science unplugged: School students doing real computing without computers, *The New Zealand Journal of Applied Computing and Information Technology* 13(1), 20–29.
- Biolchini, J., Mian, P. G., Natali, A. C. C. e Travassos, G. H. (2005), Systematic

- review in software engineering, *System Engineering and Computer Science Department COPPE/UFRJ, Technical Report ES 679(05)*, 45.
- Boehm, B. W. (1988), A spiral model of software development and enhancement, *Computer* 21(5), 61–72.
- Bravo, F. A., González, A. M. e González, E. (2017), A review of intuitive robot programming environments for educational purposes, *in 2017 IEEE 3rd Colombian Conference on Automatic Control (CCAC)*, IEEE, pp. 1–6.
- Casali, A., Zanarini, D., Monjolat, N. e San Martín, P. (2018), Teaching and learning computer science for primary school teachers: an argentine experience, *in 2018 XIII Latin American Conference on Learning Technologies (LACLO)*, IEEE, pp. 349–355.
- Castilho, M. I. (2002), Robótica na educação: Com que objetivos, *Monografia de Especialização em Informática na Educação*. Porto Alegre: Universidade Federal do Rio Grande do Sul .
- Conde, M. Á., Fernández-Llamas, C., Rodríguez-Sedano, F. J., Guerrero-Higueras, Á. M., Matellán-Olivera, V. e García-Peñalvo, F. J. (2017), Promoting computational thinking in k-12 students by applying unplugged methods and robotics, *in Proceedings of the 5th International Conference on Technological Ecosystems for Enhancing Multiculturality*, ACM, p. 7.
- Cunha, F. e Nascimento, C. R. (2018), Uma abordagem baseada em robótica e computação desplugada para desenvolver o pensamento computacional na educação básica, *in Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*, Vol. 29, p. 1845.

- Federici, S., Sergi, E. e Gola, E. (2019), Easy prototyping of multimedia interactive educational tools for language learning based on block programming, pp. 140–153.
- Fracasso, N., da Silva, V., Martins, M., Salina, F. V., Cutigi, J. F. e de Oliveira, L. B. R. (2018), Análise do impacto da robótica educacional no desempenho e nas escolhas acadêmicas de alunos do ensino técnico integrado: Um estudo no escopo do ifsp são carlos, *in* 3<sup>o</sup> Workshop de Inovação, Pesquisa, Ensino e Extensão.
- Goud Yadagiri, R., Krishnamoorthy, S. e Kapila, V. (2017), A blocks-based visual environment to teach robot-programming to k-12 students, *Computers in Education Journal* 8, 24–32.
- Grover, S. e Basu, S. (2017), Measuring student learning in introductory block-based programming: Examining misconceptions of loops, variables, and boolean logic, *in* Proceedings of the 2017 ACM SIGCSE technical symposium on computer science education, ACM, pp. 267–272.
- Hopcroft, J. E., Ullman, J. D. e Motwani, R. (2002), Introdução à teoria de autômatos, linguagens e computação, *Rio de Janeiro: Campus* .
- Jagušt, T., Krzic, A. S., Gledec, G., Grgić, M. e Bojic, I. (2018), Exploring different unplugged game-like activities for teaching computational thinking, *in* 2018 IEEE Frontiers in Education Conference (FIE), IEEE, pp. 1–5.
- Jancheski, M. (2017), Improving teaching and learning computer programming in schools through educational software, *Olympiads in Informatics* 11, 55–75.
- João, P., Nuno, D., Fábio, S. F. e Ana, P. (2019), A cross-analysis of block-based and visual programming apps with computer science student-teachers, *Education Sciences* 9(3), 181.

- Jurafsky, D. e Martin, J. H. (2018), Speech and language processing (draft), *Chapter A: Hidden Markov Models (Draft of September 11, 2018)*. Retrieved March 19, 2019.
- Kim, S.-W. e Lee, Y. (2017), A study of educational method using app inventor for elementary computing education., *Journal of Theoretical & Applied Information Technology* 95(18).
- Kitchenham, B. (2004), Procedures for performing systematic reviews, *Keele, UK, Keele University* 33(2004), 1–26.
- Kitchenham, B. e Charters, S. (2007), Guidelines for performing systematic literature reviews in software engineering.
- Kloc, A. E., Koscianski, A. e Pilatti, L. A. (2009), Robótica: uma ferramenta pedagógica no campo da computação, *I Simpósio Nacional de Ensino de Ciência e Tecnologia, UTFPR* .
- Krishnamoorthy, S. P. e Kapila, V. (2016), Using a visual programming environment and custom robots to learn c programming and k-12 stem concepts, *in Proceedings of the 6th Annual Conference on Creativity and Fabrication in Education, ACM*, pp. 41–48.
- Mafra, S. N. e Travassos, G. H. (2006), Estudos primários e secundários apoiando a busca por evidência em engenharia de software, *Relatório Técnico, RT-ES* 687(06).
- Marinho, F. (2020), Construção de uma linguagem visual baseada em blocos. URL : <https://www.youtube.com/playlist?list=PLIE3otGP5p5ek1E3bPlphaan8RxRBo61d>

- Marinho, F., Monteiro, E. e Barreto, R. (2020), Urobotics: Robótica educacional utilizando linguagem visual baseada em blocos, *in* Brazilian Symposium on Computing Systems Engineering, SBC.
- May, S. (2017), *What Is Robotics?* URL : [https://www.nasa.gov/audience/forstudents/5-8/feature/nasa-knows/what\\_is\\_robotics\\_58.html](https://www.nasa.gov/audience/forstudents/5-8/feature/nasa-knows/what_is_robotics_58.html)
- Miller, B., Kirn, A., Anderson, M., Major, J. C., Feil-Seifer, D. e Jurkiewicz, M. (2018), Unplugged robotics to increase k-12 students' engineering interest and attitudes, *in* 2018 IEEE Frontiers in Education Conference (FIE), IEEE, pp. 1–5.
- Monk, S. (2013), *Programação com Arduino: começando com Sketches*, Bookman Editora.
- Nakamura, T. e Kawasaki, T. (2019), Computer science unplugged for developing computational thinking and mathematical thinking, *in* 2019 International Joint Conference on Information, Media and Engineering (IJCIME), IEEE, pp. 305–308.
- Neto, M., dos Santos, C. A. M., de Souza, E. E. e Fonseca, M. G. (2018), Robótica educacional uma ferramenta para ensino de lógica de programação no ensino fundamental, *in* Anais do Workshop de Informática na Escola, Vol. 24, p. 315.
- Pasternak, E., Fenichel, R. e Marshall, A. N. (2017), Tips for creating a block language with blockly, *in* 2017 IEEE Blocks and Beyond Workshop (B&B), IEEE, pp. 21–24.
- Piaget, J. e Merlone, M. (1976), *A equilibração das estruturas cognitivas: problema central do desenvolvimento*, Zahar.

- Pressman, R. e Maxim, B. (2016), *Engenharia de Software-8ª Edição*, McGraw Hill Brasil.
- Queiroz, R., Sampaio, F. F. e dos Santos, M. P. (2017), 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, *in Anais dos Workshops do Congresso Brasileiro de Informática na Educação*, Vol. 6, p. 25.
- Read, J. C. e MacFarlane, S. (2006), Using the fun toolkit and other survey methods to gather opinions in child computer interaction, *in Proceedings of the 2006 conference on Interaction design and children*, pp. 81–88.
- Resende, G., de Freitas, M. d. G. B. *et al.* (2013), Principais dificuldades percebidas no processo ensino-aprendizagem de matemática em escolas do município de divinópolis (mg) the mains difficulties looking of the process teaching-learning of mathematics in schools of the district of divinópolis, mg, *Educação Matemática Pesquisa: Revista do Programa de Estudos Pós-Graduados em Educação Matemática* 15(1).
- Rocha, J. G. (2018), Arquitetura em camadas com uso do paradigma mvc e processo unificado na programação de software orientado a objetos, *TECNOLOGIAS EM PROJEÇÃO* 9(1), 31–49.
- Rosales, V. A., Queiruga, C., Tzancoff, C. B., Kimura, I. M. e Bartneche, M. B. (2017), Programming competitions in high school classrooms: Rita en red, *in 2017 XLIII Latin American Computer Conference (CLEI)*, IEEE, pp. 1–8.
- Rose, S., Habgood, J. e Jay, T. (2018), Pirate plunder: game-based computational

- thinking using scratch blocks, Academic Conferences and Publishing International Limited.
- Sales, S. *et al.* (2017), Utilizando scratch e arduino como recursos para o ensino da matemática, *in* XXIII Workshop de Informática na Escola, Recife.
- Santos, C. F. R. d. (2016), A robótica educacional e seu potencial como ferramenta de explicitação de invariantes operatórios relacionados a conceitos matemáticos.
- Sommerville, I. (2011), Software engineering 9th edition, *ISBN-10* 137035152, 18.
- Sousa, B., Ripardo, D., Campos, I., Maciel, J., Santos, R., Romano, W., Raiol, A. A. e Bezerra, F. (2018), Robótica educacional e computação desplugada: Experiência em oficinas para calouros, *in* Anais dos Workshops do Congresso Brasileiro de Informática na Educação, Vol. 7, p. 540.
- Souza, I. M., Andrade, W. L., Sampaio, L. M. e Araujo, A. L. S. O. (2018), A systematic review on the use of lego® robotics in education, *in* 2018 IEEE Frontiers in Education Conference (FIE), IEEE, pp. 1–9.
- Souza, M. A. e Duarte, J. R. (2015), Low-cost educational robotics applied to physics teaching in brazil, *Physics Education* 50(4), 482.
- Swaid, S. e Suid, T. (2019), Computational thinking education: Who let the dog out?, *in* 2019 International Conference on Computational Science and Computational Intelligence (CSCI), IEEE, pp. 788–792.
- Vieira, A., Passos, O. e Barreto, R. (2013), Um relato de experiência do uso da técnica computação desplugada, *Anais do XXI WEI* pp. 670–679.

- Vinayakumar, R., Soman, K. e Menon, P. (2018a), Ct-blocks analyser: Analysing ct-blocks projects, *in* 2018 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT), IEEE, pp. 1–5.
- Vinayakumar, R., Soman, K. e Menon, P. (2018b), Digital storytelling using scratch: Engaging children towards digital storytelling, *in* 2018 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT), IEEE, pp. 1–6.
- Vinayakumar, R., Soman, K. e Menon, P. (2018c), Map-blocks: Playing with online data and infuse to think in a computational way, *in* 2018 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT), IEEE, pp. 1–6.
- Weigend, M., Vanicek, J., Pluhar, Z. e Pesek, I. (2019), Computational thinking education through creative unplugged activities, *Olympiads in Informatics* 13, 171–192.
- Weintrop, D., Hansen, A. K., Harlow, D. B. e Franklin, D. (2018), Starting from scratch: Outcomes of early computer science learning experiences and implications for what comes next, *in* Proceedings of the 2018 ACM Conference on International Computing Education Research, ACM, pp. 142–150.
- Weintrop, D. e Wilensky, U. (2019), Transitioning from introductory block-based and text-based environments to professional programming languages in high school computer science classrooms, *Computers & Education* 142, 103646.
- Wing, J. M. (2006), Computational thinking, *Communications of the ACM* 49(3), 33–35.

# Apêndice A

## Plano de Ensino



## PLANO DE ENSINO DE DISCIPLINA

**Professor(a):** Francisco Gabriel Teixeira Marinho

**Curso:** Pensamento Computacional e Robótica Desplugada

**Público:** Alunos de 7 a 11 anos

**Carga Horária Anual:** 2h

**Nº Aulas Semanais:** 1

**Turno:** Vespertino

**Ano:** 2022

### EMENTA

- Conceitos de Grandezas e Unidades de Medida
- Conceitos básicos de Ângulos e suas unidades de medida
- Noções de Internet das Coisas
- Noções de Robótica Educacional

### OBJETIVO GERAL

Desenvolver o pensamento computacional dos alunos para que eles possam resolver problemas do mundo real, utilizando robótica educacional e linguagem visual baseada em blocos, aplicando conceitos de matemática básica na resolução de problemas.

