



**UNIVERSIDADE ESTADUAL DO SUDOESTE DA BAHIA**  
**DEPARTAMENTO DE CIÊNCIAS EXATAS E TECNOLÓGICAS**  
**BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO**

**PABLO SANTOS DA SILVA**

**ROBÓTICA EDUCACIONAL: PROPOSTA DE UMA ARQUITETURA**  
**DE HARDWARE PARA ENSINO DE ROBÓTICA COM BLOCOS**  
**FÍSICOS**

**Vitória da Conquista**

**2019**

**Pablo Santos da Silva**

**ROBÓTICA EDUCACIONAL: PROPOSTA DE UMA ARQUITETURA  
DE HARDWARE PARA ENSINO DE ROBÓTICA COM BLOCOS  
FÍSICOS**

Trabalho de conclusão de curso, apresentado ao curso de Ciência da Computação, da Universidade Estadual do Sudoeste da Bahia, em Vitória da Conquista – Bahia, como requisito parcial para a obtenção do título de Bacharel em Ciência da Computação.

Orientador: Prof. Ms. Gidevaldo Novais dos Santos

Coorientador: Prof. Dra. Alzira Ferreira da Silva

Vitória da Conquista

2019

*Dedico este trabalho a minha família e amigos de caminhada, em especial aos meus Pais, Vane e Paulo, a minha Namorada Débora e a meus Avós maternos, Dona Zélia e Seu Durval, que nunca deixaram de acreditar em mim e no meu potencial, e sempre me apoiaram nessa longa jornada!*

# Agradecimentos

*Agradeço a Deus e a Nossa Senhora Aparecida por me ajudar e permitir que eu chegasse até aqui, ele sabe melhor do que ninguém o quão árdua foi a caminhada e sem ele chegaria onde cheguei.*

*Agradeço em especial a minha melhor amiga e namorada Débora, por sempre me apoiar e acreditar no meu potencial em todos esses semestres de curso. É a pessoa mais especial e importante da minha vida, que eu amo ao infinito e além.*

*A meus pais Vane e Paulo, por se esforçarem ao máximo para que eu pudesse chegar aonde cheguei e por toda esperança depositada em mim durante todos esses anos, a meus avós maternos Dona Zélia e Seu Durval por terem contribuído com minha vida acadêmica desde o ensino básico até agora, a minha madrinha Ni, por todos os cuidados e ensinamentos a mim confiados durante minha vida, a meu irmão Paulo Júnior. A todos os outros familiares que de alguma forma contribuíram com minha vida escolar e acadêmica, direta ou indiretamente.*

*Agradeço ao meu orientador professor mestre Gil, a minha coorientadora professora Dra. Alzira. E a todos os outros amigos do corpo docente que me orientaram em algum momento da jornada, professoras Dra. Máisa, Dra. Cátia e Dra. Alexsandra. Aos professores Dr. Hélio, Msc. Marcos Prado, Dr. Fábio e Dr. Roque.*

*Agradeço com muito carinho ao meu Anjo da guarda, Celina, que com toda sua gentileza e amizade fez meu caminho um pouco menos triste nos momentos complicados do curso, sem ela não teria chegado aonde cheguei.*

*Quero agradecer especialmente meus amigos de turma Dalton, Gustavo Lima, Lucas, Matheus Sales e Raphael, foram excelentes anos. Agradeço aos amigos de LinDalva Bernardo, Iago, Laira, Matheus Alves, Matheus Lima, Matheus Thiago, Rodrigo, Wandeson e Yan. Aos amigos de turma Laverty e Marcelo.*

*E por fim, mas não menos importante, um agradecimento especial aos meus sogros Lôra e Devaldo e meu cunhado Bruno, por terem me acolhido tão bem e serem uma segunda família para mim. Quero agradecer também a Marina, uma aluna muito especial que somente com gestos e atitudes me fez perceber que não importa quão grande for a dificuldade, com um pouquinho de perseverança e muito esforço, sempre iremos superar.*

*A todos, os meus mais sinceros agradecimentos, obrigado, de coração.*

*Nunca deixe alguém te dizer que você não consegue fazer algo.*

*Você tem um sonho.*

*Você tem que protegê-lo.*

*Que a força esteja com você.*

*À Procura da Felicidade / Star Wars*

## Resumo

A educação é um campo extremamente fértil para o uso da tecnologia e a crescente evolução da robótica, torna o aprendizado mais dinâmico e motivador, principalmente para crianças e jovens. Porém a robótica e a programação estão longe de serem fáceis de aprender, ainda mais quando se está nos estágios iniciais da adolescência, muito menos são atraentes para a maioria dos alunos. Este trabalho desenvolveu um protótipo de uma arquitetura de hardware para o ensino de programação em blocos e robótica, utilizando dispositivos concretos para um melhor entendimento de crianças e jovens. Incrementando assim, o conceito de estruturas abstratas de forma lúdica e desconstruindo o paradigma que a programação e a robótica são bichos de sete cabeças para aprender. O resultado esperado para o projeto é, uma arquitetura de hardware completa para o ensino.

**Palavras-chave:** Robótica educacional. Robótica. Educação. Programação. Aprendizagem.

# Abstract

Education is an extremely fertile field for the use of technology and the growing evolution of robotics makes learning more dynamic and motivating, especially for children and young people. But robotics and programming are far from easy to learn, especially when you are in the early stages of adolescence, much less attractive to most students. This work developed a prototype of a hardware architecture for the teaching of block programming and robotics, using concrete devices for a better understanding of children and young people. Incrementing the concept of abstract structures in a playful way and deconstructing the paradigm that programming and robotics are seven-headed creatures to learn. The expected result for the project is, a complete hardware architecture for teaching.

**Keywords:** Educational robotic. Robotic. Education. Programming. Learning.

# Lista de Figuras

Figura 1 – Informações sobre o Scratch. ....	19
Figura 2 – Interface do Scratch.....	20
Figura 3 – Plataforma Blockly.....	21
Figura 4 – Comparação entre linguagens de programação de um mesmo bloco construído pelo Blockly. ....	21
Figura 5 – Interface inicial do Code.org.....	23
Figura 6 – Tela de criação do Code.org.....	23
Figura 7 – Diagrama de funcionamento do App Inventor.....	24
Figura 8 – Tela de criação do App Inventor.....	25
Figura 9 – Placa Makey Makey, visão frontal.....	27
Figura 10 – Placa Makey Makey, visão traseira.....	28
Figura 11 – Arduino Uno. ....	29
Figura 12 – Sensores e Módulos do Arduino. ....	29
Figura 13 – Exemplo de um kit Arduino.....	30
Figura 14 – Kit Lego NXT disponível em: ....	30
Figura 15 – Tipos de padrões de arquitetura para diferentes tipos de abordagens.....	34
Figura 16 – Diagrama de casos de uso de como irá funcionar a aplicação.....	37
Figura 17 – Interface da plataforma LARA.....	38
Figura 18 – Protótipo inicial da interface do bloco. ....	40
Figura 19 – Arquitetura de hardware proposta para o dispositivo físico.....	42
Figura 20 – Bloco Frente.....	47
Figura 21 – Bloco Trás. ....	47
Figura 22 – Bloco Cima. ....	48
Figura 23 – Bloco Baixo.....	48
Figura 24 – Bloco número 0 (Zero).....	48
Figura 25 – Bloco número 1 (Um). ....	49
Figura 26 – Bloco número 2 (Dois).....	49
Figura 27 – Bloco número 3 (Três). ....	49
Figura 28 – Bloco número 4 (Quatro). ....	50
Figura 29 – Bloco número 5 (Cinco).....	50
Figura 30 – Bloco número 6 (Seis).....	50
Figura 31 – Bloco número 7 (Sete). ....	51



Figura 32 – Bloco número 8 (Oito). .....	51
Figura 33 – Bloco número 9 (Nove).....	51
Figura 34 – Bloco Se. ....	52
Figura 35 – Bloco Senão. ....	52
Figura 36 – Bloco Enquanto.....	52
Figura 37 – Bloco Início.....	53
Figura 38 – Bloco Fim.....	53

# Lista de quadros

Quadro 1 – Comparativo entre as tecnologias da área da robótica e programação em blocos.32

## Lista de Abreviaturas e Siglas

Adroid	Sistema operacional de smartphones variados
C/C++	Linguagem de programação
IDE	Integrated Development Environment
IOS	Sistema operacional de smartphones da Apple
LED	Light Emitting Diode
USB	Universal Serial Bus
ZDP	Zona de Desenvolvimento Proximal

# Sumário

<b>1 Introdução .....</b>	<b>11</b>
1.1 Objetivos .....	13
1.1.1 Objetivos Específicos .....	13
1.2 Justificativa .....	13
1.3 Estrutura do documento .....	13
<b>2 Ensino de programação .....</b>	<b>14</b>
2.1 Aprendizagem .....	14
2.2 Desenvolvimento cognitivo .....	15
2.3 Formação de conceito .....	16
2.4 Ensino de programação .....	17
2.5 Programação em blocos .....	18
2.5.1 Scratch .....	19
2.5.2 Google Blockly .....	20
2.5.3 Code.org .....	22
2.5.4 App Inventor .....	23
<b>3 Robótica educacional.....</b>	<b>25</b>
3.1 Robótica .....	25
3.2 Robótica Educacional .....	26
3.3 Tecnologias existentes .....	27
3.3.1 Makey Makey .....	27
3.3.2 Arduino .....	28
3.4 Lego Mindstorms .....	30
<b>4 Metodologia .....</b>	<b>Erro! Indicador não definido.</b>
<b>5 Desenvolvimento da arquitetura do hardware .....</b>	<b>32</b>
5.1 Arquitetura .....	33
5.2 Levantamento de requisitos .....	35
5.2.1 Diagrama de casos de uso .....	37
5.2.2 LARA .....	37
5.3 Identificação dos materiais utilizados para construção dos blocos.....	38

5.4 Identificação da estrutura de hardware para o controle .....	41
5.5 Conjunto final das peças .....	42
<b>6 Conclusão .....</b>	<b>44</b>
6.1 Trabalhos futuros .....	44
<b>Referências .....</b>	<b>45</b>
<b>Apêndices.....</b>	<b>47</b>

# 1 Introdução

A educação é um campo extremamente fértil para o uso da tecnologia e a crescente evolução da robótica. Torna o aprendizado mais dinâmico e motivador, principalmente para crianças e jovens. A robótica educacional tem se destacado no campo tecnológico, possibilitando aos estudantes aprimorar certas habilidades e competências no âmbito social, como o trabalho em equipe, trabalho de pesquisas, resolução de problemas utilizando o raciocínio lógico, entre outros (ZILLI, 2004).

É inegável dizer que houve mudanças nas necessidades de aprender do homem. Dia após dia descobertas são realizadas nas mais diversas áreas do conhecimento, impondo às pessoas obrigações de se aprender de uma forma mais rápida e dinâmica e manterem-se atualizadas, principalmente acerca de suas profissões (ZILLI, 2004).

Zilli (2004) cita Litwin (1997), apontando que quando os estudantes têm à sua disposição a tecnologia, seus objetivos são progredir as habilidades individuais cognitivas e estéticas, tudo isso irá depender de como o docente realizará as atividades nos espaços de interação em grupo.

Segundo Papert (1994, p.5),

a habilidade mais importante na determinação do padrão de vida de uma pessoa já se tornou a capacidade de aprender novas situações, de lidar com o inesperado. Isso será crescentemente verdadeiro no futuro: a habilidade competitiva será a habilidade de aprender.

A escola não proporciona todo o aprendizado do aluno, existe toda uma bagagem incluída nas suas fontes, como o ambiente familiar, amigos dentre outros. Criar a conexão entre o aprendizado escolar e as vivências do mundo externo à escola é fundamental e um dos maiores desafios da educação (YUS, 2004).