### OBJETIVOS ESPECÍFICOS

- Permitir que o aluno resolva problemas do dia a dia utilizando a lógica e conceitos de matemática
- Permitir que o aluno programe um robô para cumprir determinadas tarefas

### CONTEÚDO PROGRAMÁTICO

- Grandezas e Medidas: Comprimento, Área, Tempo, Ângulo e Volume.
- Ângulos: Graus e Radianos
- IoT: Sensores, Atuadores e Componentes de IoT.
- Robótica Educacional: Plugada e Desplugada

### METODOLOGIA DE ENSINO

Aulas expositivas, discussões e exercícios práticos

### RECURSOS DIDÁTICOS

- Ferramenta Educacional Online UpRobotics
- Um braço robótico educacional (**meArm**)
- LEDs e Sensores
- Protoboard e Jumpers
- Um Arduino UNO

### AVALIAÇÃO

- Aplicar o conhecimento de tempo e grandezas a um contexto real, nesse caso, um semáforo controlado por um Arduino Uno.

- Aplicar o conhecimento sobre Ângulos e medidas em um Robô Físico (Braço Robótico).
- Aplicar o conhecimento sobre Ângulos e medidas em um Robô Virtual (Thymioll)

#### BIBLIOGRAFIA BÁSICA

BRASIL. **Parâmetros Curriculares Nacionais: Matemática. 2.** ed Brasília: MEC/SEF, 1997.

GEOGEBRA. **Noções de grandezas físicas e unidades de medidas.** Disponível em: <https://www.geogebra.org/m/kv99m9ru>

GEOGEBRA. **Ângulos.** Disponível em: <https://www.geogebra.org/m/fztrdsfv>

#### TEXTOS COMPLEMENTARES

**Grandezas e Medidas.** Disponível em:

[http://webeduc.mec.gov.br/portaldoprofessor/matematica/condigital1/01\\_grandezas\\_medidas/conteudo01.html](http://webeduc.mec.gov.br/portaldoprofessor/matematica/condigital1/01_grandezas_medidas/conteudo01.html)

**Medidas de Tempo.** Disponível em:

<https://www.educamaisbrasil.com.br/enem/matematica/medidas-de-tempo>

**Matemática: Grandezas e Medidas.** Disponível em:

<https://meuartigo.brasilecola.uol.com.br/pedagogia/matematica-grandezas-medidas.htm>

**Medidas de Tempo.** Disponível em: <https://escolakids.uol.com.br/matematica/unidades-de-medida-de-tempo.htm>

**Ângulos.** Disponível em: <https://www.educamaisbrasil.com.br/enem/matematica/angulos>

**Ângulos.** Disponível em: <https://mundoeducacao.uol.com.br/matematica/angulos.htm>

**UpRobotics: Robótica Educacional Utilizando Linguagem Visual Baseada em Blocos.** Disponível em:

[https://sol.sbc.org.br/index.php/sbesc\\_estendido/article/view/13101/12954](https://sol.sbc.org.br/index.php/sbesc_estendido/article/view/13101/12954)

# Apêndice B

## Planos de Aula



UFAM

UNIVERSIDADE FEDERAL DO AMAZONAS  
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA  
INSTITUTO DE COMPUTAÇÃO

PLANO DE AULA

1. IDENTIFICAÇÃO

<b>Professor:</b> Francisco Gabriel Teixeira Marinho	<b>Disciplina:</b> Matemática Básica	<b>Tema:</b> Grandezas e medidas	<b>Data/Hora:</b> 19 de Fevereiro de 2022, 1 hora / aula.
---	---	-------------------------------------	--

2. PLANO

	OBJETIVOS	CONTEÚDOS	RECURSOS
<b>GERAL</b>	Aplicar os conceitos de grandezas e medidas em um contexto real com o auxílio de robótica educacional	<ul style="list-style-type: none"><li>✓ Cumprimento</li><li>✓ Área</li><li>✓ Tempo</li><li>✓ Volume.</li><li>✓ Conceitos básicos de IoT</li></ul>	<ul style="list-style-type: none"><li>✓ Projetor multimídia;</li><li>✓ Computador com <i>software</i> de apresentação e de exibição de vídeo;</li><li>✓ Ferramenta UpRobotics</li><li>✓ Arduino UNO virtual e físico</li></ul>
<b>ESPECÍFICOS</b>	<ul style="list-style-type: none"><li>✓ Aprender os tipos de grandezas e medidas</li><li>✓ Entender como essas grandezas estão ao nosso redor e como são aplicadas na tecnologia</li><li>✓ Saber aplicar esses conhecimentos para resolver um problema real</li><li>✓ Controlar um semáforo utilizando conceitos de unidades de medidas de tempo</li></ul>		

3. PROCEDIMENTOS

INTRODUÇÃO	DESENVOLVIMENTO	CONCLUSÃO
<ul style="list-style-type: none"><li>✓ Inicialmente, haverá uma aula dos conceitos sobre grandezas e medidas como pré-requisitos para o desenvolvimento</li></ul>	<ul style="list-style-type: none"><li>✓ Após a aula, será apresentada uma situação-problema cuja solução irá construir o conceito de unidade de medidas</li><li>✓ Na sequência, será apresentada um pequeno vídeo sobre grandezas e medidas da plataforma Geogebra;</li><li>✓ Será enfatizado o assunto medidas de tempo;</li><li>✓ Será exibido um vídeo com uma reportagem sobre como funcionam os semáforos;</li></ul>	<ul style="list-style-type: none"><li>✓ A aula terminará com os alunos respondendo um formulário de avaliação de usabilidade e funcionalidade;</li></ul>

da aula.	<ul style="list-style-type: none"><li>✓ Os alunos serão convidados a criar grupos para resolverem em equipe um problema;</li><li>✓ O problema dado a eles será a construção de um algoritmo que controle um semáforo, que serão representados por 3 LEDs no microcontrolador Arduino;</li><li>✓ O código deverá ser construído e testado no UpRobotics;</li></ul> <p>Desta forma, o objetivo do exercício será simular o funcionamento de um semáforo com Arduino A lógica do funcionamento de um sinal, de modo geral, segue sempre o mesmo padrão:</p> <p>Verde &gt; Amarelo &gt; Vermelho &gt; Verde</p> <p>O aluno deverá definir pausas em determinados segundos para as mudanças de um sinal para o outro, de modo que apenas um sinal fique ligado.</p> <p>Abaixo, um exemplo de como poderá ficar a solução:</p>	
----------	--	--

The screenshot displays the Blockly IDE interface for a traffic light simulation. The workspace contains the following blocks:

- Initial state: Vermelho 10, Amarelo 5, Verde 10.
- Loop (Repetir):
  - Acender o LED (Red)
  - Apagar o LED (Red)
  - Aguardar 2 segundos (Red)
  - Acender o LED (Yellow)
  - Apagar o LED (Yellow)
  - Aguardar 2 segundos (Yellow)
  - Acender o LED (Green)
  - Apagar o LED (Green)
  - Aguardar 2 segundos (Green)

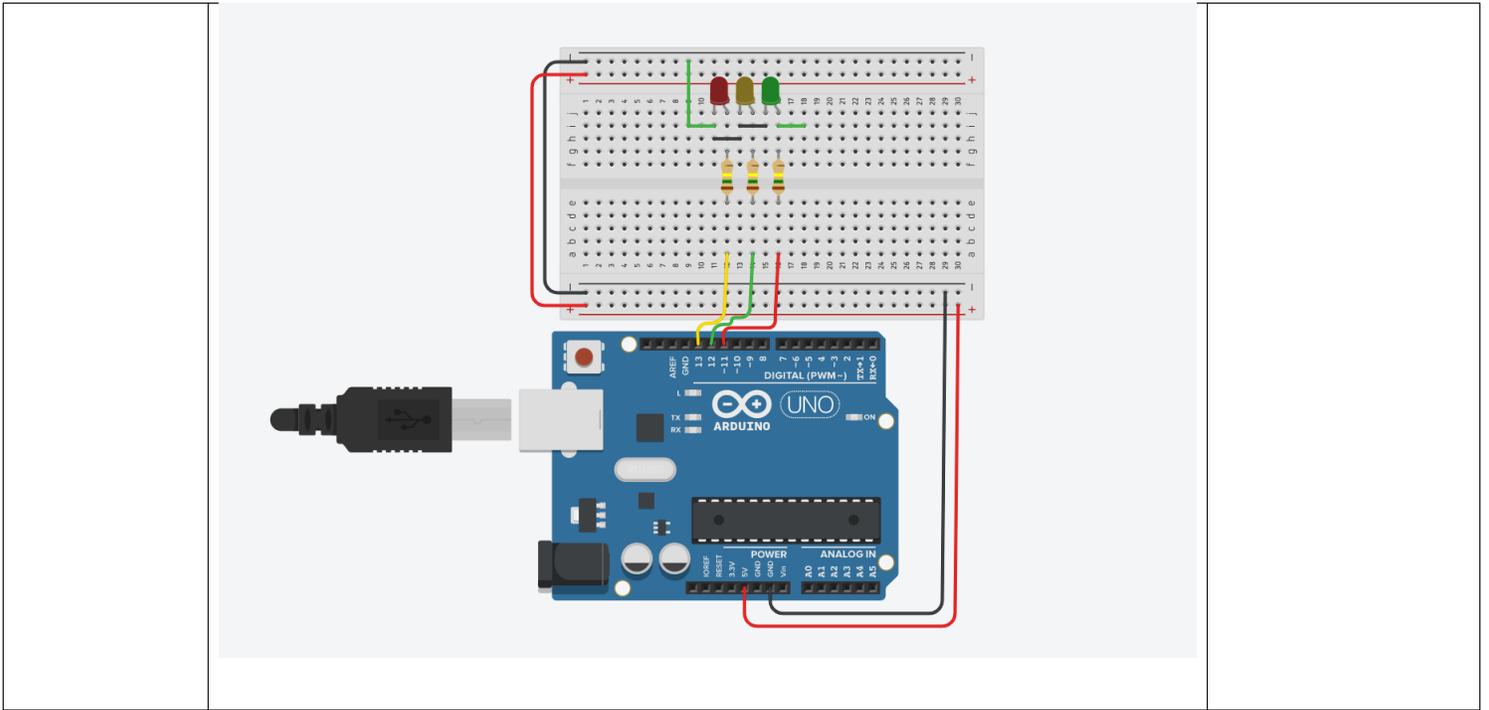
The 'Código-fonte' window on the right shows the following C++ code:

```
int red = 13;
int green = 11;
int yellow = 12;

void setup()
{
  pinMode(red, OUTPUT);
  pinMode(yellow, OUTPUT);
  pinMode(green, OUTPUT);
}

void loop()
{
  digitalWrite(red, HIGH);
  delay(2000);
  digitalWrite(red, LOW);
  digitalWrite(green, HIGH);
  delay(2000);
  digitalWrite(green, LOW);
  digitalWrite(yellow, HIGH);
  delay(2000);
  digitalWrite(yellow, LOW);
}
```

Abaixo está o circuito do Arduino utilizado na aula:



#### 4. AVALIAÇÃO

1	Percebe que alguma coisa não está funcionando como o esperado	5	4	3	2	1	0	NA
2	Conserva a meta original	5	4	3	2	1	0	NA
3	Tem uma hipótese por causa do problema	5	4	3	2	1	0	NA
4	Tenta solucionar o problema	5	4	3	2	1	0	NA
5	Consegue usar o UpRobotics para acender os LEDs	5	4	3	2	1	0	NA
6	Desenvolve o algoritmo corretamente pra acender os LEDs	5	4	3	2	1	0	NA

## 5. INDICAÇÕES BIBLIOGRÁFICAS

BRASIL. **Parâmetros Curriculares Nacionais: Matemática.2.** ed Brasília: MEC/SEF, 1997.

GEOGEBRA. **Noções de grandezas físicas e unidades de medidas.** Disponível em: <https://www.geogebra.org/m/kvggmgru>



UFAM

UNIVERSIDADE FEDERAL DO AMAZONAS  
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA  
INSTITUTO DE COMPUTAÇÃO

PLANO DE AULA

1. IDENTIFICAÇÃO

<b>Professor:</b> Francisco Gabriel Teixeira Marinho	<b>Disciplina:</b> Matemática Básica	<b>Tema:</b> Ângulos	<b>Data/Hora:</b> 19 de Fevereiro de 2022, 1 hora / aula.
---	---	-------------------------	--

2. PLANO

	OBJETIVOS	CONTEÚDOS	RECURSOS
<b>GERAL</b>	Aplicar os conceitos de ângulos em contextos reais com o auxílio de robótica educacional	<ul style="list-style-type: none"><li>✓ Ângulos e tipos</li><li>✓ Exemplos de Ângulos no dia a dia</li><li>✓ Robótica Educacional</li></ul>	<ul style="list-style-type: none"><li>✓ Projetor multimídia;</li><li>✓ Computador com <i>software</i> de apresentação e de exibição de vídeo;</li><li>✓ Ferramenta UpRobotics</li><li>✓ Arduino UNO virtual e físico</li><li>✓ Circuito virtual do Braço Robótico</li><li>✓ Braço Robótico Físico</li><li>✓ Thymio II no simulador presente no UpRobotics</li></ul>
<b>ESPECÍFICOS</b>	<ul style="list-style-type: none"><li>✓ Aprender os tipos de ângulos</li><li>✓ Entender como ângulos estão ao nosso redor e como são aplicadas na tecnologia</li><li>✓ Saber aplicar esses conhecimentos para resolver problemas reais</li><li>✓ Controlar braço robótico utilizando conceitos de unidades de medidas de tempo</li><li>✓ Controlar um robô carrinho Thymio II utilizando conceitos de unidades de ângulos</li></ul>		

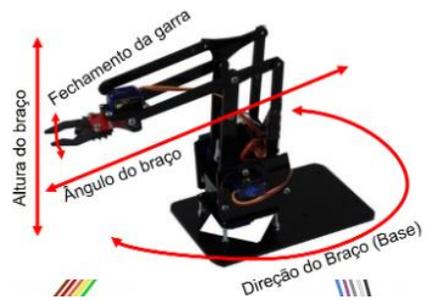
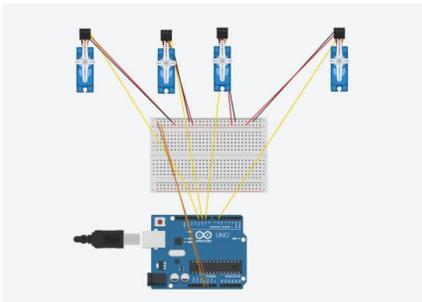
3. PROCEDIMENTOS

INTRODUÇÃO	DESENVOLVIMENTO	CONCLUSÃO
<ul style="list-style-type: none"><li>✓ Inicialmente, haverá uma aula dos conceitos sobre ângulos</li></ul>	<ul style="list-style-type: none"><li>✓ Após a aula, será apresentada uma situação-problema cuja solução irá construir o conceito de ângulo</li><li>✓ Na sequência, será apresentada um pequeno vídeo sobre ângulos da plataforma Geogebra;</li></ul>	<ul style="list-style-type: none"><li>✓ A aula terminará com os alunos respondendo um formulário de</li></ul>

como pré-requisito para o desenvolvimento da aula.

- ✓ Será enfatizado ângulos na robótica educacional;
- ✓ Será exibido um vídeo com uma reportagem sobre como funcionam os ângulos nos robôs;
- ✓ Os alunos serão convidados a criar grupos para resolverem em equipe um problema;
- ✓ O problema dado a eles serão dois:
  - 1) Movimentar a **altura, base, garra e direção/ângulo** de um robô usando o conhecimento de ângulos;
  - 2) Movimentar um carrinho para que ele ande até um determinado ponto (alvo)Os códigos deverão ser construídos e testados no UpRobotics;

Abaixo está o circuito do Arduino utilizado na aula para o braço robótico, braço utilizado e um exemplo de algoritmo usando blocos do UpRobotics:

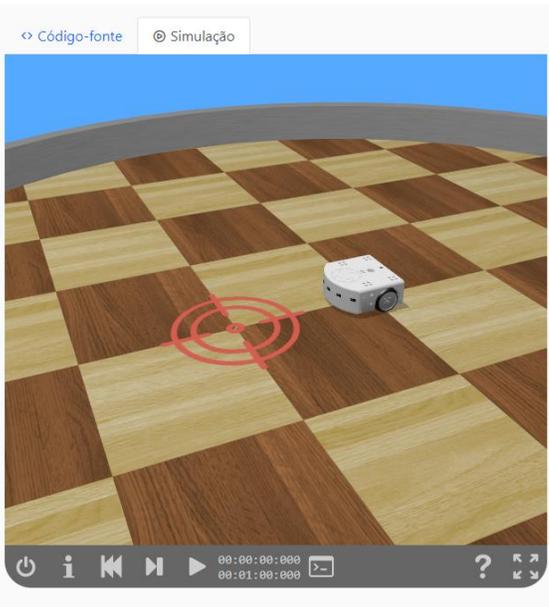


avaliação de usabilidade e funcionalidade;

The image shows a Scratch script for controlling a Thymio II robot arm. The script is composed of several blocks:

- Importar todos os motores de controle** (Import all control motors)
- Ativar motores de controle** (Activate control motors)
- Definir portas no Arduino** (Define ports on Arduino) with parameters: base: 11, ângulo: 9, altura: 10 e garra: 6.
- Movimentos do braço robótico:** (Robot arm movements) - a large orange block containing:
  - Inicializar braço robótico nas posições** (Initialize robot arm in positions) with parameters: Base: 90 graus, Ângulo: 90 graus, Altura: 90 graus e Garra: 35 graus.
  - fechar a garra** (close the gripper) with parameter: 60 graus.
  - gire o braço** (rotate the arm) with parameter: 90 graus para a esquerda.
  - mover altura do braço para a posição** (move arm height to position) with parameter: 20 graus.
  - abrir a garra** (open the gripper) with parameter: 35 graus.
  - mover ângulo do braço para a posição** (move arm angle to position) with parameter: 20 graus.
  - gire o braço** (rotate the arm) with parameter: 91 graus para a direita.

A abaixo está o cenário do carrinho Thymio II para ir em direção ao alvo e um exemplo de solução no bloco:



- ⚙ Defina a posição alvo do motor em  radianos
- 💬 Definir mensagem de saída no console
- ⚠ Definir mensagem de erro no console

#### 4. AVALIAÇÃO

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

4	Tenta solucionar o problema	5	4	3	2	1	0	NA
5	Consegue usar o UpRobotics para programar os movimentos do braço robótico	5	4	3	2	1	0	NA
6	Desenvolve o algoritmo corretamente para permitir que o Thymio II atinja o seu alvo	5	4	3	2	1	0	NA

#### 5. INDICAÇÕES BIBLIOGRÁFICAS

BRASIL. **Parâmetros Curriculares Nacionais: Matemática.2**. ed Brasília: MEC/SEF, 1997.

GEOGEBRA. **Ângulos**. Disponível em: <https://www.geogebra.org/m/fztrdsfv>

**UpRobotics: Robótica Educacional Utilizando Linguagem Visual Baseada em Blocos**. Disponível em: [https://sol.sbc.org.br/index.php/sbesc\\_estendido/article/view/13101/12954](https://sol.sbc.org.br/index.php/sbesc_estendido/article/view/13101/12954)

# Apêndice C

## Questionários de Avaliação

O questionário de avaliação foi subdividido em 4 questionários:

QUESTIONÁRIO 1 (Mostrado na Figura C.1), é sobre a aula onde o aluno programou um Semáforo usando conhecimentos de grandezas e medidas.

QUESTIONÁRIO 2 (Mostrado na Figura C.2), é sobre a aula onde o aluno programou um Braço Robótico usando conhecimentos de Ângulos.

QUESTIONÁRIO 3 (Mostrado na Figura C.3), é sobre a aula onde o aluno programou um robô carrinho Thymio II usando conhecimento de Ângulos.

QUESTIONÁRIO 4 (Mostrado nas Figuras C.4 e C.5), é sobre a ferramenta UpRobotics, num contexto geral.

1	O que você achou da aula sobre Grandezas e Medidas?	 Muito Legal  Legal  Bom  Mais ou Menos  Ruim
2	O que você achou de programar um semáforo com o UpRobotics?	 Muito Legal  Legal  Bom  Mais ou Menos  Ruim
3	O que você achou de ter aplicado seu conhecimento sobre grandezas e medidas para programar um semáforo?	 Muito Legal  Legal  Bom  Mais ou Menos  Ruim
4	Você achou fácil ou difícil encontrar os blocos que precisava para fazer seu programa?	 Muito Fácil  Fácil  Mais ou menos  Difícil  Muito Difícil
5	Os ícones dos blocos ajudaram você a encontrar o bloco que você precisava?	 Sim  Mais ou Menos  Não
6	Você achou fácil ou difícil encaixar e desencaixar os blocos?	 Muito Fácil  Fácil  Mais ou menos  Difícil  Muito Difícil
7	Você achou fácil ou difícil testar o seu algoritmo no simulador?	 Muito Fácil  Fácil  Mais ou menos  Difícil  Muito Difícil
8	Quando você programou o semáforo, isso ajudou você a entender um pouco mais sobre medida de tempo?	 Sim  Mais ou Menos  Não

Figura C.1: QUESTIONÁRIO 1 – Sobre a aula onde o aluno programou um Semáforo usando conhecimentos de grandezas e medidas

1	O que você achou da aula sobre Ângulos?	 Muito Legal	 Legal	 Bom	 Mais ou Menos	 Ruim
2	O que você achou de programar um braço robótico com o UpRobotics?	 Muito Legal	 Legal	 Bom	 Mais ou Menos	 Ruim
3	O que você achou de ter aplicado seu conhecimento sobre ângulos para os movimentos do braço robótico?	 Muito Legal	 Legal	 Bom	 Mais ou Menos	 Ruim
4	Você achou fácil ou difícil encontrar os blocos que precisava para fazer seu programa?	 Muito Fácil	 Fácil	 Mais ou menos	 Difícil	 Muito Difícil
5	Os ícones dos blocos ajudaram você a encontrar o bloco que você precisava?	 Sim	 Mais ou Menos	 Não		
6	Você achou fácil ou difícil encaixar e desencaixar os blocos?	 Muito Fácil	 Fácil	 Mais ou menos	 Difícil	 Muito Difícil
7	Você achou fácil ou difícil testar o seu algoritmo no simulador?	 Muito Fácil	 Fácil	 Mais ou menos	 Difícil	 Muito Difícil
8	Quando você programou o braço robótico isso ajudou você a entender um pouco mais sobre ângulos?	 Sim	 Mais ou Menos	 Não		

Figura C.2: QUESTIONÁRIO 2 – Sobre a aula onde o aluno programou um Braço Robótico usando conhecimentos de Ângulos.

1	O que você achou da aula sobre Ângulos?	 Muito Legal	 Legal	 Bom	 Mais ou Menos	 Ruim
2	O que você achou de programar o carrinho Thymio II com o UpRobotics?	 Muito Legal	 Legal	 Bom	 Mais ou Menos	 Ruim
3	O que você achou de ter aplicado seu conhecimento sobre ângulos o movimentar o Thymio II?	 Muito Legal	 Legal	 Bom	 Mais ou Menos	 Ruim
4	Você achou fácil ou difícil encontrar o bloco que precisava para fazer seu programa?	 Muito Fácil	 Fácil	 Mais ou menos	 Difícil	 Muito Difícil
5	Os ícones do bloco ajudaram você a saber onde inserir o valor correto para colocar o carrinho no alvo?	 Sim	 Mais ou Menos	 Não		
6	Você achou fácil ou difícil programar esse robô usando apenas um bloco?	 Muito Fácil	 Fácil	 Mais ou menos	 Difícil	 Muito Difícil
7	Você achou fácil ou difícil testar o seu algoritmo no simulador?	 Muito Fácil	 Fácil	 Mais ou menos	 Difícil	 Muito Difícil
8	Quando você programou o Thymio II isso ajudou você a entender um pouco mais sobre ângulos?	 Sim	 Mais ou Menos	 Não		

Figura C.3: QUESTIONÁRIO 3 – Sobre a aula onde o aluno programou um robô carrinho Thymio II usando conhecimento de Ângulos.