Futurekids (2004) é citado por Zilli (2004) conceituando que a robótica está muito mais inserida na vida das pessoas do que se imagina. Eletrodomésticos, aparelhos eletrônicos, dentre outros, todos têm a sua parte robô. Máquinas de lavar são robôs que lavam as roupas automaticamente, sanduicheiras esquentam os lanches num piscar de olhos. As máquinas, que cada vez mais estão ficando automatizadas, vieram para facilitar o trabalho do homem.

Por estar constantemente próxima do cotidiano, a robótica pode ser utilizada como uma forte ferramenta na aquisição e difusão do conhecimento. Ela possibilita uma aprendizagem ativa e participativa, na qual o aluno é o sujeito que participa do processo de

construção do conhecimento. Além disso, permite união de vários recursos tecnológicos para o ensino e a aprendizagem de uma maneira lúdica e divertida, dando oportunidades de estimular certas habilidades de engenharia e computação, atividades que são relevantes para o currículo escolar (EXPOENTE, 2004).

O presente trabalho foi motivado principalmente por ser de caráter educacional e que abrange uma área que é bastante interessante, a robótica. Além de ser um campo que vem em crescimento constante na atualidade, a possibilidade de trabalhar com robótica na educação é animadora.

Poder ensinar crianças e jovens de maneira lúdica e prática, é fascinante. A robótica envolve muito mais do que “robôs”, envolve matemática, programação, lógica, criatividade entre outras habilidades. Proporciona também uma melhora de raciocínio, que serve não somente para a robótica, mas para o desenvolvimento em todas as áreas escolares por parte do aluno.

Existem atualmente várias plataformas voltadas para a robótica educacional, conjuntos de montagem que permitem a programação como os kits LEGO; placas compostas por microcontroladores como o Arduino, RaspberryPi, BananaPi, Thinker Board, dentre outras.

Para trabalhar com robótica é necessário também trabalhar com programação, e existe uma grande dificuldade por parte dos iniciantes na programação. A principal barreira na hora de aprender, é entender a abstração, realizar o controle de estruturas abstratas e a programação baseia-se no conceito da abstração, logo o não conhecimento da prática dificulta o aprendizado. Crianças muito jovens ainda não conseguem entender em sua totalidade o conceito de abstração, para elas algo só é real se é definitivamente concreto, a partir da pré-adolescência ela já começa a entender um pouco do que é abstração. Mas o real problema é, como ensinar o controle de estruturas abstratas para quem não entende ou mal entende o que seria a abstração.

Para a resolução do problema, pode pensar em qual seria uma arquitetura de um dispositivo físico para a programação em bloco para o ensino de programação utilizando um carrinho robô. Já que a programação em blocos físicos evitaria inicialmente esse contato com a abstração.

## 1.1 Objetivos

O objetivo geral desse trabalho é definir o protótipo de uma arquitetura de hardware para um dispositivo físico para ensino de programação em bloco para crianças de 6 a 12 anos.

### 1.1.1 Objetivos Específicos

- Realizar um levantamento de requisitos.
- Identificar materiais para a construção dos blocos.
- Identificar estrutura de hardware para o controle.

## 1.2 Justificativa

Crianças um pouco mais velhas, geralmente a partir dos 12 anos de idade começam a perceber de maneira básica a abstração, e no ensino da robótica se faz necessário o conhecimento de programação. Existem alguns tipos de linguagens que buscam facilitar esse entendimento e aprendizagem, como por exemplo, as linguagens de programação em bloco que facilitam o ensino/aprendizagem e são utilizadas principalmente para o ensino de crianças e jovens que estão iniciando no ramo da programação.

Aprender programação, no entanto, não é uma tarefa tão simples para as crianças menores de 12 anos, visto que até mesmo as linguagens mais simples, exigem no mínimo um conhecimento para diferenciar e entender o que é concreto e o que é abstrato. Crianças mais jovens não conseguem entender essa diferença.

Diante dessas dificuldades foi proposto o desenvolvimento de uma arquitetura de hardware que vai permitir que crianças consigam programar um carrinho de forma física, posicionando os blocos de acordo com o que for proposto pelo professor para realizar o movimento. Possibilitando assim um melhor entendimento por parte das crianças mais jovens de como funciona a abstração da programação.

## 1.3 Estrutura do documento

Além desse primeiro capítulo, o trabalho estará dividido em:

- Nos capítulos 2 e 3 está o referencial teórico que serviram para dar embasamento teórico ao trabalho.
- No capítulo 4 estará a metodologia.
- O capítulo 5 conterá o desenvolvimento da arquitetura de hardware.



- Caberá ao capítulo 6 a conclusão do trabalho e a apresentação de trabalhos futuros.

## **2 Ensino de programação**

### **2.1 Aprendizagem**

As formas de se aprender estão em constante evolução. A forma de transmitir e adquirir o conhecimento não é mais a mesma de 20, 30, 40 anos atrás, as tecnologias evoluíram e com elas a educação, modificando a forma de aprender das pessoas.

“A aprendizagem é uma atividade contínua, iniciando-se nos primeiros minutos da vida e estendendo-se ao longo dela” (LEITE et al, 2009, p.204). O conceito de aprendizagem precisa ser expandido, não pode ficar restrito somente ao ambiente escolar e deve ocorrer tanto na infância quanto na vida adulta. A escola é um dos muitos ambientes em que ocorrem o conhecimento (LEITE et al, 2009, p.204).

Para quebrar o paradigma que o aprendizado é somente no modelo tradicional de professor falando e aluno ouvindo, se faz necessária uma mudança por parte dos educadores, eles precisam se atualizar sobre as pesquisas na área da aprendizagem e propiciar aos alunos oportunidades para gerar e não somente consumir o conhecimento. O que desenvolverá capacidades para continuar a aprender pelo resto da vida (LEITE et al, 2009). Ainda segundo Leite et al (2009) para que o aprendizado não se torne uma atividade individualista, a prática pedagógica precisa ser mediadora, comprometida, coerente e ao mesmo tempo consciente e competente.

A teoria de Vygotsky propõe que é de extrema importância compreender que a aprendizagem impulsiona o desenvolvimento e que é um processo contínuo que está intrinsecamente interligado com as relações sociais (FERRARI, 2014). Para perceber melhor o aprendizado nas crianças, é preciso se atentar a dois fatores, o que a criança pode fazer sozinha e o que a criança faz a partir de ajuda, pistas e orientações externas (FERRARI, 2014).

Diante do exposto, Vygotsky (1984) diferenciou 3 tipos de desenvolvimento da aprendizagem, desenvolvimento potencial, desenvolvimento real e desenvolvimento proximal.

- Zona de desenvolvimento potencial, é toda atividade ou conhecimento que uma criança ainda não sabe, mas que se espera que ela independente da cultura que está inserida possa realizar (FERRARI, 2014).

- Zona de desenvolvimento real, é tudo aquilo que uma criança é capaz de realizar sozinha. “Processos mentais que já se estabeleceram; ciclos de desenvolvimento que já se completaram” (LEITE et al, 2009, p.206).

- Zona de desenvolvimento proximal que será mais bem explicada mais à frente.

Segundo Ferrari (2014) é importante destacar o nível de desenvolvimento real e o nível de desenvolvimento potencial. Essa distância entre o real e o potencial configura a zona de desenvolvimento proximal que é onde ocorre o aprendizado, “pois o que hoje a criança faz com ajuda, amanhã fará sozinha” (FERRARI, 2014, p.30).

## 2.2 Desenvolvimento cognitivo

Antes de falar do desenvolvimento cognitivo, se faz necessária uma breve introdução sobre o que é a cognição. A cognição pode ser descrita como um conjunto de habilidades cerebrais que são necessárias para a obtenção de conhecimento. Essas habilidades envolvem desde o pensamento até a resolução de problemas, passando pelo raciocínio lógico, criatividade, memória, atenção, entre outros.

A cognição, portanto, são os processos cognitivos que nos acompanham desde os primórdios da infância, até a idade mais avançada, o que nos leva a perceber que o desenvolvimento está diretamente ligado ao aprendizado, ou seja, um não ocorre sem o outro.

Podemos então definir que o desenvolvimento cognitivo é: um processo no qual os indivíduos adquirem um conhecimento sobre o mundo ao longo da vida.

O desenvolvimento ocorre gradualmente ao longo da vida e tem diferentes fases de acordo com os fatores a que o ser humano é exposto. Uma criança de 5 anos por exemplo tem uma forma diferente de progredir de um adulto com 43 anos, mas independentemente da idade, todos estão em constante processo de desenvolvimento cognitivo (FERRARI, 2014).

Vygotsky tem para si que a sociedade e a vida material são responsáveis por alterações na forma de pensar e desenvolver do humano, para ele o desenvolvimento cognitivo tem como base principal a sociedade ao qual o ser humano está inserido e suas culturas, de forma que alterações na sociedade e na cultura local afetam a forma da construção e evolução do desenvolvimento. É um processo construído de fora para dentro (FERRARI, 2014). Ainda segundo (FERRARI, 2014, p.27):

a criança nasce apenas com as funções cognitivas elementares que se ampliam para as funções complexas a partir do contato com a cultura, o que

não acontece automaticamente, mas sim por meio de intermediações de outros sujeitos, sendo essas intermediações responsáveis por formar significados e valores sociais e históricos.

O desenvolvimento mental ocorre de formas diferentes, duas pessoas jamais terão um desenvolvimento cognitivo iguais. Os gêmeos, por exemplo, na maioria das vezes eles são tratados da mesma forma, dividem os mesmos ambientes e tem o mesmo tipo de educação, mas ainda assim são totalmente diferentes na forma de se desenvolverem. A unicidade de cada ser ocorre a partir de como o mundo pode ser experienciado, “a cultura torna-se parte da natureza humana, num processo histórico, que ao longo do desenvolvimento da espécie e do indivíduo, molda o funcionamento psicológico do homem” (OLIVEIRA, 1992, p.24).

Vygotsky identificou níveis de desenvolvimento, um real, já adquirido ou formado, que determina o que uma pessoa é capaz de fazer por si própria, e um potencial, ou seja, a capacidade de aprender com outra pessoa e o proximal que é a distância entre o real e o potencial (SILVA, et al). A aprendizagem se relaciona intimamente com o desenvolvimento que proporciona a abertura das Zonas de Desenvolvimento Proximal (ZDP) que podem ser descritas como sendo a distância entre o desenvolvimento real e o potencial desenvolvimento, onde as interações sociais são as principais responsáveis por este. (SILVA et al).

Ainda segundo Vygotsky, as ZDP podem ser vistas como as zonas que estão em constante processo de maturação. Tais zonas poderiam revelar a dinamicidade do processo de desenvolvimento cognitivo, podendo assim prever o resultado a ser obtido futuramente quando o conhecimento for assimilado, ou seja, poderá mostrar a real evolução, o que uma pessoa poderá fazer sozinha após aprender. As ZDP possibilitam prever o desenvolvimento de uma pessoa ao observar a diferença entre o que ela faz e o que ela pode fazer (SILVA et al).

### 2.3 Formação de conceito

A formação do conceito no processo de desenvolvimento do indivíduo é de fundamental importância. Sobre a formação do conceito, Silva (2009) citou Veer e Valsiner (2001), concluindo que “a percepção e a linguagem são indispensáveis à formação do conceito”.

Vygotsky (1998) citado por Silva (2009, p.19) indica que existem três trajetórias básicas na formação do conceito:

- i) Agregação desorganizada - amontoados vagos de objetos desiguais, fatores percentuais são irrelevantes; predomínio do sincretismo.

ii) Pensamento por complexos - os objetos associam-se não apenas devido às impressões subjetivas da criança, mas também devido às relações concretas e factuais que de fato existem entre esses objetos, podendo, entretanto, mudar uma ou mais vezes durante o processo de ordenação.

iii) Generalização e diferenciação - na terceira fase da formação de conceitos, o grau de abstração deve possibilitar a simultaneidade da generalização (unir) e da diferenciação (separar); essa fase exige uma tomada de consciência da própria atividade mental porque implica numa relação especial com o objeto, internalizando o que é essencial do conceito e na compreensão de que ele faz parte de um sistema.