1	O que você achou da ferramenta UpRobotics?	 Muito Legal	 Legal	 Bom	 Mais ou Menos	 Ruim
2	O UpRobotics é fácil ou difícil de usar?	 Muito Fácil	 Fácil	 Mais ou menos	 Difícil	 Muito Difícil
3	Você acha fácil ou difícil encontrar os blocos que precisa para fazer seu programa?	 Muito Fácil	 Fácil	 Mais ou menos	 Difícil	 Muito Difícil
4	Os desenhos que têm nos blocos ajudam você a encontrar o bloco que você está procurando?	 Sim	 Mais ou Menos	 Não		
5	Os textos dos blocos ajudam você a encontrar o bloco que está procurando?	 Sim	 Mais ou Menos	 Não		
6	Você achou fácil ou difícil selecionar o robô desejado?	 Muito Fácil	 Fácil	 Mais ou menos	 Difícil	 Muito Difícil
7	Você achou fácil ou difícil copiar o seu código?	 Muito Fácil	 Fácil	 Mais ou menos	 Difícil	 Muito Difícil
8	Você achou fácil ou difícil testar o seu código no simulador?	 Muito Fácil	 Fácil	 Mais ou menos	 Difícil	 Muito Difícil

Figura C.4: QUESTIONÁRIO 4 - Sobre a ferramenta UpRobotics, num contexto geral (parte 1)

9	Você achou fácil ou difícil acessar os tutoriais no UpRobotics?	 <p>Muito Fácil      Fácil      Mais ou menos      Difícil      Muito Difícil</p>
10	Você achou fácil ou difícil encontrar as instruções para programar os robôs no UpRobotics?	 <p>Muito Fácil      Fácil      Mais ou menos      Difícil      Muito Difícil</p>
11	Você gostaria de usar o UpRobotics em outras aulas?	 <p>Sim      Mais ou Menos      Não</p>
12	Você gostaria de programar outros robôs no UpRobotics?	 <p>Sim      Mais ou Menos      Não</p>

Figura C.5: QUESTIONÁRIO 4 - Sobre a ferramenta UpRobotics, num contexto geral (parte 2)

## Apêndice D

### Avaliação de Usabilidade e Funcionalidade - Resultados mostrados em gráficos

# UpRobotics - Avaliação de Usabilidade e Funcionalidade

5 respostas

[Publicar análise](#)

## Caracterização do participante

Qual é o seu nome?

5 respostas

B \*\*\*\*\*

F \*\*\*\*\* e

J \*\*\*\*\* i

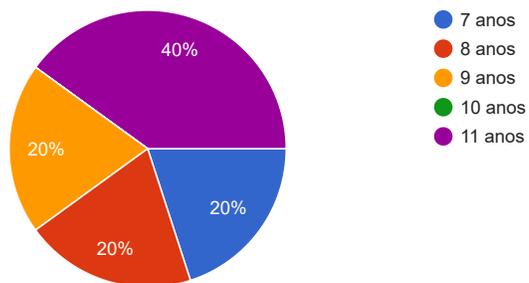
E \*\*\*\*\*

N \*\*\*\*\*



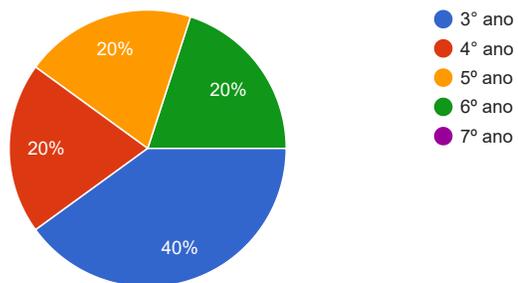
### Quantos anos você tem?

5 respostas



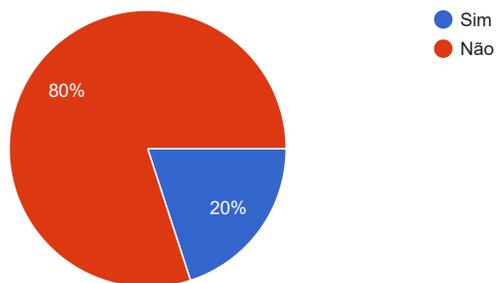
### Em que série você está?

5 respostas



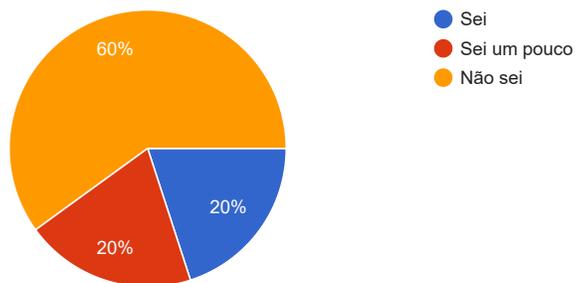
Você já ouviu falar sobre Grandezas e Medidas?

5 respostas



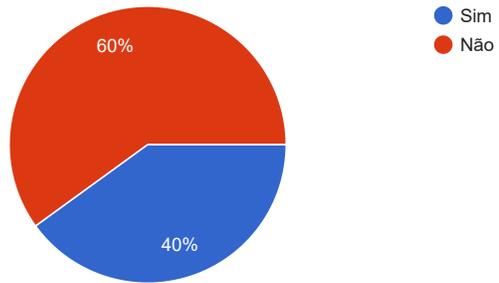
Seu nível de conhecimento sobre Grandezas e Medidas?

5 respostas



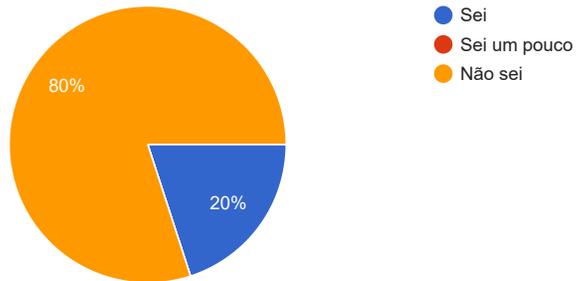
Você já ouviu falar sobre Ângulos?

5 respostas



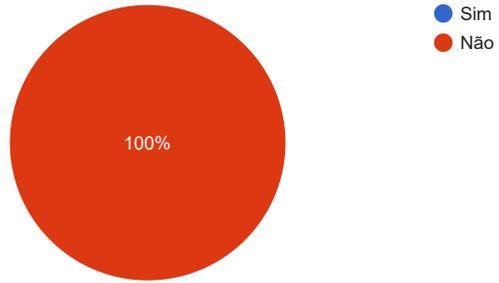
Seu nível de conhecimento sobre Ângulos?

5 respostas



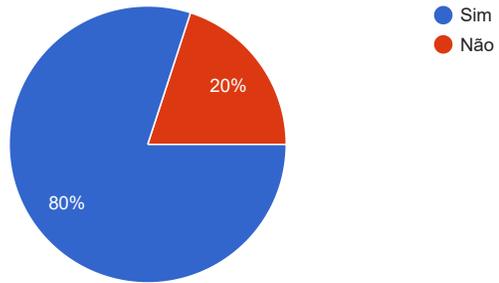
Você já ouviu falar sobre Pensamento Computacional?

5 respostas



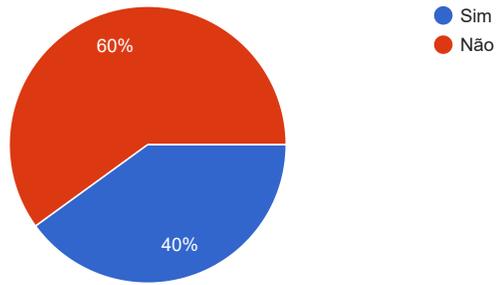
Você já ouviu falar sobre Robótica?

5 respostas



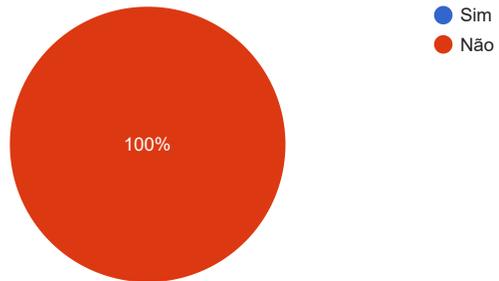
Você já ouviu falar sobre Robótica Educacional?

5 respostas



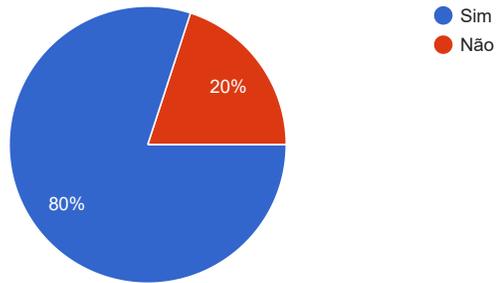
Você já ouviu falar sobre Robótica Desplugada?

5 respostas



Você já ouviu falar sobre Internet das Coisas?

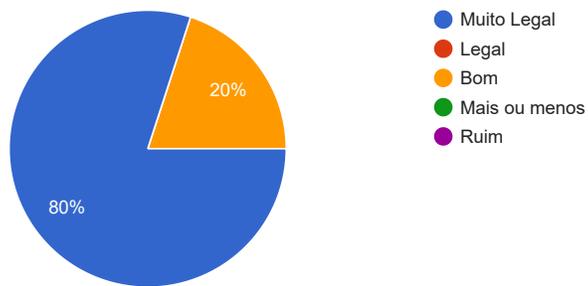
5 respostas



### Questionário 1 - Semáforo

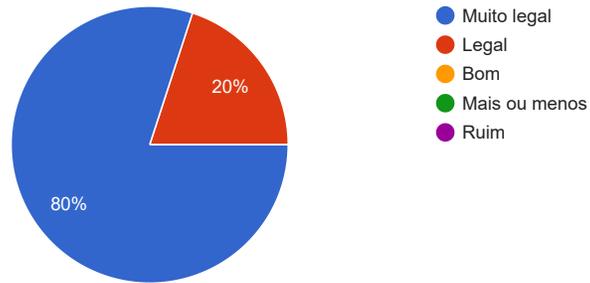
1. O que você achou da aula sobre Grandezas e Medidas?

5 respostas



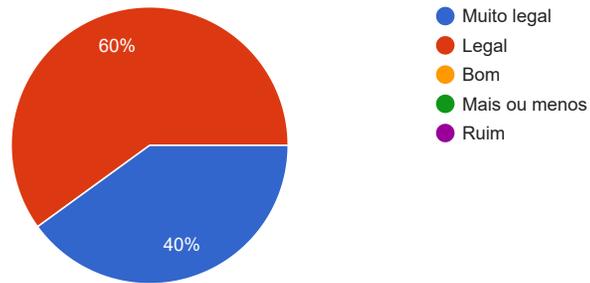
2. O que você achou de programar um semáforo com o UpRobotics?

5 respostas



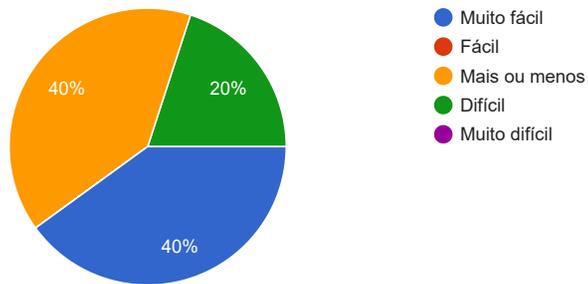
3. O que você achou de ter aplicado seu conhecimento sobre Grandezas e Medidas para programar um semáforo?

5 respostas



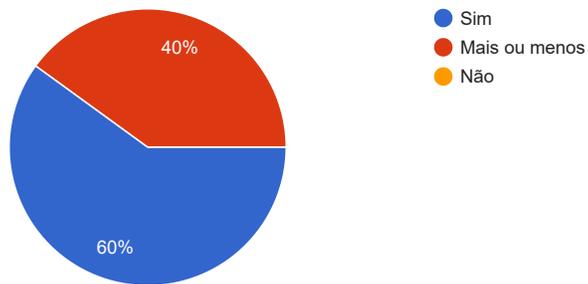
4. Você achou fácil ou difícil encontrar os blocos que precisava para fazer seu programa?

5 respostas



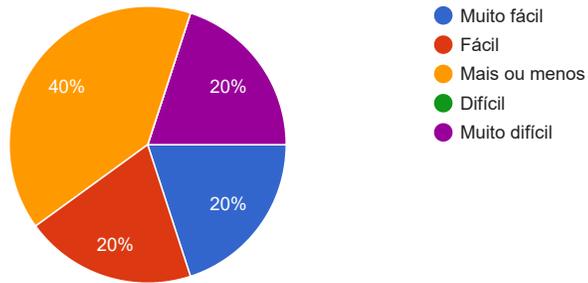
5. Os ícones dos blocos ajudaram você a encontrar o bloco que você precisava?

5 respostas



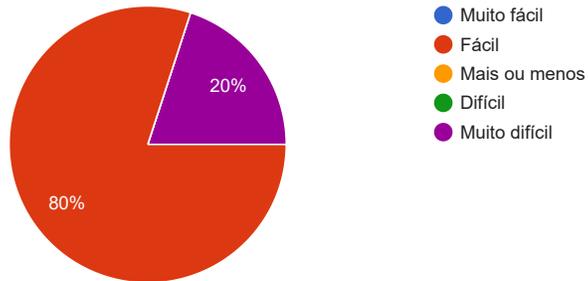
6. Você achou fácil ou difícil encaixar e desencaixar os blocos?

5 respostas



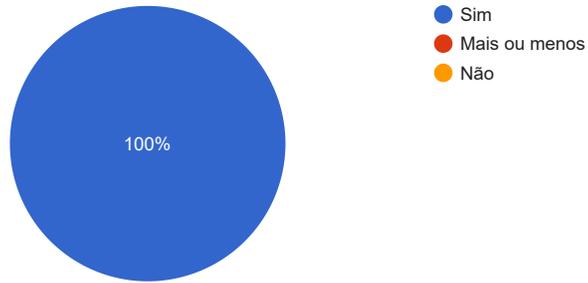
7. Você achou fácil ou difícil testar o seu algoritmo no simulador?

5 respostas



8. Quando você programou o semáforo, isso ajudou você a entender um pouco mais sobre medida de tempo?

5 respostas

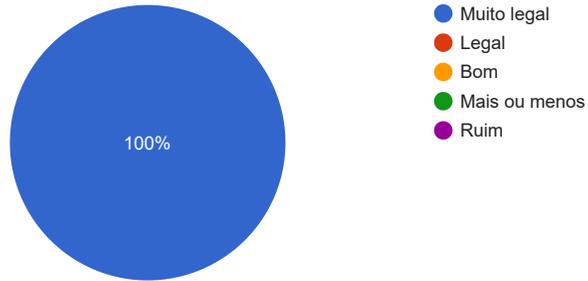


Questionário 2 - Sobre o Braço Robótico



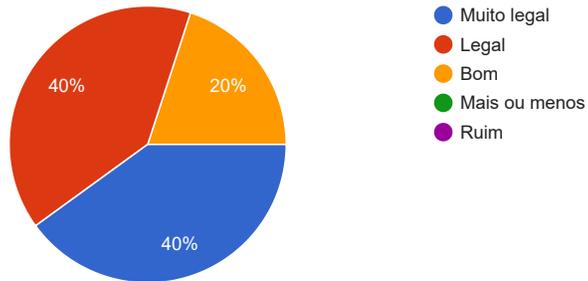
1. O que você achou da aula sobre Ângulos?

5 respostas



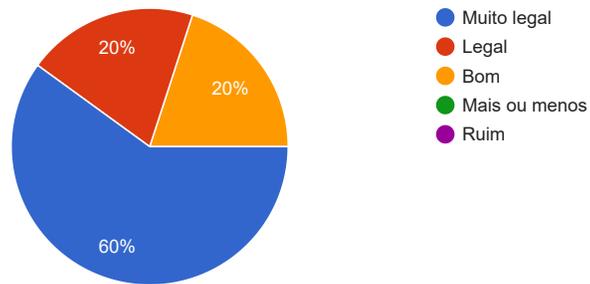
2. O que você achou de programar um braço robótico com o UpRobotics?

5 respostas



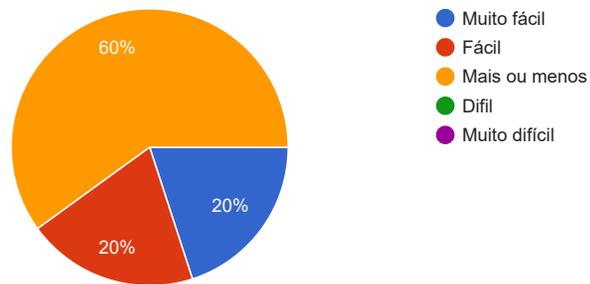
3. O que você achou de ter aplicado seu conhecimento sobre ângulo para os movimentos do braço robótico?

5 respostas



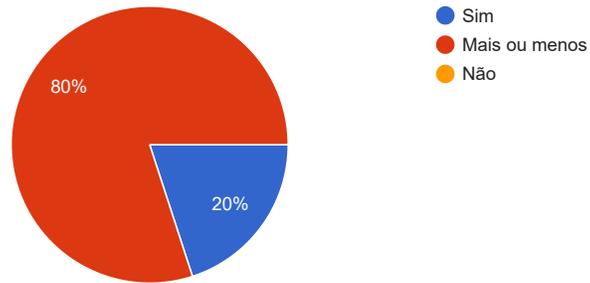
4. Você fácil ou difícil encontrar os blocos que precisava para fazer seu programa?

5 respostas



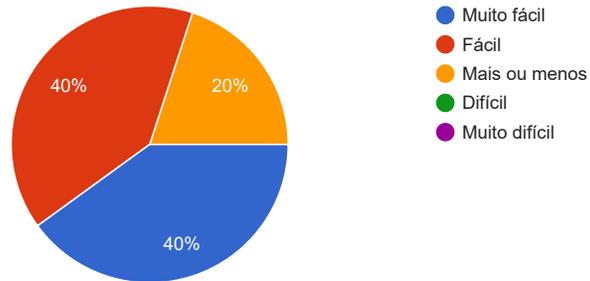
5. Os ícones dos blocos ajudaram você a encontrar o bloco que precisava?

5 respostas



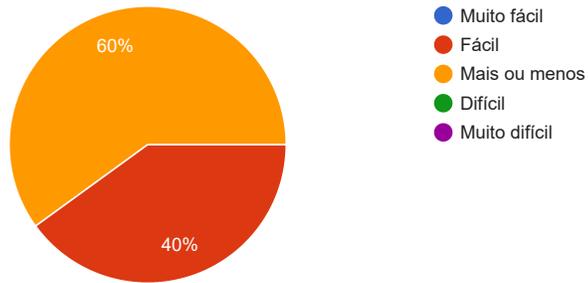
6. Você achou fácil ou difícil encaixar e desencaixar os blocos?

5 respostas



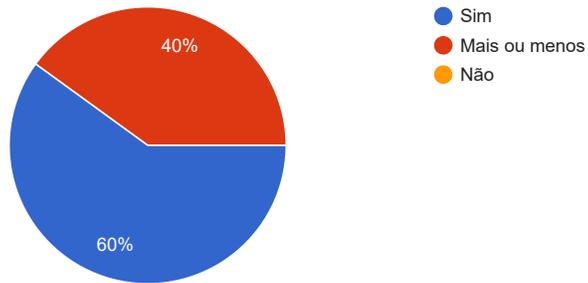
7. Você achou fácil ou difícil testar o seu algoritmo no simulador?

5 respostas



8. Quando você programou o braço robótico, isso ajudou você a entender um pouco mais sobre Ângulos?

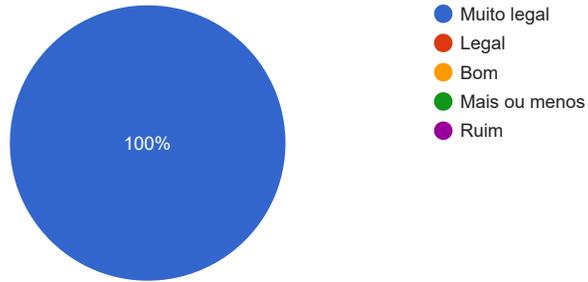
5 respostas



### Questionário 3 - Sobre o Thymio II

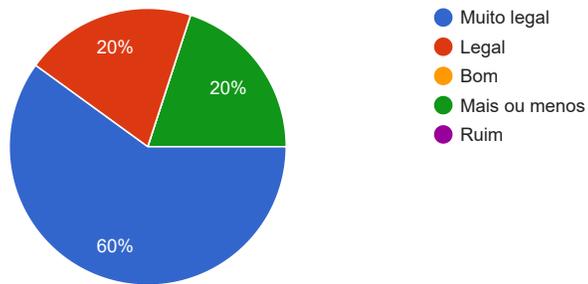
1. O que você achou da aula sobre Ângulos?

5 respostas



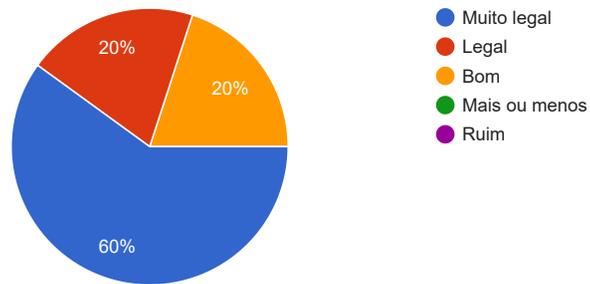
2. O que você achou de programar Thymio II com o UpRobotics?

5 respostas



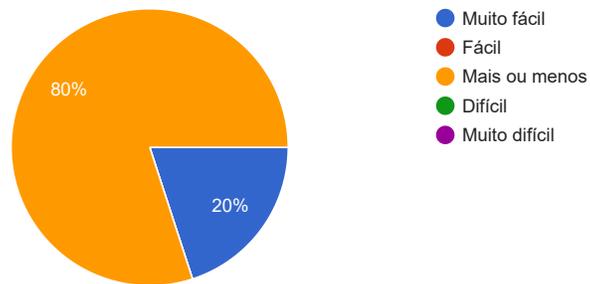
3. O que você achou de ter aplicado seu conhecimento sobre ângulos movimentar o Thymio II?

5 respostas



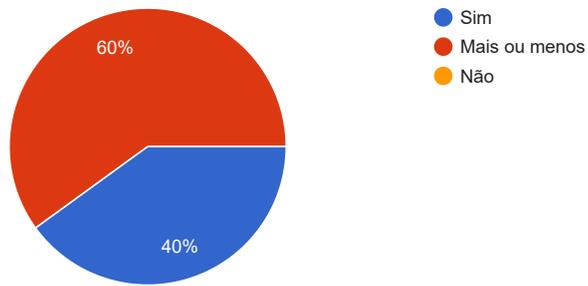
4. Você achou fácil ou difícil encontrar o bloco que precisava para fazer seu programa?

5 respostas



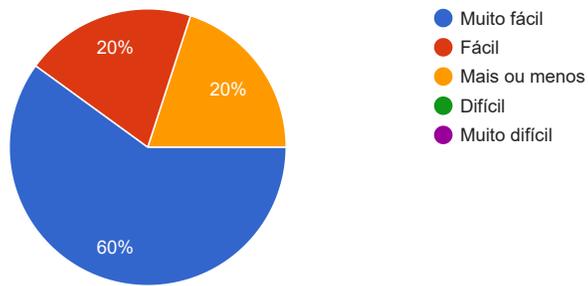
5. Os ícones do bloco ajudaram você a saber onde inserir o valor correto para colocar o carrinho no alvo?

5 respostas



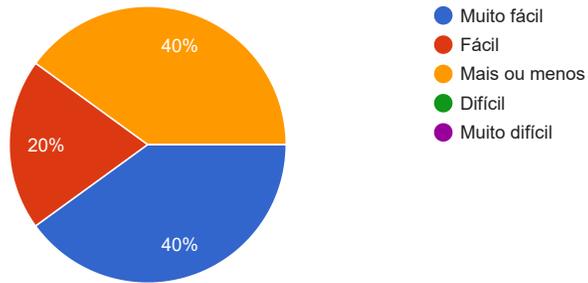
6. Você achou fácil ou difícil programar esse robô usando apenas um bloco?

5 respostas



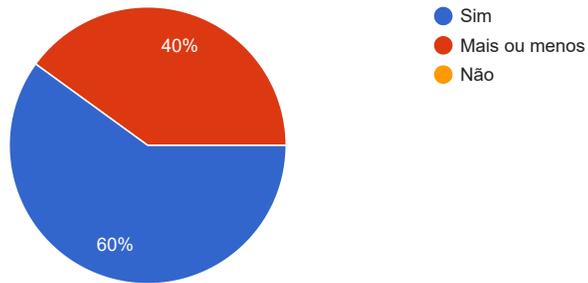
7. Você achou fácil ou difícil testar o seu algoritmo no simulador?