Vygotsky defendia que os conceitos potenciais eram formados primeiro e que se baseavam na junção de certas propriedades. Somente depois, por volta da adolescência os verdadeiros são consolidados. Porém os conceitos mais elementares não são esquecidos pelos adolescentes, operando predominantemente em muitas áreas do seu pensamento (SILVA, 2009).

Para Vygotsky, existem dois tipos de conceitos, os cotidianos e os científicos. Os conceitos cotidianos podem ser definidos como aqueles em que a criança desenvolve durante o seu processo de desenvolvimento. São formulados à medida que a criança utiliza a linguagem para dar nome a objetos e fatos que estão presentes em sua vida cotidiana. Já os conceitos científicos são considerados por Vygotsky como aqueles que são formados a partir da aprendizagem sistematizada, ou seja, a partir do momento em que a criança se depara com um trabalho escolar (SILVA, 2009). “Os conceitos científicos são todos aqueles que derivam de um corpo articulado de conhecimento e que aparecem nas propostas curriculares, como fundamentais na organização de conteúdos a serem trabalhados com os alunos” (SILVA, 2009, p.19).

#### 2.4 Ensino de programação

A educação é a base de qualquer sociedade, ela depende das variações temporais, ou seja, varia de acordo as exigências de cada época, mas o sujeito principal sempre é o ser humano, que depende do meio cultural e natural em que vive (ZILLI, 2004).

Segundo Coll (1999, p.9):

a educação é um conceito genérico utilizado para designar um conjunto de práticas e atividades mediante as quais, e graças as quais, os grupos sociais promovem o desenvolvimento e a socialização de seus membros e garantem o funcionamento de um dos mecanismos essenciais da evolução da espécie humana: a herança cultural.

As constantes evoluções das formas de educação possibilitaram a sociedade revoluções tecnológicas. Como por exemplo, a revolução industrial que inicialmente trouxe

para a Inglaterra do século XVIII as máquinas, que foram inventadas com o propósito de facilitar a vida humana e poupar tempo de trabalho manual e as máquinas a vapor que retiravam a água acumulada nas minas de ferro e de carvão e fabricavam tecidos.

Desde a época da revolução industrial, as tecnologias não pararam de evoluir e a maior responsável por essa evolução é a educação, que permitiu ao homem criar e moldar o mundo tecnológico.

Em meados do século XX chegou-se à Terceira Revolução Tecnológica, que integrou a ciência, a tecnologia e a produção. Aumentando o uso de novas tecnologias para produzir e fabricar novos produtos para serem consumidos pela sociedade. Esse período tem sido importante também para o desenvolvimento da biotecnologia, da robótica, da genética entre outros avanços.

“A educação é um campo fértil para o uso da tecnologia, tendo em vista a gama de possibilidades que apresenta, tornando a aprendizagem mais dinâmica e motivadora” (ZILLI, 2004, p.7). (YUS, 2004) citado por Zilli (2004) diz que somente a escola não proporciona todo o aprendizado do aluno, tem toda uma bagagem incluída nas suas fontes, como ambiente familiar, amigos, dentre outros. Criar a conexão entre o aprendizado escolar com as vivências do mundo externo à escola, é de suma importância e um dos maiores desafios da educação.

## 2.5 Programação em blocos

Zanetti e Bonacin (2014) salientam que com o uso da robótica pedagógica (robótica educacional) “é possível observar o impacto de instruções realizadas por meio de dispositivos concretos e associar os comandos abstratos aos movimentos e ações do mesmo.” (ZANETTI; BONACIN, 2014, p.1234). Para os iniciantes em programação a maior dificuldade é compreender e saber utilizar as estruturas abstratas, e em crianças essa dificuldade é ainda maior.

A programação em blocos pode ser definida como uma metodologia diferente para o ensino de conceitos de programação, resolução de problemas e desenvolvimento do raciocínio lógico. É um paradigma que auxilia a transmissão das estruturas abstratas para algo mais sólido, concreto. Atualmente existem vários softwares e ferramentas para o ensino da prática, como por exemplo o Scratch, o Google Blockly, Code.org, App inventor, entre outras.

### 2.5.1 Scratch

O Scratch é um software voltado para o ensino de programação em blocos e utiliza blocos lógicos, itens de som e imagens para que o usuário possa desenvolver suas próprias histórias interativas, jogos e animações, além de poder também compartilhar online suas criações. O projeto foi desenvolvido pelo grupo Lifelong Kindergarten no Media Lab do Instituto de Tecnologia de Massachusetts (MIT) nos Estados Unidos, e foi idealizado por Mitchel Resnick, (Scratch Brasil, 2014).

Ainda segundo Scratch Brasil (2014) o projeto foi inicialmente desenvolvido no intuito de abranger jovens entre 8 e 16 anos, mas é utilizado por pessoas de todas as idades atualmente. O Scratch (figura 1) é utilizado em mais de 150 países ao redor de todos os continentes e possui tradução para mais de 40 idiomas além de possuir versões para as principais distribuições dos sistemas operacionais (Windows, Linux e Mac), tudo isso de forma gratuita. O Scratch é uma excelente ferramenta para o ensino e aprendizagem de programação, principalmente para quem está nos primeiros estágios do aprendizado. Funciona bem, de forma lúdica e cumpre com o que propõe no que diz respeito ao ensino da programação.

Escrever programas de computadores é parte essencial da alfabetização da atual sociedade. Quando os usuários aprendem a programar com o Scratch, estão aprendendo não só a programar, como também a serem capazes de desenvolver técnicas de resolução de problemas, projetos de design e comunicação de ideias. O Scratch é amplamente utilizado em várias instituições escolares, desde o primário até a universidade e em várias disciplinas (matemática, física, história) dentre outras aplicações (Scratch Brasil, 2014).



Figura 1 – Informações sobre o Scratch. Disponível em:

<http://www.scratchbrasil.net.br/index.php/sobre-o-scratch/73-conhece-scratch.html>

A interface inicial do Scratch é bastante lúdica e colorida (figura 2), o que prende a atenção do usuário logo de cara. A usabilidade da plataforma é boa e de fácil interação, em poucos cliques o usuário já consegue iniciar um novo projeto e se for um usuário inexperiente ou novato, o software dispõe de um tutorial para guiá-lo nos primeiros passos, e ao fim do mesmo o aluno já estará familiarizado com o Scratch.

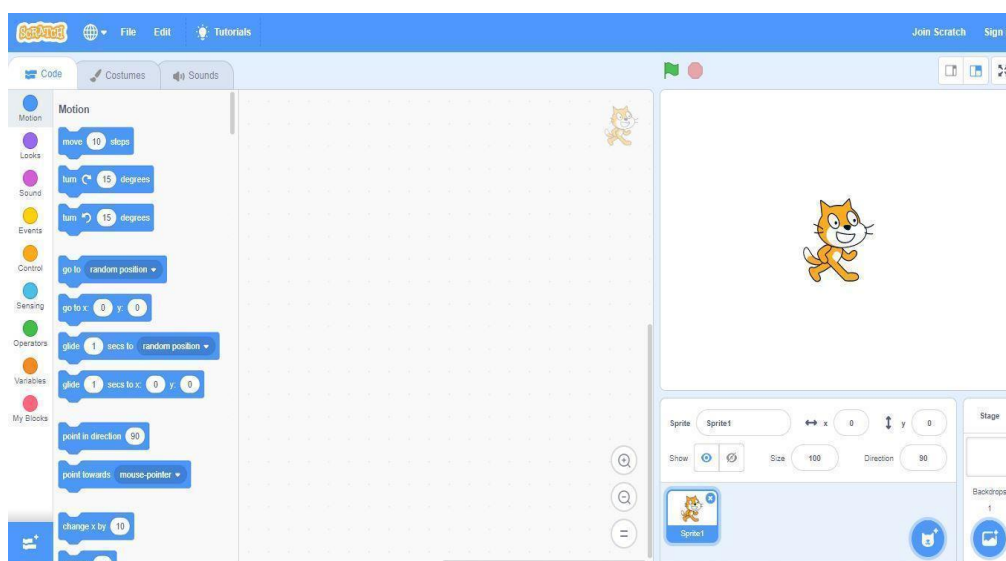


Figura 2 – Interface do Scratch disponível em: <https://scratch.mit.edu/projects/editor/?tutorial=home>

### 2.5.2 Google Blockly

A gigante da tecnologia Google, lançou em meados de 2012 uma plataforma para desenvolvimento baseada em blocos, porém, diferente do Scratch o software da empresa não é voltado para fins educacionais de jovens em idade mais reduzida, seu foco está voltado para a educação de um nível intermediário para avançado, iniciantes terão dificuldades em aprender de cara a utilizar o software. O objetivo da plataforma é permitir o desenvolvimento e criação de aplicativos que evite que o programador utilize códigos e linhas de comando, basta apenas arrastar as peças, como se fosse um quebra cabeças interativo.

O projeto é open source, ou seja, é gratuito e o usuário pode manipulá-lo da forma que achar melhor, além de conter também uma ferramenta que permite a edição do aplicativo desenvolvido na plataforma a partir de outras linguagens de programação, como JavaScript, Python, PHP e etc. A interface do Blockly é bonita e bastante interativa (vide figura 3), porém não trabalha da mesma forma lúdica que o Scratch, o que pode dificultar as coisas para usuários iniciantes no ramo da programação, além de que a plataforma até o presente momento (julho de 2019) não conta com uma versão em português, fato esse que acrescenta

um nível maior de dificuldade, visto que uma boa parte da população não entende perfeitamente o inglês que é o idioma nativo da plataforma.

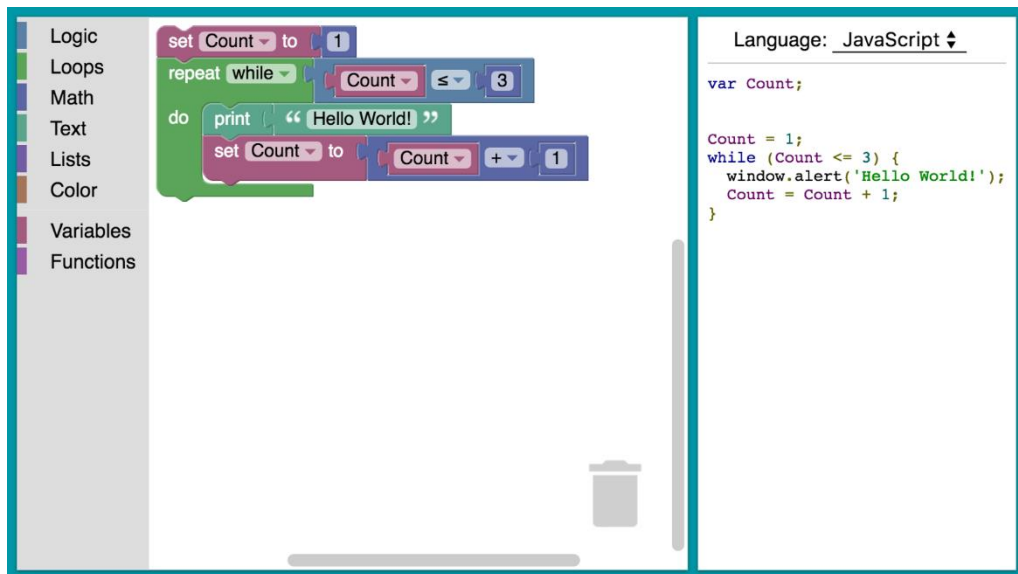


Figura 3 – Plataforma Blockly disponível em: <https://developers.google.com/blockly/>

Uma vantagem do Blockly sobre o Scratch, é que ele tem a possibilidade de o usuário visualizar o que ele está programando em vários tipos de linguagens de programação, como mostra a figura 4. Nela podemos verificar os códigos gerados a partir dos blocos, de 5 linguagens de programação convencionais, JavaScript, Python, PHP, Lua e Dart, respectivamente.