5 respostas



8. Quando você programou o Thymio II isso ajudou você a entender um pouco mais sobre ângulos?

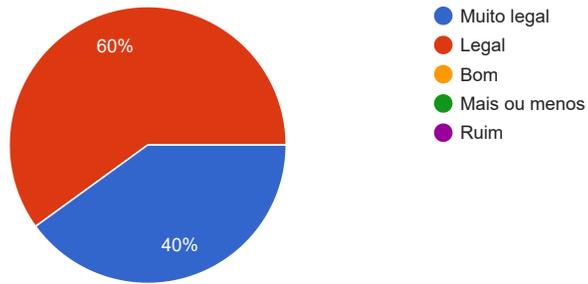
5 respostas



### QUESTIONÁRIO 4 – Sobre o UpRobotics.

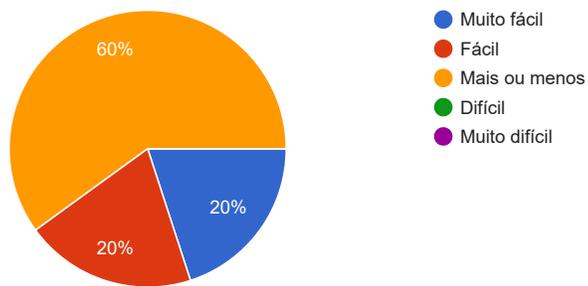
1. O que você achou da ferramenta UpRobotics?

5 respostas



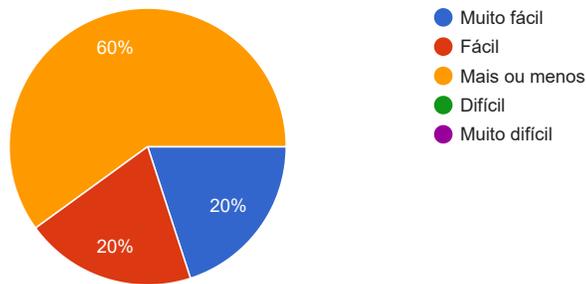
2. O UpRobotics é fácil ou difícil de usar?

5 respostas



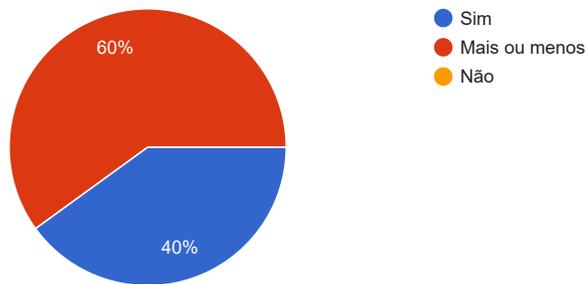
3. Você acha fácil ou difícil encontrar os blocos que precisa para fazer seu programa?

5 respostas



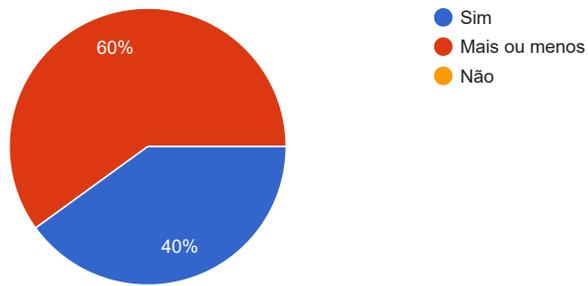
4. Os desenhos que têm nos blocos ajudam você a encontrar o bloco que você está procurando?

5 respostas



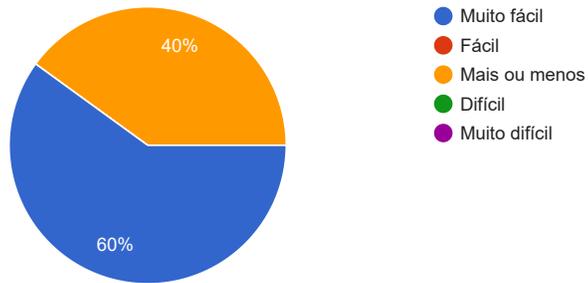
5. Os desenhos que têm nos blocos ajudam você a encontrar o bloco que você está procurando?

5 respostas



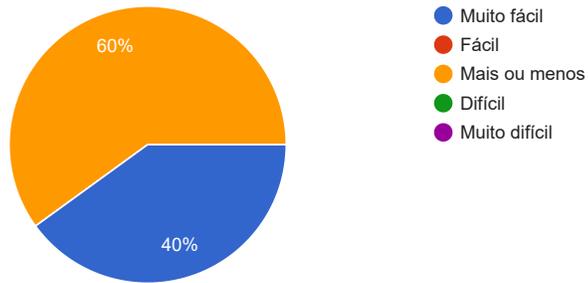
6. Você achou fácil ou difícil selecionar o robô desejado?

5 respostas



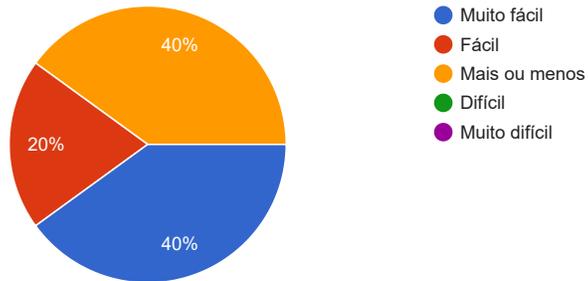
### 7. Você achou fácil ou difícil copiar o seu código?

5 respostas



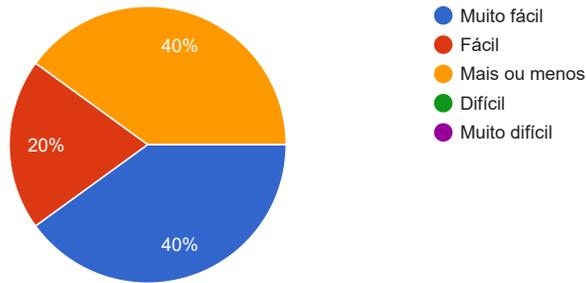
### 8. Você achou fácil ou difícil testar o seu código no simulador?

5 respostas



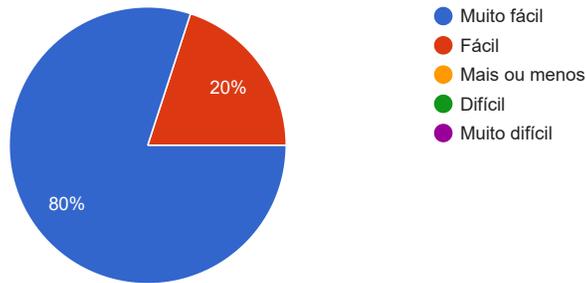
9. Você achou fácil ou difícil acessar os tutoriais no UpRobotics?

5 respostas



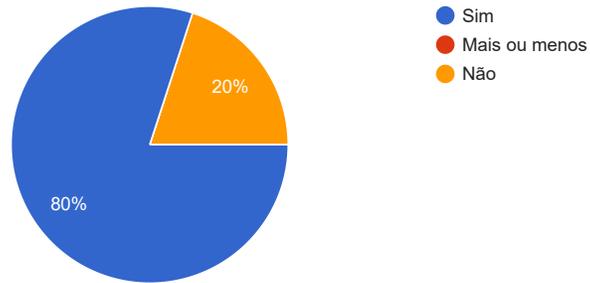
10. Você achou fácil ou difícil encontrar as instruções para programar os robôs no UpRobotics?

5 respostas



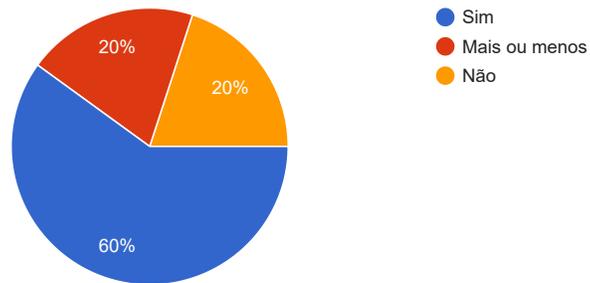
### 11. Você gostaria de usar o UpRobotics em outras aulas?

5 respostas



### 12. Você gostaria de programar outros robôs no UpRobotics?

5 respostas



Este conteúdo não foi criado nem aprovado pelo Google. [Denunciar abuso](#) - [Termos de Serviço](#) - [Política de Privacidade](#)



# Apêndice E

## UpRobotics: Blocos e seus códigos-fonte equivalentes

Para utilizar a ferramenta UpRobotics, ver os blocos e acompanhar a evolução do projeto, acesse: <https://uprobotics-ten.vercel.app/>