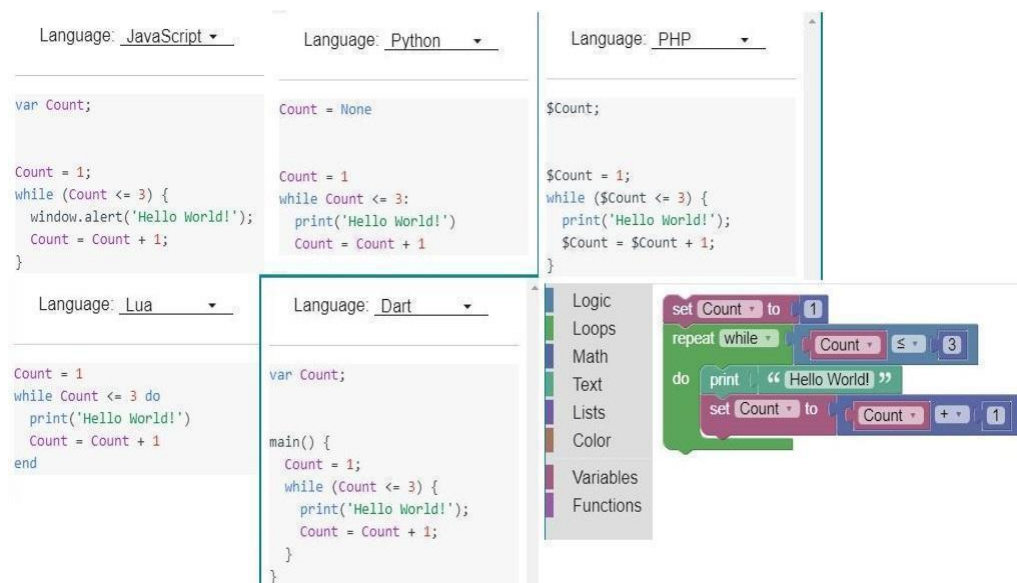


Figura 4 – Comparação entre linguagens de programação de um mesmo bloco construído pelo Blockly.



### 2.5.3 Code.org

“A Code.org é uma organização sem fins lucrativos dedicada a expandir o acesso a ciência da computação em escolas e aumentar a participação das mulheres e das minorias não representadas” (CODE.org, 2019, Sobre). O objetivo da plataforma é que todos os estudantes de todas as escolas tenham a oportunidade de aprender ciência da computação, de tal forma como aprende outras matérias, como química, biologia, matemática, etc. A Code.org é apoiada por gigantes da tecnologia como Amazon, Facebook, Microsoft, entre outras (CODE.org, 2019).

A iniciativa Code.org foi lançada em 2013 pelos irmãos gêmeos Hadi e Ali Partovi, através de um vídeo no Youtube que se tornou número 1 do mundo por um dia e, cerca de 15000 escolas pediram ajuda para implementar o projeto. Desde então a plataforma não para de crescer e se expandir ao redor do mundo e ganhando apoiadores e adeptos famosos, como Bill Gates, fundador da Microsoft, Mark Zuckerberg, fundador do Facebook e outras personalidades. “Acreditamos que a qualidade da educação da ciência da computação deveria estar disponível para todas as crianças, não apenas para algumas” (CODE.org, 2019, Sobre).

Ainda segundo Code.org (2019), todo o trabalho gira em torno da educação, projetando todos os cursos ou fazendo parceria com outros, treinando professores e expandindo internacionalmente via parcerias e fazendo a quebra de estereótipos. Assim como o Scratch, a Code.org tem um alcance internacional amplo, com mais de 40% do tráfego vindo de fora dos Estados Unidos e com mais de 100 parcerias internacionais.

A tela inicial da plataforma disponibiliza um sub menu de criação que contém uma diversidade de modelos disponíveis para criação. Modelos esses que vão desde jogos e aplicativos até jogos com histórias e aplicativos, leituras, desenhos e etc.



Figura 5 – Interface inicial do Code.org

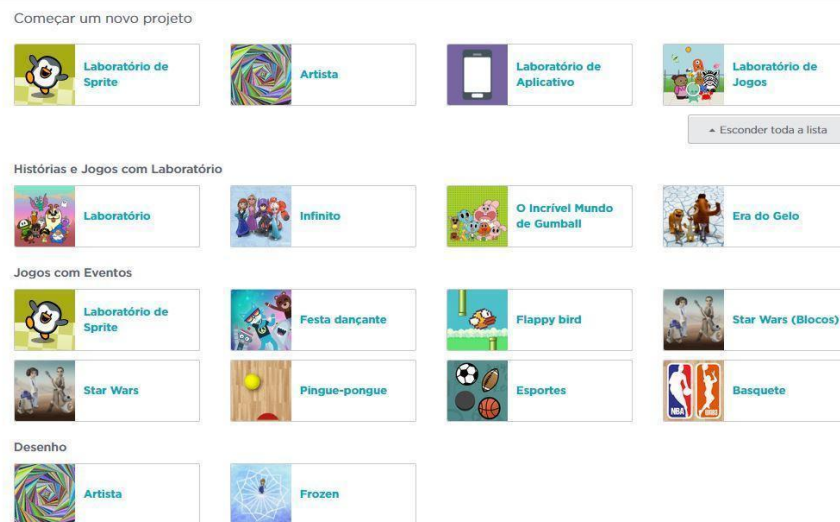


Figura 6 – Tela de criação do Code.org

#### 2.5.4 App Inventor

O App Inventor já é uma ferramenta mais robusta que as anteriores e visa o desenvolvimento de aplicativos para plataforma Android. Assim como o Scratch, ele foi desenvolvido pelo MIT e permite a criação dos aplicativos via navegador web. Todo o processo de criação é intuitivo e feito através de blocos, mas o software não é tão indicado para quem está começando no mundo da programação (CORDEIRO, 2018). Apesar de ser todo programável em blocos, exige uma maior capacidade de raciocínio lógico e habilidade em resolução de problemas para desenvolver um aplicativo, por mais simples que ele seja.



*Figura 7 – Diagrama de funcionamento do App Inventor. Disponível em:  
<https://www.androidpro.com.br/blog/desenvolvimento-android/app-inventor/>*

Caso o usuário não disponha de um smartphone Android, poderá construir os aplicativos e rodá-los utilizando um emulador do próprio App Inventor. A plataforma está disponível para Mac, Linux e Windows e em vários modelos de smartphones, além de que as aplicações criadas com o software podem ser instaladas em qualquer Android (CORDEIRO, 2018). O software está traduzido para o português e sua interface inicial é intuitiva (figura 8), mas complexa para usuários iniciantes.

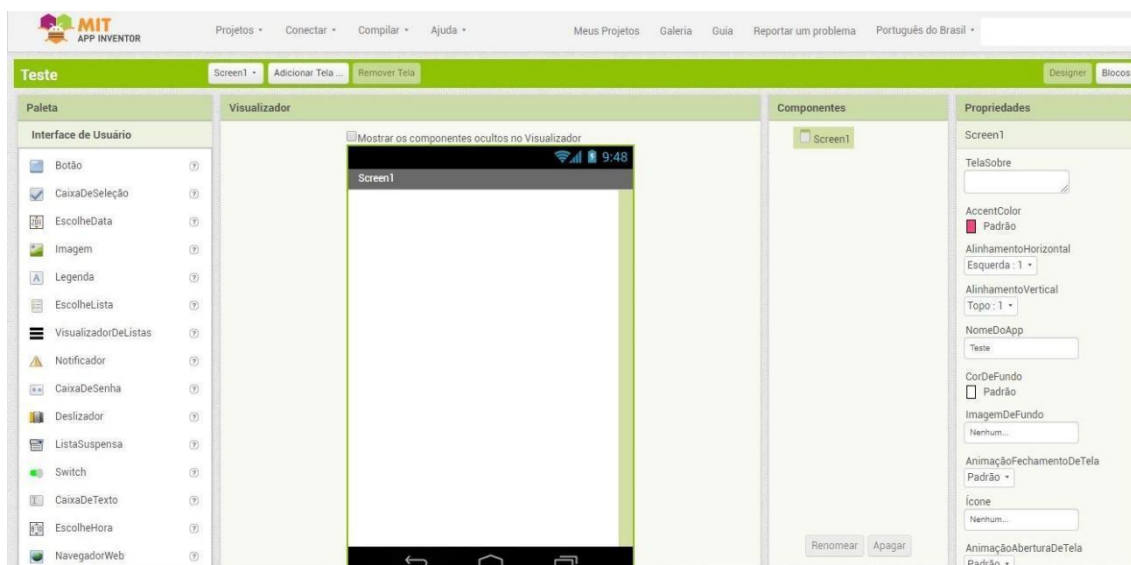


Figura 8 – Tela de criação do App Inventor

## 3 Robótica educacional

### 3.1 Robótica

“A robótica é a ciência ou o estudo da tecnologia associado com o projeto, fabricação, teoria e aplicação dos robôs.” (SILVA, 2009, p.27).

A ideia de ter um ser artificial que age como humano e tem “vida” existe desde a antiguidade, com os egípcios, gregos e judeus, porém a concretização mecânica, elétrica e computacional do robô só veio em meados do século XX.

Segundo Silva (2009) o termo interpretado atualmente, surgiu com o escritor tcheco Karel Capek, em seu romance “R.U.R. (*Robôs Universais de Rossum*)”, em 1921, em que o personagem Rossum projeta e constrói um exército de robôs que ficam inteligentes e acabam dominando o mundo. Silva também aborda sobre o escritor russo-americano Isaac Asimov (1920-1992), que também introduziu o termo robô em seus livros, que mais tarde foram exportados para as telas de cinema como *Eu, Robô* (2004) e *O Homem Bicentenário* (1999).

Contudo a robótica não é ficção científica, é uma ciência em expansão e transdisciplinar e os robôs já estão presentes em nossa sociedade há algum tempo, e envolve desde robôs mais simples como elevadores, caixas eletrônicos até as inteligências artificiais mais avançadas, como as assistentes virtuais do Google Assistant<sup>1</sup> (2016) e a Siri<sup>2</sup> (2011) da Apple (SILVA, 2009).

<sup>1</sup> Assistente virtual inteligente desenvolvido pela Google para smartphones Android.

Zilli (2004) cita (FUTUREKIDS, 2004) dizendo que a robótica está muito mais inserida nas vidas das pessoas do que se imagina. Eletrodomésticos, aparelhos eletrônicos, dentre outros, todos tem a sua parte robô, máquinas de lavar são robôs que lavam as roupas automaticamente, sanduicheiras, esquentam seu lanche num piscar de olhos. As máquinas, que cada vez mais estão ficando automatizadas, vieram para facilitar o trabalho do homem.

### 3.2 Robótica Educacional

“O ambiente de aprendizagem em que o professor ensina ao aluno a montagem, automação e controle de dispositivos mecânicos que podem ser controlados pelo computador é denominado de *Robótica Pedagógica* ou *Robótica Educacional*.” (SILVA, 2009, p.27). Ao longo dos últimos anos a robótica educacional se firmou como uma poderosa ferramenta de ensino e aprendizado para as diversas áreas, não somente curriculares, como do cotidiano (RIBEIRO, COUTINHO, COSTA, 2011).

“A robótica educacional possibilita ao estudante desenvolver habilidades e competências como trabalho de pesquisa, a capacidade crítica o senso de saber contornar as dificuldades na resolução de problemas e o desenvolvimento do raciocínio lógico” (ZILLI, 2004, p.7).