## E.1 Semáforo

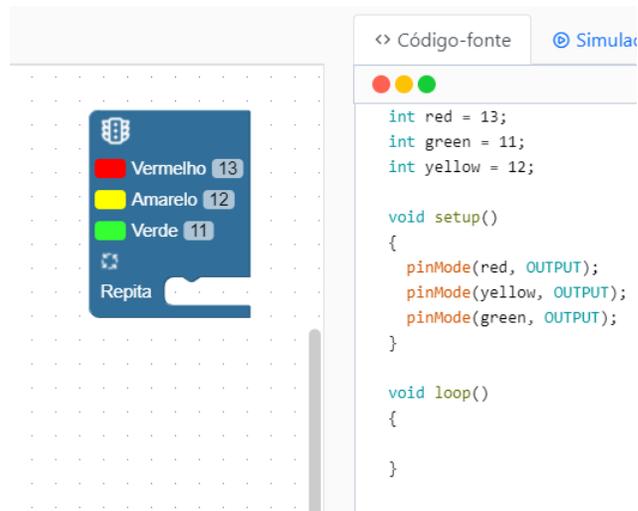


Figura E.1: Semáforo: definir portas no Arduino

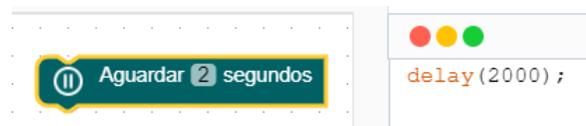


Figura E.2: Semáforo: aguardar N segundos



Figura E.3: Semáforo: acender o LED vermelho



Figura E.4: Semáforo: apagar o LED vermelho



Figura E.5: Semáforo: acender o LED verde



Figura E.6: Semáforo: apagar o LED verde



Figura E.7: Semáforo: acender o LED amarelo



Figura E.8: Semáforo: apagar o LED amarelo

## E.2 Braço Robótico - Modo básico

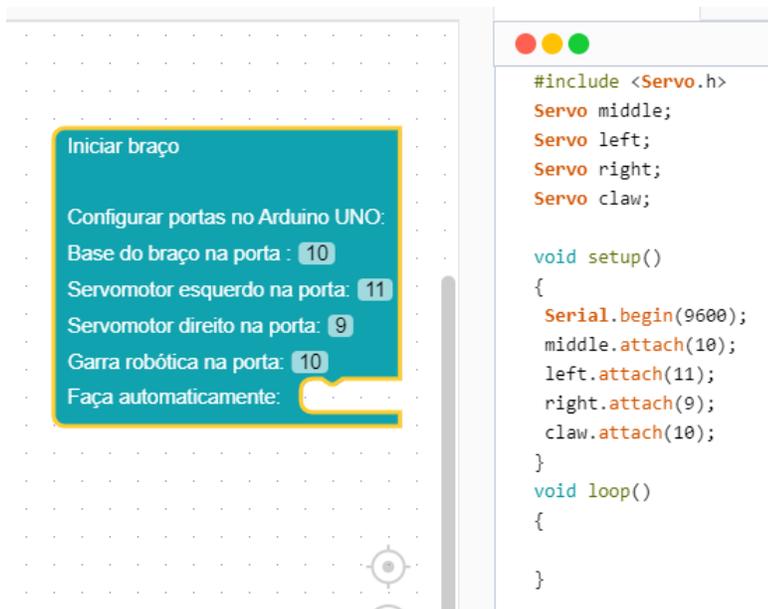


Figura E.9: Braço Robótico - Modo básico: Definir portas no arduino

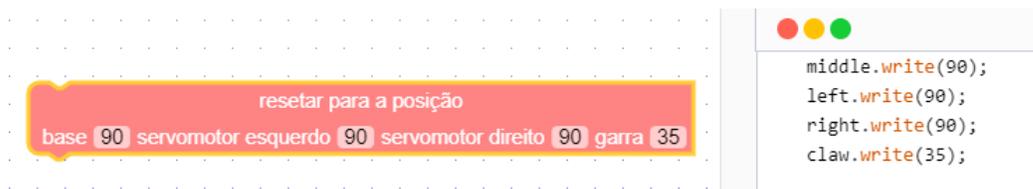


Figura E.10: Braço Robótico - Modo básico: resetar movimento

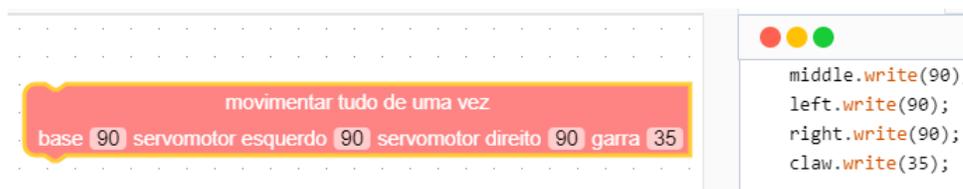


Figura E.11: Braço Robótico - Modo básico: movimentar de uma vez só



Figura E.12: Braço Robótico - Modo básico: base



Figura E.13: Braço Robótico - Modo básico: altura



Figura E.14: Braço Robótico - Modo básico: ângulo/orientação



Figura E.15: Braço Robótico - Modo básico: garra

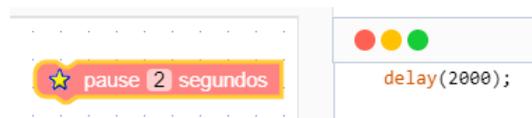


Figura E.16: Braço Robótico - Modo básico: pausa

### E.3 Braço Robótico - Modo avançado

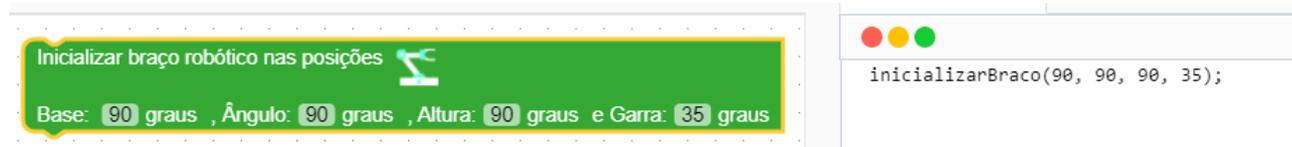


Figura E.17: Braço Robótico - Modo avançado: inicializar braço robótico a partir de uma determinada posição

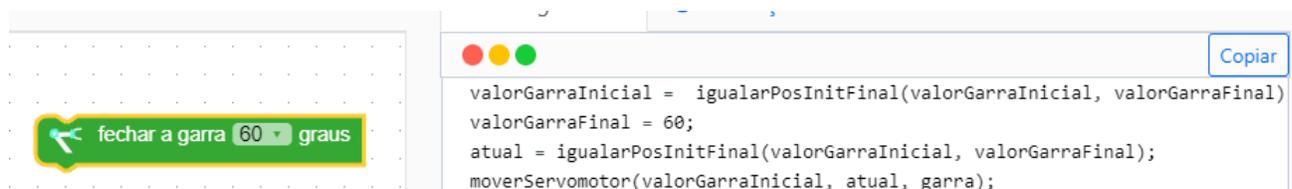


Figura E.18: Braço Robótico - Modo avançado: fechar a garra N graus

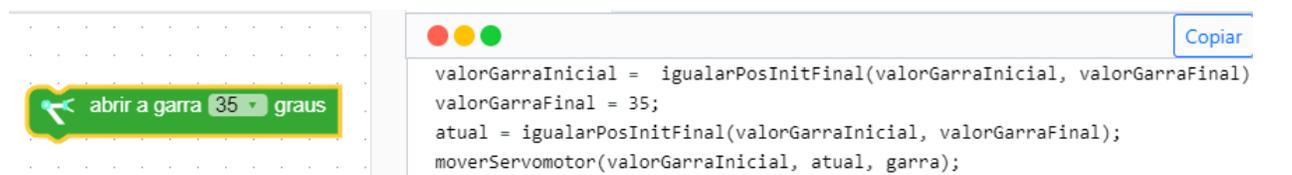


Figura E.19: Braço Robótico - Modo avançado: abrir a garra N graus

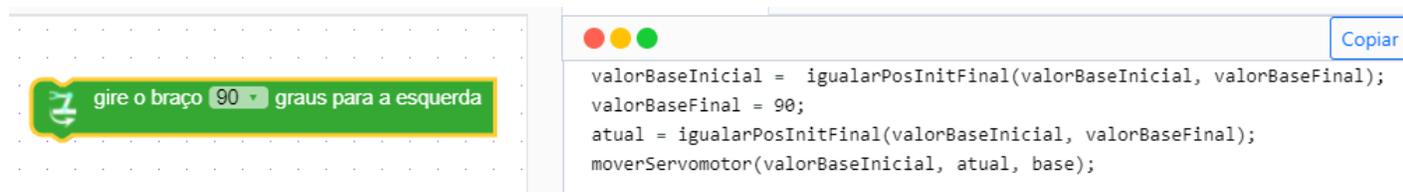


Figura E.20: Braço Robótico - Modo avançado: girar o braço para a esquerda N graus

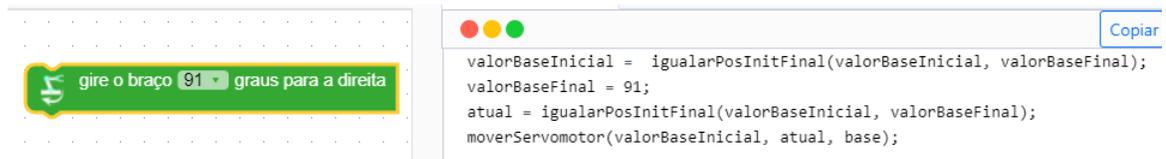


Figura E.21: Braço Robótico - Modo avançado: girar o braço para a direita

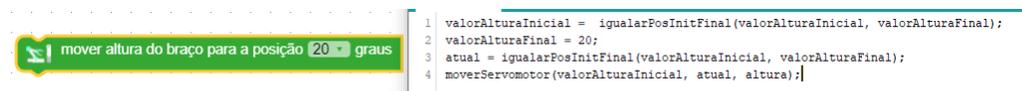


Figura E.22: Braço Robótico - Modo avançado: mover a altura do braço N graus

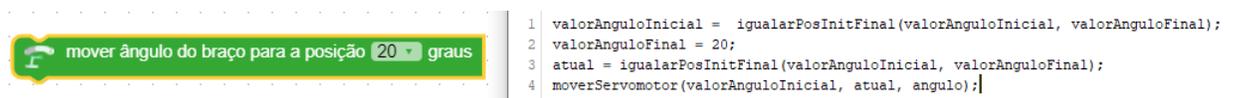


Figura E.23: Braço Robótico - Modo avançado: mover o ângulo/orientação do braço N graus

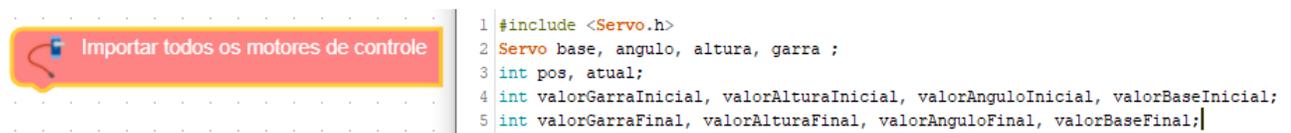
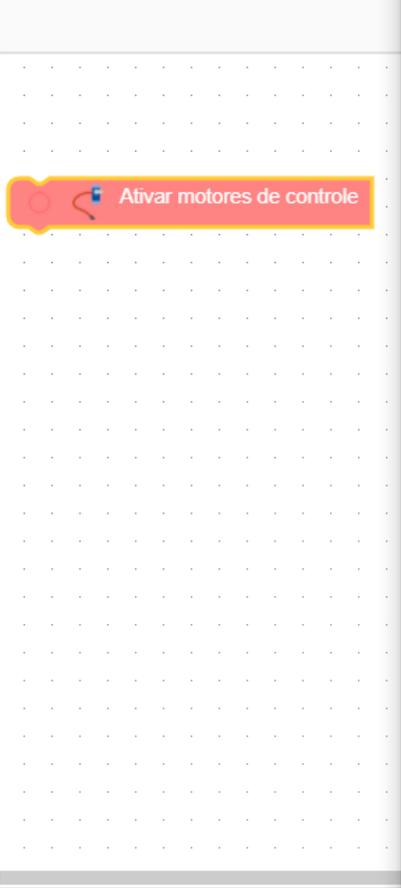


Figura E.24: Braço Robótico - Modo avançado: importar todos os motores



```

1 int igualarPosInitFinal(int PosInit, int PosFinal) {
2   PosInit = PosFinal;
3   return PosInit;
4 }
5 int moverServomotor(int a, int b, Servo motor) {
6   int pos;
7   if (a < b) {
8     for (pos = a; pos <= b; pos += 1)
9     {
10      motor.write(pos);
11      delay(15);
12    }
13  }
14  if (a >= b) {
15    for (pos = a; pos >= b; pos -= 1)
16    {
17      motor.write(pos);
18      delay(15);
19    }
20  }
21 }
22 int inicializarBraco(int bas, int ang, int alt, int gar) {
23   valorBaseInicial = bas;
24   valorAnguloInicial = ang;
25   valorAlturaInicial = alt;
26   valorGarraInicial = gar;
27   valorBaseFinal = valorBaseInicial;
28   valorAnguloFinal = valorAnguloInicial;
29   valorAlturaFinal = valorAlturaInicial;
30   valorGarraFinal = valorGarraInicial;
31   base.write(valorBaseInicial);
32   angulo.write(valorAnguloInicial);
33   altura.write(valorAlturaInicial);
34   garra.write(valorGarraInicial);
35   delay(300);
36 }
37
38

```

Figura E.25: Braço Robótico - Modo avançado: ativar todos os motores



```

void loop()
{
}

```

Figura E.26: Braço Robótico - Modo avançado: movimentos automáticos

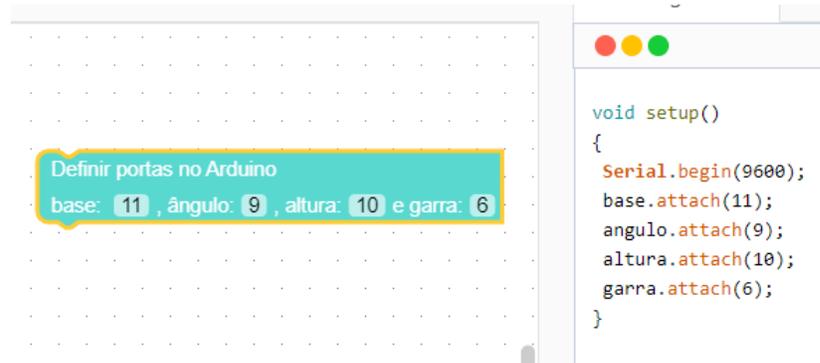


Figura E.27: Braço Robótico - Modo avançado: definir portas no arduino

## E.4 Thymio II - Alvo

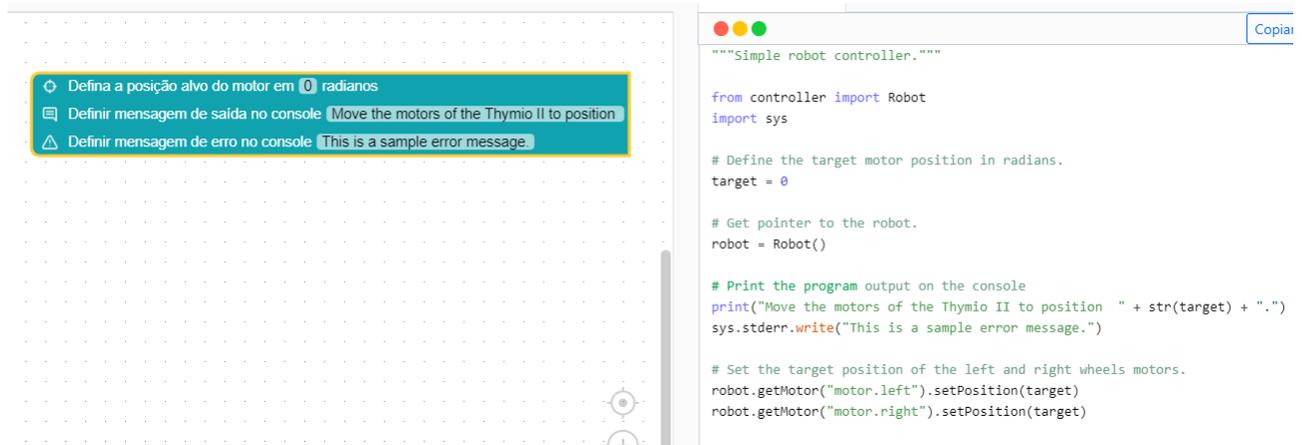


Figura E.28: Thymio II - Alvo: definir ângulos, para deslocar o carrinho para um determinado lugar