A robótica na educação é bastante importante para o desenvolvimento da criatividade, do raciocínio lógico, e a depender de como ela for trabalhada, desenvolve a habilidade de trabalhar em equipe e o instinto de liderança.

“O casamento entre robótica e educação tem todos os ingredientes para dar certo” (SILVA, 2009, p.31). Primeiro, o robô é o elemento tecnológico que possui vários conceitos científicos e matemáticos, cujos princípios são abordados na escola. Segundo, os robôs mexem com a imaginação, definindo novas formas de interagir e exigindo uma nova maneira de lidar com os símbolos (SILVA, 2009). A robótica educacional envolve vários processos, como a motivação, a colaboração, a construção e a reconstrução, para tal é necessário que se faça o uso de vários conceitos de várias disciplinas para construir os modelos, motivando os alunos a ter uma variada e rica experiência interdisciplinar (SILVA, 2009). Ainda segundo Silva (2009), o robô como uma ferramenta de trabalho dá a possibilidade de novas formas de interagir com o mundo. A aprendizagem é também uma experiência social na qual o aluno interage pela linguagem e pela ação, e essas interações devem proporcionar a cooperação e a autonomia.

---

<sup>2</sup> Assistente virtual inteligente desenvolvido pela Apple para os dispositivos IOS.

### 3.3 Tecnologias existentes

Atualmente existem diversas tecnologias que possibilitam o ensino e aprendizagem de robótica no meio educacional. São plataformas, kits, ou simplesmente microcontroladores que vão desde o ensino de forma básica e lúdica, até maneiras mais complexas e que exigem um pouco mais de esforço para entendimento.

Conjuntos educacionais são os mais simples para iniciantes no campo da robótica, o mais famoso deles são os kits lego, que serão introduzidos de maneira mais completa na próxima subseção. Existem também placas de microcontroladores tais quais Arduino, Raspberry, Tinker board, Banana Pi e algumas menos conhecidas como por exemplo a placa Makey Makey.

#### 3.3.1 Makey Makey

É uma placa que possui embutida um microcontrolador Atmega32u4 e que pode transformar quase qualquer objeto em um controle para computador e abrange desde os usuários iniciantes até os mais avançados (vide figuras 9 e 10).

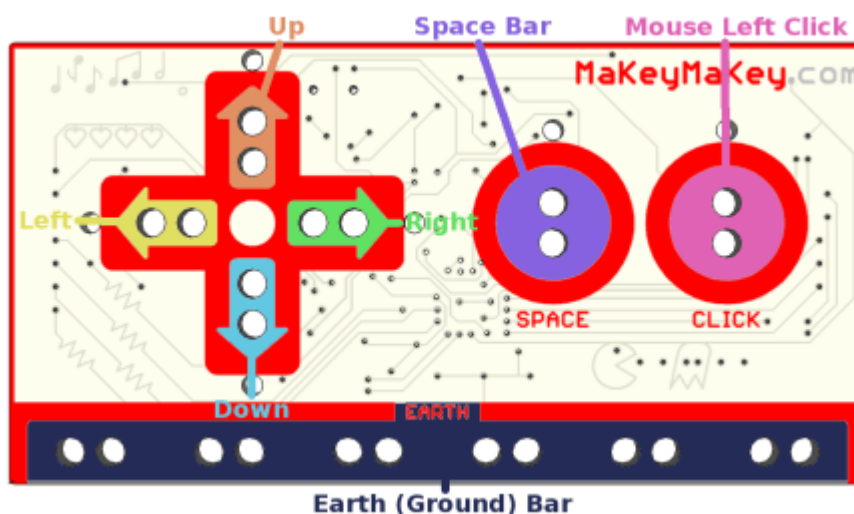


Figura 9 – Placa Makey Makey, visão frontal. Disponível em:

<https://www.embarcados.com.br/make-makey>

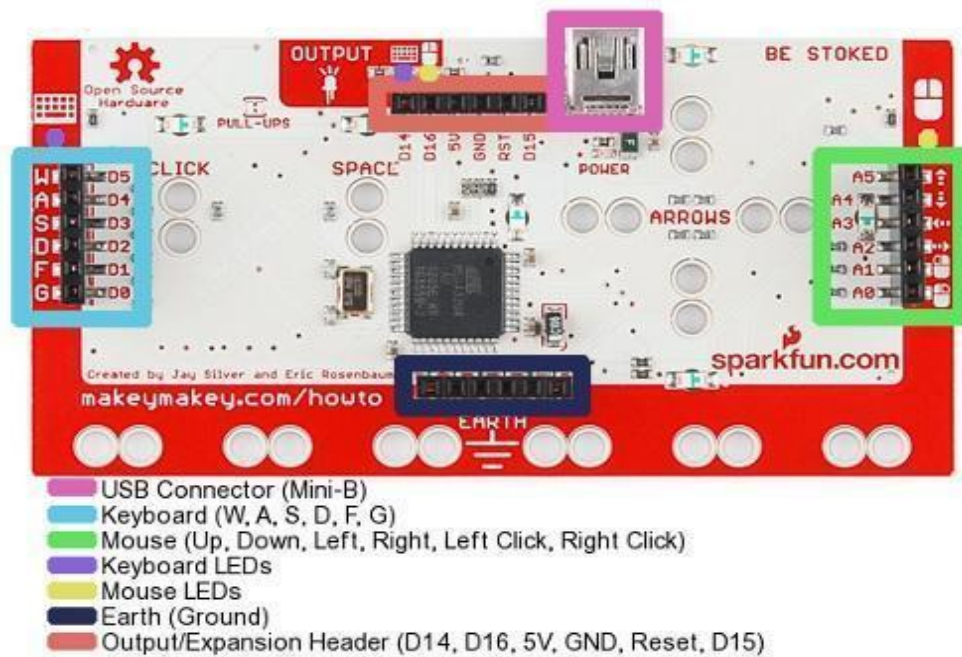


Figura 10 – Placa Makey Makey, visão traseira. Disponível em:

<https://www.embarcados.com.br/makey-makey>

A partir da parte frontal é possível ligar *plugs* do tipo “jacarés” em cada um dos espaços e a outra extremidade em objetos, como por exemplo, bananas ou massinha de modelar, ou até mesmo mergulhar em baldes d’água. Na parte inferior encontra-se o espaço “*Earth*” que serve para o usuário fazer o aterramento, no qual é ligado um jacaré ao espaço e a outra ponta na própria pessoa.

A parte traseira é um pouco mais complexa e é indicada para usuários mais experientes. A placa possui diversas entradas para conexão de jumpers para se trabalhar com entradas analógicas e digitais

A placa Makey Makey é totalmente livre de programação, é só plugar e utilizar.

### 3.3.2 Arduino

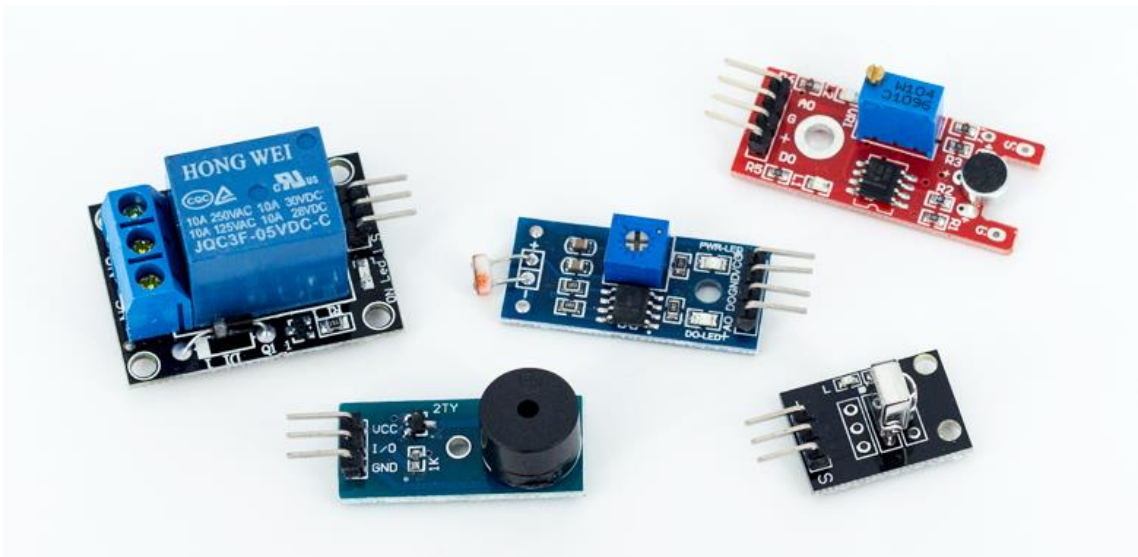
O Arduino é uma placa de baixo custo e hardware livre que possui um microcontrolador do tipo atmel, foi desenvolvido em 2005 e possui circuitos de entrada e saída. O Arduino é conectado ao computador através de um cabo USB e é programado através de um Ambiente de desenvolvimento integrado (IDE), utilizando a linguagem Arduino que é baseada em C/C++.





*Figura 11 – Arduino Uno.*

Na figura 11 pode-se ver um dos modelos mais conhecidos, o Arduino Uno, existem outros, como por exemplo o Arduino Nano, Arduino Leonardo e o mais poderoso deles, o Arduino Mega, que possui mais portas, e um microcontrolador mais potente. Essa plaquinha permite a conexão de vários sensores e módulos para incrementar os projetos, sensores de cor, ultrasônicos, infravermelho, dentre outros, são apenas alguns poucos da grande variedade existente (figura 12).



*Figura 12 – Sensores e Módulos do Arduino.*

A gama de projetos que podem ser desenvolvidos com Arduino é imensa, milhares de coisas podem ser feitas, desde ligar um simples LED, até controlar sua casa ou carro, passando por automação, criação de brinquedos e etc, para tal basta ter um pouco de criatividade e imaginação e um kit Arduino que pode ser visto na figura 13.



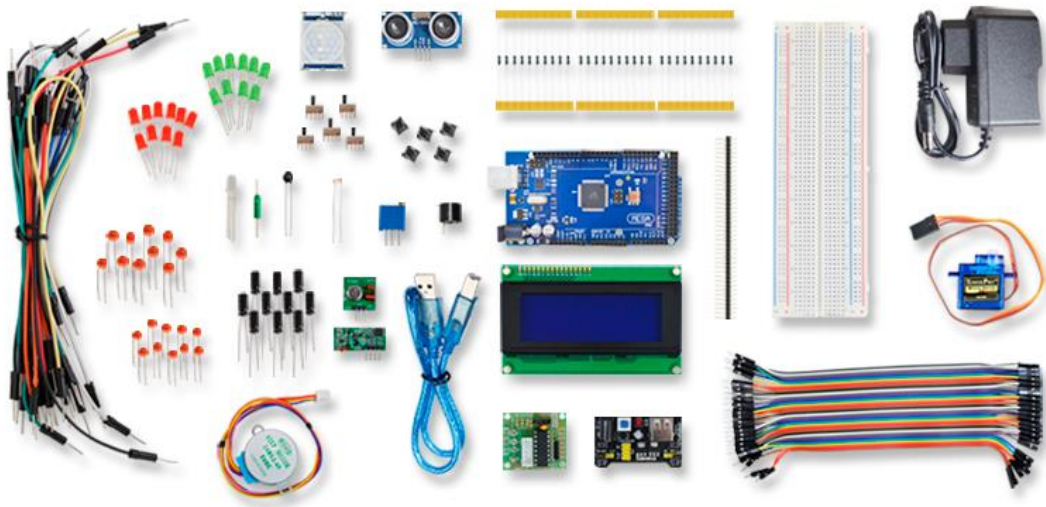


Figura 13 – Exemplo de um kit Arduino.

### 3.4 Lego Mindstorms

“Para o trabalho com robótica, é necessário o uso de equipamentos que favorecem o contato dos alunos com planejamento, construção e controle dos robôs.” (SILVA, 2009, p.34).

Os kits Lego Mindstorms consistem num minicomputador programável que possui motores elétricos, sensores e blocos para a montagem dos robôs, além de possuir peças como engrenagens, eixos, polias, rodas entre outros. (SILVA, 2009).

O Lego tem linguagem de programação proprietária e baseada na programação em blocos, o que o torna mais fácil e lúdica para o ensino de robótica no campo educacional para crianças e jovens que estão começando na área da robótica. O Lego Mindstorms inclui os kits NXT lançado em 2006 (figura 4) e o EV3, mais avançado, lançado em 2013.



Figura 14 – Kit Lego NXT disponível em:

<http://robnaescola.blogspot.com/2012/04/conhecendo-os-kits-lego-mindstorms-nxt.html>

No geral, cada ferramenta tem suas características principais e um público alvo definido, além de trabalhar com diferentes tecnologias como o Arduino e o Makey Makey que são placas microcontroladoras, o Lego Mindstorms que mistura programação em blocos com blocos físicos para montagem dos mais diversos tipos de projetos. O Scratch, Google Blockly, Code.org e App Inventor, trabalham com programação em blocos, cada qual com seu diferencial. No quadro abaixo encontra-se um comparativo entre as tecnologias existentes e que foram citadas neste trabalho.

<b>Plataforma</b>	<b>Principais características</b>	<b>Público alvo</b>
Makey Makey	Facilidade de uso por parte de qualquer pessoa e funcionamento da placa são os principais atrativos.	O público alvo dessa plataforma é principalmente as crianças do ensino fundamental I (4º e 5º ano).
Arduino	O baixo custo do arduino sem dúvidas é a sua principal característica.	Pessoas em geral com que se interessem em conhecer a plataforma e que possua conhecimento mínimo em lógica de programação.
Lego Mindstorms	Os kits da plataforma Lego Mindstorms tem como principal característica o aprendizado lúdico para o ensino de robótica por meio dos seus kits.	O principal público alvo dos kits lego são crianças a partir dos 6 anos de idade até adolescentes de 15 anos.
Scratch	O Scratch se utiliza de imagens e sons para promover um ensino lúdico de programação em blocos.	O público alvo do Scratch é menos restrito, ele é para pessoas de qualquer idade que se interessem por aprender programação de maneira simples e descomplicada.

Google Blockly	A principal característica do Google Blockly além da programação em blocos, é poder ver o código fonte dos blocos em várias linguagens diferentes.	O Google Blockly tem um foco mais educacional, dentro da escola e visa crianças a partir do ensino fundamental I.
Code.org	O Code.org tem uma gama enorme de projetos variados, que vão desde projetos para crianças de 4 anos até adolescentes de 18 anos ou mais.	O Code.org é mais voltado para a educação escolar, com foco em crianças a partir dos 4 anos de idade.
App Inventor	A principal característica do App Inventor é a facilidade que qualquer pessoa tem para construir um aplicativo mobile.	O App Inventor é mais indicado para pessoas que já tenham algum conhecimento de lógica de programação e querem desenvolver aplicativos mobile

*Quadro 1 – Comparativo entre as tecnologias da área da robótica e programação em blocos. Fonte: O autor.*

## 4 Metodologia

A metodologia que foi utilizada aqui é de caráter exploratório. Como existem poucas pesquisas relacionadas ao ensino de robótica utilizando a programação em blocos físicos, coube ao presente trabalho desenvolver um protótipo de uma arquitetura de hardware que possibilite a aprendizagem da robótica por meio de programação em blocos em crianças de 6 a 12 anos, que são o público alvo dessa monografia. Existem poucos trabalhos científicos nessa área, logo as referências das tecnologias existentes foram retiradas dos seus respectivos sites, caracterizando assim, uma pesquisa documental.

A metodologia fundamentou-se em pesquisas exploratórias para obtenção de uma arquitetura de hardware que fosse capaz de suprir as necessidades dos objetivos que foram propostos. Os requisitos para a conclusão da pesquisa que geraram a arquitetura foram levantados por meio do estudo de referências bibliográficas e documentações de ferramentas parecidas. Com os requisitos levantados, o modelo da arquitetura pode ser concluído.

## 5 Desenvolvimento da arquitetura do hardware

Crianças muito jovens, geralmente menores de 12 anos ainda não possuem uma clareza para entender o que é, e como funciona a abstração. Compreender esse tipo de estrutura é fundamental para o aprendizado das linguagens de programação, sendo esse um dos maiores obstáculos no ensino da prática para crianças.

A partir dessa constatação, faz-se necessário o desenvolvimento de uma arquitetura de hardware para um dispositivo físico que visa o ensino da programação em blocos para crianças e jovens adolescentes. A metodologia aqui utilizada foi embasada por meio de alguns estudos já realizados na área e foi desenvolvida por meio do levantamento de requisitos para construção do dispositivo, pela identificação dos materiais que serão utilizados para a construção dos blocos e pela identificação da estrutura de hardware para o controle.

O levantamento de requisitos foi feito através de análises do referencial teóricos e de alguns estudos relacionados como em Zilli (2004) que utilizou a robótica educacional para avaliar como ela influencia no desenvolvimento de certas competências como, raciocínio lógico, resolução de problemas, capacidade de trabalho em equipe e outras, em crianças do ensino fundamental 2 (6º ao 9º ano); Silva (2009), que utilizou uma metodologia baseada na teoria sócio-histórica de Vygotsky, em conjunto com kits Lego e um software educacional para o ensino de robótica para crianças entre 8 e 10 anos. Figueira (2012), que fez uma análise de como as empresas de desenvolvimento de software de Vitória da Conquista – Bahia, utilizam as técnicas de levantamentos de requisitos para desenvolverem seus produtos, além de analisar e perceber que determinadas técnicas facilitam a elicitación dos requisitos e o relacionamento com o cliente.

### 5.1 Arquitetura

Em meados da década de 90, o planejamento de um software ocupava uma parte relativamente pequena do desenvolvimento de um sistema, a maior parte do tempo era gasta com planejamento e testes no hardware, o que acarretava em raríssimos erros. Porém, a maior facilidade e praticidade em alterar um software do que um hardware, fez com que o

planejamento do software ocupasse parte significativa do desenvolvimento dos sistemas. Para tais tarefas surgiram diversas técnicas para o planejamento de softwares. Uma arquitetura de software consiste num modelo de alto nível que possibilita o entendimento e análise mais simples de um software a ser desenvolvido. Além de possuir a possibilidade de usá-la como ferramenta de comunicação entre os *stakeholders*<sup>3</sup> que fazem parte do desenvolvimento do sistema (DEVMEDIA, 2008). Para cada tipo de abordagem existe um tipo de padrão disponível para o planejamento da arquitetura do software.

Abordagem	Foco	Padrões
Programação Estruturada	Sistemas de pequeno porte	Estruturas de controle
Abstração e modularização	Sistemas de médio porte	Encapsulamento e ocultação de informações
Componentes e conectores	Sistemas de grande porte	Estilos arquiteturais

*Figura 15 – Tipos de padrões de arquitetura para diferentes tipos de abordagens. Disponível em: <https://www.devmedia.com.br/arquitetura-de-software-desenvolvimento-orientado-para-arquitetura/8033>.*

A arquitetura de software serve para minimizar os erros do projeto antes de começar a desenvolvê-lo, conseqüentemente reduz o custo financeiro e também economiza tempo que poderia ser perdido refazendo módulos do sistema por conta de erros que seriam identificados se tivesse utilizado a arquitetura de forma correta.

Foi proposta no presente trabalho o desenvolvimento de uma arquitetura de hardware para o desenvolvimento de um dispositivo físico para o ensino de robótica para crianças de 6 a 12 anos fazendo o uso de programação em blocos físicos. É de grande importância que as crianças de hoje aprendam a programar, não somente para desenvolver softwares e aplicações. Aprender a programar é importante pois estimula o desenvolvimento do raciocínio lógico, aprimora a capacidade de resolução de problemas, melhora a aprendizagem, dentre outras vantagens. E a arquitetura é parte importante deste projeto porque é ela quem vai definir todos os detalhes e as especificações da criação do protótipo do dispositivo físico que foi proposto neste trabalho.

---

<sup>3</sup> Termo utilizado na engenharia de software para designar as pessoas que fazem parte e/ou interagem com o sistemas e são afetadas pelo seu desenvolvimento, como clientes, usuários, gerentes de projeto, entre outros Figueira (2012, p.16).

## 5.2 Levantamento de requisitos

Segundo Zanetti e Bonacin (2014), aprender a programar utilizando linguagens de programação convencionais é um processo difícil, que exige um grau muito alto de abstração e compreensão de sintaxes de linguagens complexas. Para isso existem maneiras mais práticas e lúdicas para o ensino da programação para crianças, uma das quais, foi utilizada na pesquisa, programação em blocos. A partir dos estudos analisados foi possível definir quais os requisitos necessários para o desenvolvimento da arquitetura do hardware.

Levantamento de requisitos, ou elicitación de requisitos, pode ser compreendida como a fase em que os desenvolvedores utilizam um conjunto de atividades para descobrir a necessidade dos *stakeholders*, de forma que eles possam encontrar a melhor maneira de aplicar os requisitos essenciais do sistema a ser desenvolvido. O levantamento de requisitos pode ser dividido em quatro etapas: entendimento do domínio da aplicação, entendimento do problema, entendimento do negócio e o entendimento das necessidades e das restrições dos *stakeholders* (FIGUEIRA, 2012).

Realizar um levantamento de requisitos envolve todo um conjunto de tarefas e ações que visam capturar e registrar informações que darão a base para o entendimento total do problema e a consequente especificação dos requisitos (KOURI, 2007). Existe uma dificuldade evidente na obtenção da elicitación de requisitos, visto que inúmeros fatores ocasionam tal dificuldade, como por exemplo a diferença de conhecimento e habilidade dos projetistas que estão à frente do desenvolvimento, experiências distintas dos componentes do grupo e até mesmo a omissão, às vezes involuntária, de características por parte dos *stakeholders* (FIGUEIRA, 2012).

Diversas técnicas para levantamento de requisitos foram desenvolvidas visando solucionar a dificuldade das elicitaciones, que foram citadas anteriormente, e essas técnicas podem ser utilizadas com o objetivo de colher informações como questionários e entrevistas, observações, análises, dentre outros (SOMMERVILLE, 2007).

Obter corretamente os requisitos para o desenvolvimento de um projeto é uma das partes mais importantes, senão a mais importante para que ele seja desenvolvido com sucesso. Para que isso ocorra, é crucial utilizar uma técnica de levantamento de requisitos que encaixe corretamente com o projeto e que todos os membros do grupo possam trabalhar com ela. Algumas das técnicas mais utilizadas atualmente no desenvolvimento de sistemas são, pontos de vista, que levam em consideração as percepções de cada *stakeholder* interessado no projeto

que está sendo desenvolvido; Entrevistas, que como o próprio nome já diz, consiste em realizar entrevistas particulares ou públicas com as partes interessadas, ou seja, *stakeholders*; Questionários, que são perguntas direcionadas a um grande número de pessoas, geralmente o público-alvo para o qual está sendo desenvolvido o sistema; Casos de uso que constitui-se numa notação gráfica de como o sistema irá funcionar. Existe também a técnica de cenários, que foi a utilizada para levantar os requisitos para o desenvolvimento da arquitetura de hardware para o dispositivo físico que irá ensinar robótica, através da programação em blocos físicos.