## E.5 Thymio II - Obstáculos

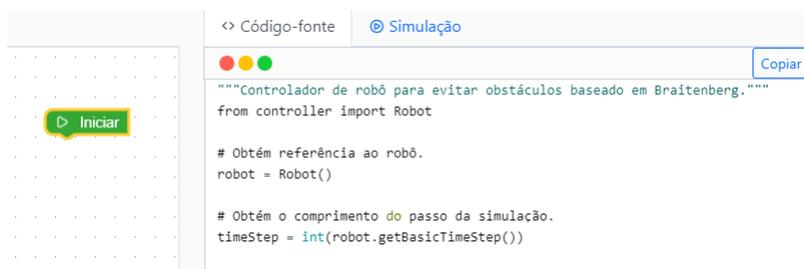


Figura E.29: Thymio II - Obstáculos: Inicializar robô



Figura E.30: Thymio II - Obstáculos: definir constantes



Figura E.31: Thymio II - Obstáculos: obter motores da roda esquerda e da roda direita

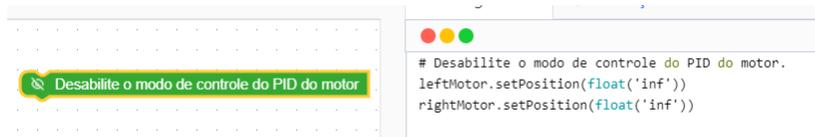


Figura E.32: Thymio II - Obstáculos: desabilitar módulo de controle

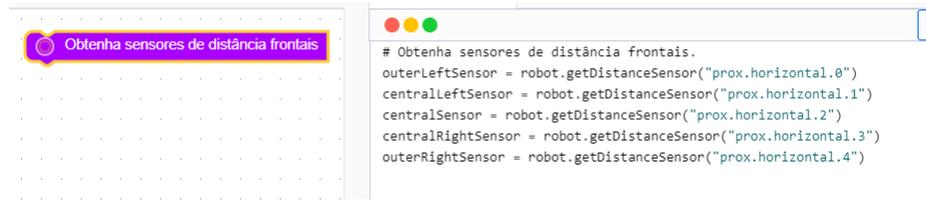


Figura E.33: Thymio II - Obstáculos: definir ângulos, para deslocar o carrinho para um determinado lugar

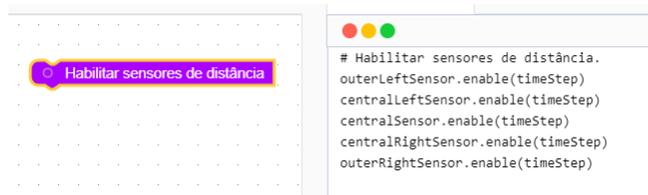


Figura E.34: Thymio II - Obstáculos: obter sensores de distância frontais

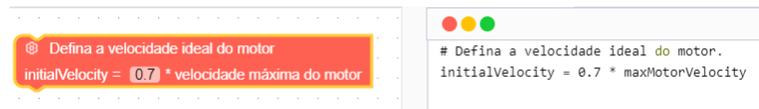


Figura E.35: Thymio II - Obstáculos: habilitar sensores de distância

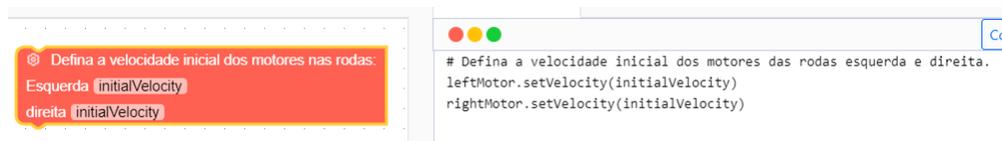


Figura E.36: Thymio II - Obstáculos: definir velocidade ideal dos motores das rodas

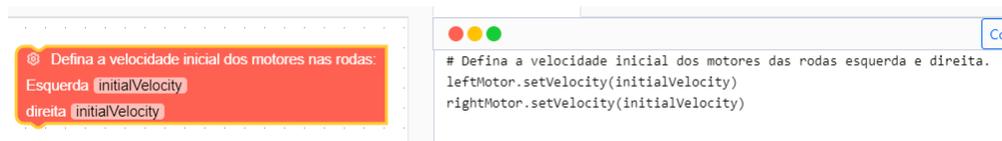


Figura E.37: Thymio II - Obstáculos: definir velocidade inicial dos motores das rodas

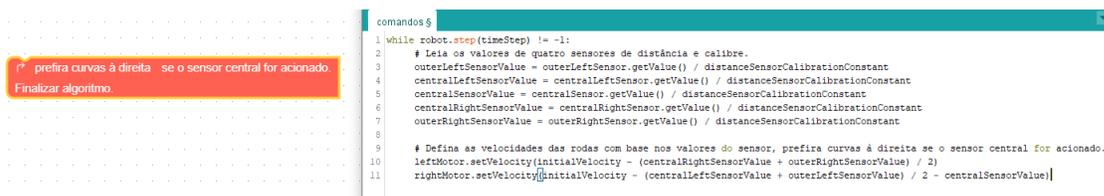


Figura E.38: Thymio II - Obstáculos: curvas à direita se o sensor frontal for acionado

## E.6 Pioneer 3D-X - Seguidor de parede

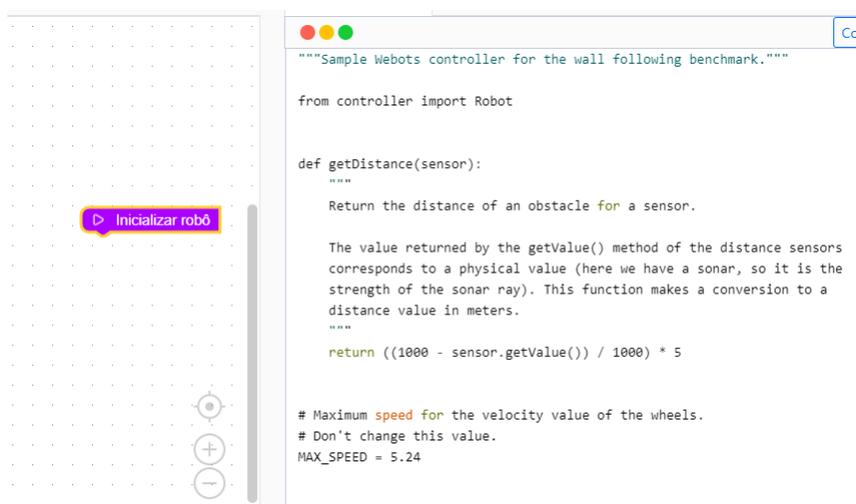


Figura E.39: Pioneer 3D-X Seguidor de parede: Iniciar o robô

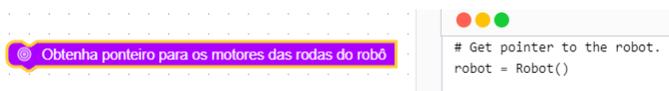


Figura E.40: Pioneer 3D-X Seguidor de parede: obter ponteiros para os motores das rodas do robô



Figura E.41: Pioneer 3D-X Seguidor de parede: Obtenha o ponteiro para o robô

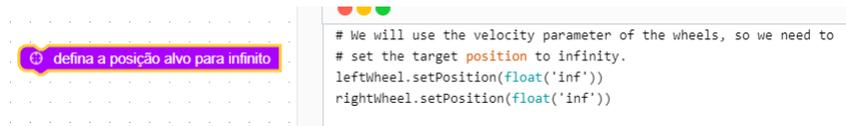


Figura E.42: Pioneer 3D-X Seguidor de parede: definir a posição alvo para o infinito

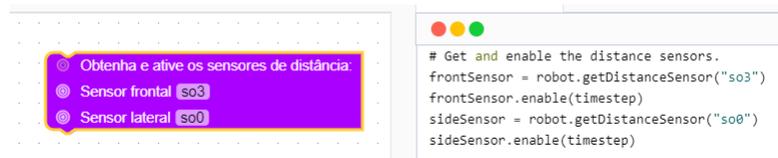


Figura E.43: Pioneer 3D-X Seguidor de parede: obter a ativar os sensores de distância

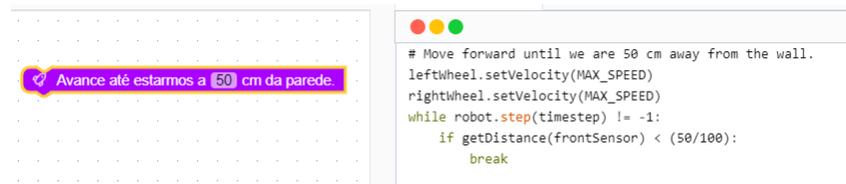


Figura E.44: Pioneer 3D-X Seguidor de parede: avançar até estarmos a N cm da parede

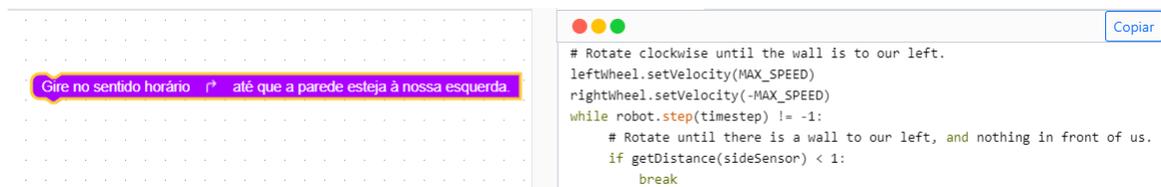


Figura E.45: Pioneer 3D-X Seguidor de parede: gire no sentido horário até que a parede esteja a nossa esquerda

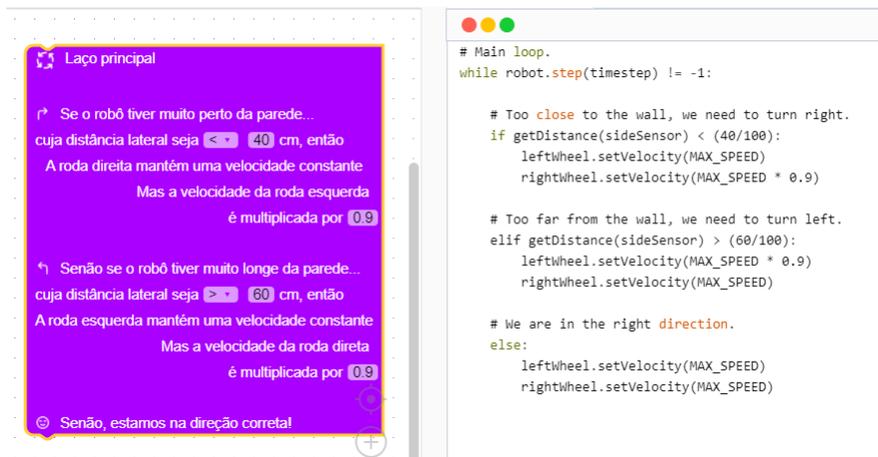


Figura E.46: Pioneer 3D-X Seguidor de parede: laço principal onde a lógica é definida

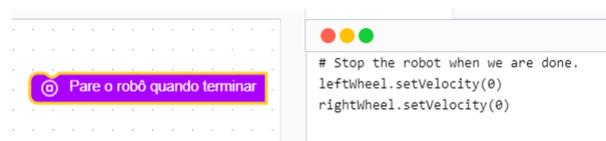


Figura E.47: Pioneer 3D-X Seguidor de parede: pare quando o robô terminar