O cenário de funcionamento do dispositivo foi imaginado e a partir dele fez-se a idealização de cada requisito que o produto final conterà, levando em conta algumas características como, facilidade de uso, durabilidade, baixo custo de produção e de aquisição, dentre outras. No cenário ideal, o dispositivo físico completo consiste em: blocos para a programação, objeto de captura (câmera de smartphone), software para “converter” os comandos físicos em linguagem de programação, robô móvel (carrinho) para visualização dos comandos programados e uma plataforma que integre tudo isso e faça toda a comunicação, desde a captura dos blocos, passando pela conversão do código, envio e visualização da execução dos comandos.

**Blocos:** A construção e funcionamento dos blocos será um subtópico a parte e será explicado mais adiante.

**Smartphone:** Um celular do tipo smartphone é necessário para o completo funcionamento da ferramenta. O usuário montará os blocos de acordo ao desafio proposto pelo educador<sup>4</sup> e com a câmera do celular, o aluno irá tirar uma foto dos blocos perfilados.

**Software:** Após a foto ser capturada, o usuário a enviará para o software conversor que realizará a leitura das tags de cada bloco e converter em uma linguagem de alto nível. Após a conversão o software fica encarregado de enviar o código para o robô que fica em um laboratório remoto.

**Robô móvel:** Para que o aluno possa visualizar aquilo que ele de fato programou, é necessário um robô móvel executando os movimentos.

---

<sup>4</sup> Educador pode ser entendido como qualquer pessoa que propõe um desafio dentro do software, não necessariamente precisa ser um professor.

### 5.2.1 Diagrama de casos de uso

Ainda com relação ao levantamento de requisitos, foi desenvolvido um pequeno diagrama de casos de uso para identificar como que o software irá funcionar no smartphone de cada tipo de usuário. São dois tipos de usuários, um é o educador e pode ser visto como um administrador do sistema, este poderá cadastrar novos desafios para os usuários da plataforma, além de editar seus próprios desafios e excluí-los também. O segundo tipo de usuário é o estudante, que irá analisar os desafios (problemas) para tentar resolvê-los. Ele pode iniciar um desafio, enviar a foto com o código que foi programado em blocos para o sistema e depois tocar em resolver o desafio, e caso a solução esteja correta, ele poderá verificar o funcionamento do carrinho.

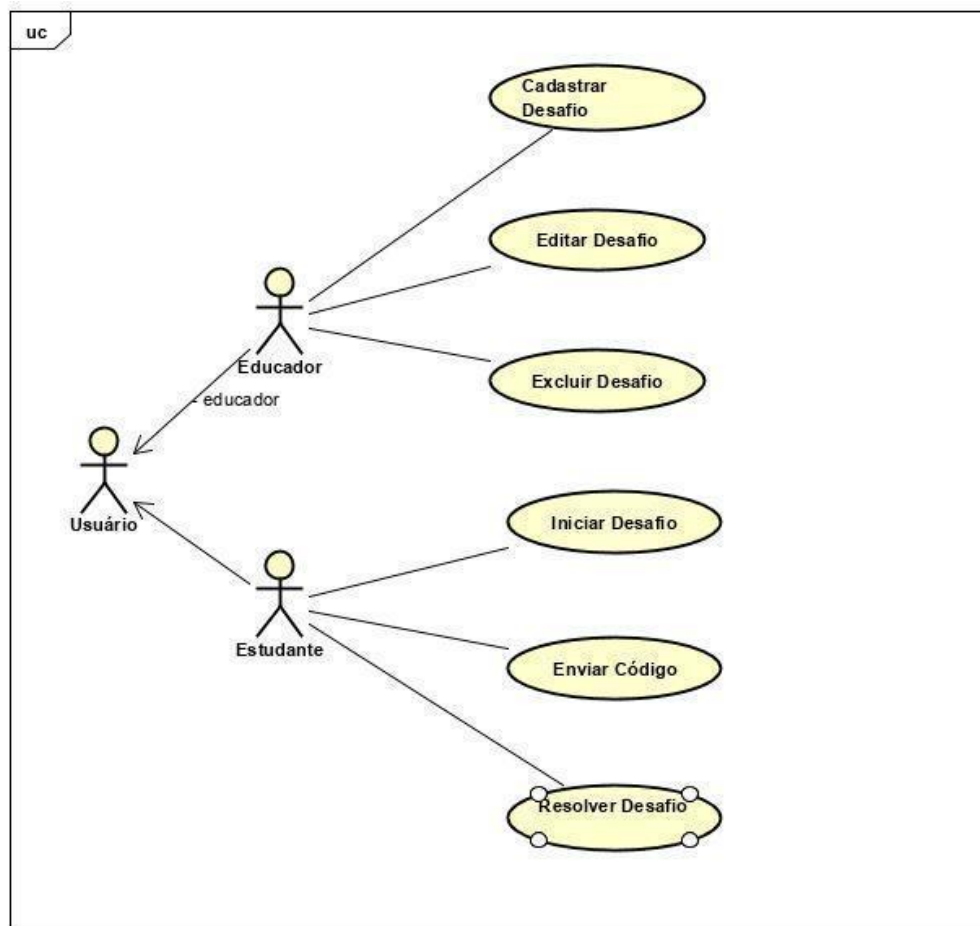


Figura 16 – Diagrama de casos de uso de como irá funcionar a aplicação. Fonte: O autor

### 5.2.2 LARA

Para que o código seja enviado ao carrinho, é necessária uma plataforma que faça essa integração, o Laboratório Remoto em AVA (Ambiente virtual de aprendizagem) – LARA. O LARA é uma plataforma que tem como objetivo “projetar e implementar uma arquitetura



pedagógica que integra recursos tecnológicos e metodologia de ensino para melhorar o processo de aprendizagem” (LOPES, 2017, p.17).

O projeto LARA fica localizado no laboratório LinDalva, na Universidade Estadual do Sudoeste da Bahia, UESB, campus Vitória da Conquista. Na figura 17 pode-se perceber o funcionamento da interface web do laboratório remoto. Está dividido em: **1** barra de ferramentas contendo, criar novo arquivo em branco, compilar, enviar para o robô, fazer upload de um código já existente, fazer download do código, abrir porta serial, ligar/desligar câmera do laboratório, chat, verificar pessoas online, tempo restante de sessão (30 minutos cada) e finalizar sessão; **2** editor de código para programação; **3** material didático disponível para auxiliar o usuário; **4** exemplos de códigos já feitos; **5** visualização do robô que fica no laboratório; **6** console que exibe as informações da compilação, mostra se houve erros e quais foram ou se a compilação foi completada com sucesso; **7** porta serial; **8** chat.

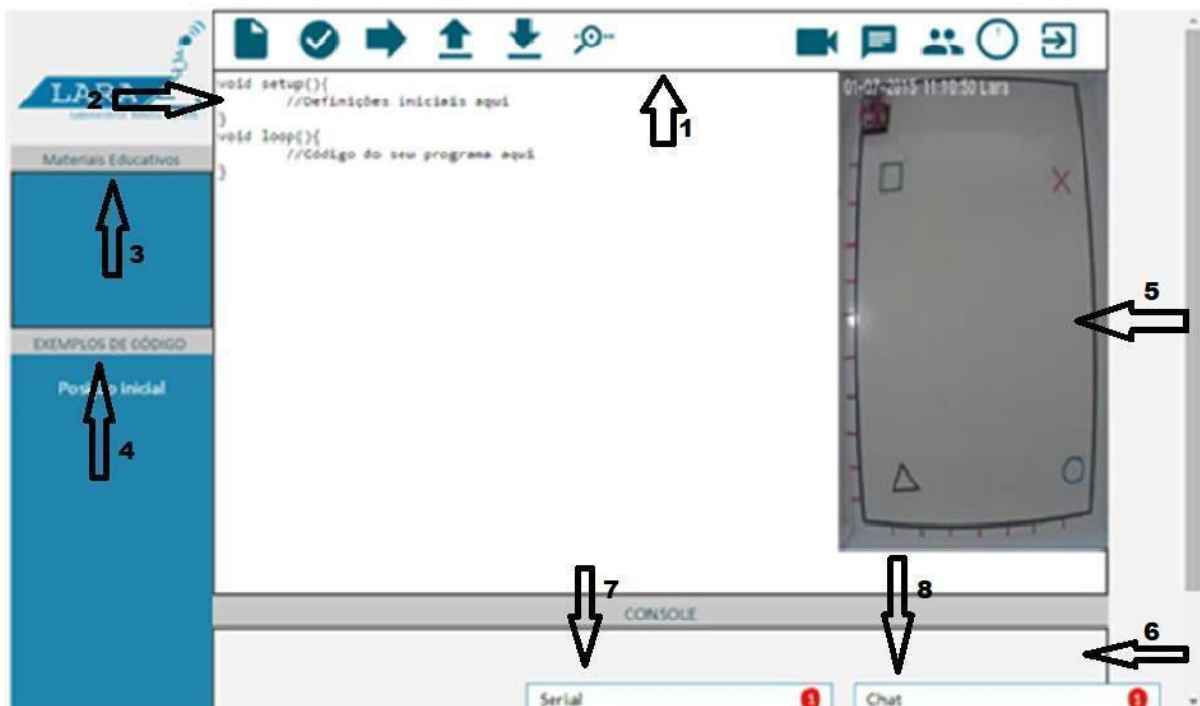


Figura 17 – Interface da plataforma LARA.

### 5.3 Identificação dos materiais utilizados para construção dos blocos

Durante o levantamento de requisitos para a construção dos blocos surgiram alguns pequenos problemas. A proposta de desenvolvimento requereu a utilização de materiais de baixo custo para que o produto seja economicamente viável para que qualquer educador que queira utilizar a premissa possa ter acesso ao dispositivo.

Diante do exposto surgiu o primeiro problema, qual tipo de material seria resistente o suficiente para ser utilizado por crianças e que seria de baixo custo? As alternativas encontradas foram utilizar blocos de madeirite revestidos por um adesivo com os respectivos desenhos de cada bloco programável, blocos feitos de papelão também revestidos pelos adesivos, ou utilizar papel cartão/cartolina com os desenhos dos blocos impressos direto no papel.

- **Madeirite:** Foi cogitada a utilização de blocos feitos de madeirite por conta de sua resistência e durabilidade desde que utilizado e armazenado de maneira correta. Apesar de ser um material de baixo custo, logo foi descartada a ideia. Como é um produto voltado para crianças o tipo de material oferece risco a segurança dos pequenos, visto que por conta de sua dureza pode machucar as crianças ao haver um contato um pouco mais forte, além de correr o risco de soltar lascas da madeira e se prender a pele dos alunos.
- **Papelão:** Usar o papelão resolveria o problema da dureza do madeirite, mas ainda não foi o material ideal. Mesmo que seja mais flexível, não é prático, pois não seria possível imprimir os desenhos dos blocos e tags diretamente na superfície o que dificultaria o processo de montagem.
- **Papel cartão/cartolina:** A solução ideal encontrada foi utilizar papel cartão já que é de baixo custo e facilmente encontrado em qualquer papelaria. Além de ser prático, os blocos podem ser impressos direto em sua superfície. Existem no entanto diversas gramaturas<sup>5</sup> para folhas de papel, incluindo o papel cartão, logo a recomendação para impressão depende do tipo de impressora do usuário. Impressoras jato de tinta que puxam o papel pela parte de trás, podem utilizar papéis com gramatura de até 300 g/m<sup>2</sup>, outros modelos que puxam o papel pela frente, a indicação é que não passe de 200 g/m<sup>2</sup>, pois o papel precisará virar 180 graus dentro da impressora, e se o papel for muito rígido a impressão nesse tipo de impressora fica impossibilitado.

Identificado o material ideal para a construção, foi a vez de pensar em como desenvolver os adesivos e as tags que acompanharão os blocos. Os adesivos serão parte fundamental no projeto, visto que ele será a parte visual do dispositivo que guiará as crianças a resolver os desafios. Cada bloco feito, terá nele impresso algum adesivo contendo um

---

<sup>5</sup> A gramatura é uma unidade de peso, que significa gramas por metro quadrado, ou seja, o peso do papel em função de sua espessura. Na maioria dos casos, quanto maior o peso, maior a espessura do papel

pequeno texto e uma pequena ilustração para o aluno identificar o que cada peça representa, como a peça “frente”, que é representada pela figura 18.

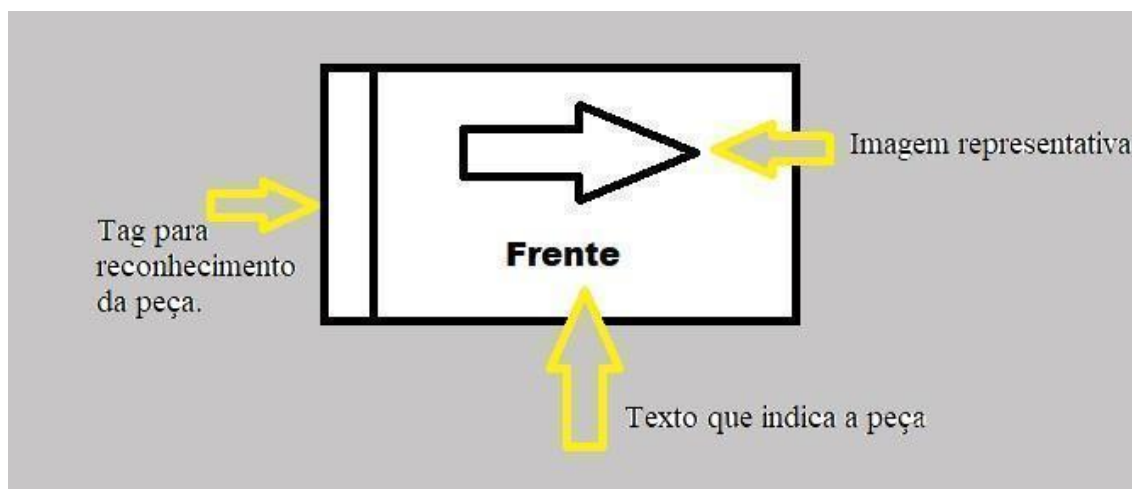


Figura 18 – Protótipo inicial da interface do bloco. Fonte o autor

Para o software fazer o reconhecimento dos blocos é necessário um conjunto de tags espalhados por cada peça. Essas tags foram desenvolvidas utilizando qr code, cada bloco recebe um código específico, que será enviado para o software através da captura da imagem dos blocos perfilados. As tags são facilmente criadas utilizando qualquer ferramenta gratuita online e juntamente impressa com o adesivo no bloco.

**Tags de identificação:** Sendo o dispositivo de fato físico, se fez necessário pensar em soluções para que houvesse uma maneira de reconhecimento dos blocos pela plataforma do LARA.

A solução para o reconhecimento foi a utilização de tags feitas com qr code. Qr (em inglês *quick response* “resposta rápida”) code, é um código de barras bidimensional que tem a finalidade de armazenar informações, como contatos telefônicos, urls, textos, etc. O código é facilmente escaneado por qualquer smartphone que tenha uma câmera e convertido para qualquer que seja a função embutida no código. Cada bloco recebe uma tag específica e a identificação de cada bloco pelo software fica nos qr codes.

**Adesivos:** O princípio da programação em blocos é poder ensinar a programação de forma lúdica e prática. Diversas técnicas são utilizadas pelas plataformas digitais que ensinam esse tipo de programação, o presente trabalho não está limitado somente aos meios digitais para o ensino, portanto outro meio foi utilizado. Como os blocos da programação são físicos a solução para o desenvolvimento se deu pela confecção de adesivos.

#### 5.4 Identificação da estrutura de hardware para o controle

Levando em consideração os tipos de arquiteturas de softwares, foi proposta para este trabalho uma arquitetura em camadas, na qual o projeto é dividido em camadas, cada qual com sua particularidade e ao final ocorre a montagem final, unindo todas as camadas na devida ordem. O projeto foi dividido em três camadas principais e duas subcamadas.

A primeira camada é a de captura, a qual é responsável por capturar via foto a programação dos blocos. Após a colocação dos blocos por parte do estudante, ele pega um smartphone que tenha o software de captura instalado tira a foto do código escrito com os blocos e envia para o servidor que é a segunda camada da arquitetura.

A camada do servidor, é dividida em duas subcamadas, a subcamada de reconhecimento e a de montagem do código. A primeira é a que recebe a foto enviada do smartphone, ela reconhece os blocos e confirma se o código está correto na parte sintática e ao final do processo, repassa para a segunda subcamada, que é a montagem do código, essa montagem é a conversão da foto em código puro de alto nível<sup>6</sup>, e a compilação e envio das tarefas a serem executadas pelo carrinho.

A terceira e última camada é a camada de exibição, como o próprio nome já diz, ela é quem vai exibir o carrinho executando as ações programadas pelo usuário com os blocos no início do processo. Ao final desse processo se conclui a execução total da atividade que foi proposta pelo educador.

A arquitetura de software foi apresentada baseada no levantamento de requisitos feito anteriormente e para total conclusão do trabalho, foi necessário mesclar a arquitetura do software com os componentes requeridos de hardware. Os blocos físicos impressos, serão a parte mais importante de todo o dispositivo, a partir deles o estudante desenvolverá métodos para a resolução dos problemas propostos pelo educador, a fim de conseguir visualizar no final da tarefa, a execução dos movimentos que ele programou para solucionar o desafio. O smartphone com câmera e o software instalado irá realizar a captura da imagem dos blocos programados pelo estudante, em superfície limpa e clara para não atrapalhar na leitura das tags de identificação.

Um computador com um servidor funcionando é necessário para que o dispositivo funcione corretamente vinte e quatro horas por dia, sete dias por semana. Nessa máquina

---

<sup>6</sup> Linguagens de alto nível são aquelas que se parecem com a linguagem natural de fato, que são mais próximas ao alfabeto real.

servidor estará rodando o laboratório virtual do LARA que será o responsável por parte da subcamada de reconhecimento e pela camada de montagem do código e pela exibição para o usuário. O carrinho para ver na prática o que o estudante programou será o mesmo utilizado nos experimentos do laboratório, L1R2 (Lara Remote Robot).

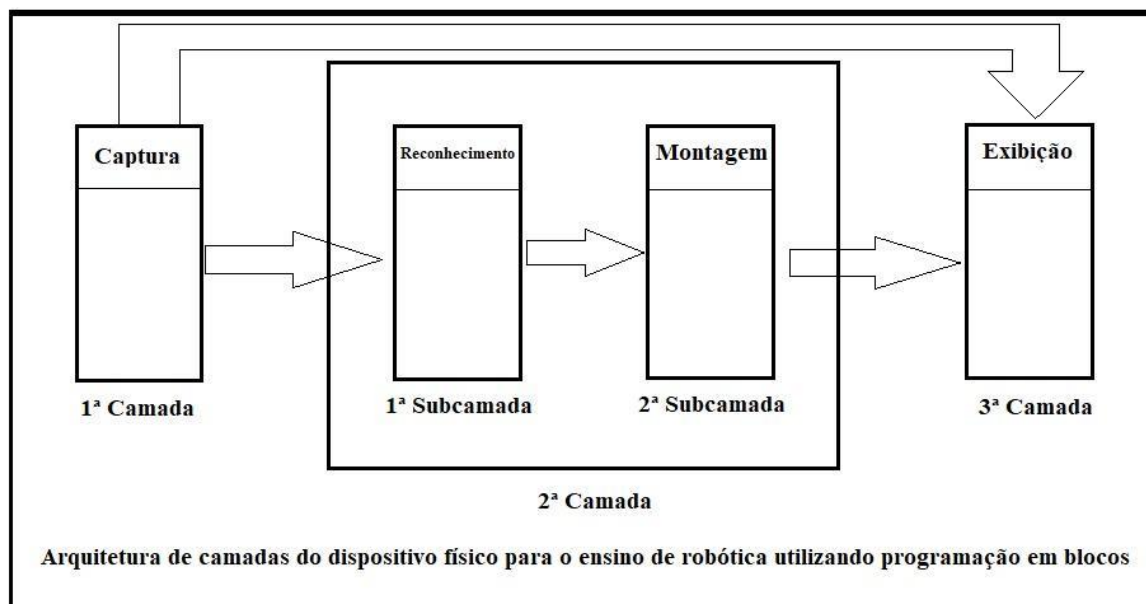


Figura 19 – Arquitetura de hardware proposta para o dispositivo físico. Fonte: O autor.

A figura 19 indica o projeto final de como a arquitetura em camadas deverá ser. A camada de captura se conecta diretamente com a camada de exibição e internamente com a segunda camada, isso significa que o usuário só enxerga a primeira e terceira camadas, da captura ele vê a exibição do que acabou de programar.

Internamente, ou seja, o processo fica invisível ao usuário, a camada de captura se conecta com a segunda camada que, em primeiro lugar passa pela subcamada do reconhecimento, a qual reconhece as tags dos blocos capturados pela primeira camada e se estiver sintaticamente correto encaminha para a próxima subcamada que irá fazer a conversão para a linguagem de alto nível e enviará o código convertido para a terceira camada que por sua vez irá exibir os movimentos para o usuário.

### 5.5 Conjunto final das peças

O conjunto final do dispositivo físico contará com 37 blocos, sendo eles:

- **Início e Fim (2 peças):** Os blocos de início e fim servem para demarcar o escopo do projeto, contém uma peça de cada são usadas para iniciar e finalizar

o programa respectivamente.

- **Blocos direcionais (20 peças):** São os blocos para frente e para trás, para a cima e para baixo. Tem como função indicar qual direção o carrinho irá seguir e são cinco peças de cada direção.
- **Blocos numéricos (20 peças):** Os blocos numéricos estão divididos em blocos que vão de zero a nove e tem como função, indicar quantidade, seja ela de movimentos ou para utilização nas estruturas condicionais ou de repetição. São duas peças de cada número.
- **Blocos condicionais (10 peças):** Os blocos condicionais são os indicadores de condição “se” e “senão”. Determinam ação de acordo com a condição exposta na programação, ex: “Se 3 for maior que 5, ande para frente 5 vezes, senão ande para trás 5 vezes”. São cinco peças de cada.
- **Blocos de repetição (5 peças):** Os blocos de repetição são os chamados loops, que tem como objetivo repetir determinada ação enquanto uma condição não for alcançada, ex: “Enquanto não andar 5 vezes, continue andando”. São 5 peças de repetição.

## 6 Conclusão

Mais do que nunca a robótica está presente na sociedade atual e a tendência é que esta presença continue a crescer cada vez mais e com isso é necessário que as crianças de hoje comecem a aprender a programação e a robótica desde cedo para que no futuro elas possam ser as responsáveis pela continuidade dessa evolução e para isso, é importante que a educação mude, evolua e dê mais atenção para o desenvolvimento do raciocínio lógico, para a resolução de problemas por meio da lógica e para a programação de computadores.

O primeiro passo para essa mudança é dar uma maior atenção a programação nos níveis iniciais da educação básica, começar desde cedo a ensinar as crianças a importância da programação e o realizar o ensino de maneira que não prejudique o aprendizado e a infância, ou seja, de maneira lúdica, mas eficiente. O projeto aqui apresentado propôs uma solução para que esse ensino possa ocorrer de maneira divertida e de baixo custo, além de não precisar necessariamente da escola para ensinar programação e robótica. A importância do aprendizado da programação vai além de apenas aprender uma profissão, a programação estimula o raciocínio lógico, instiga a resolução de problemas por meio da lógica ao invés do achismo.

O levantamento de requisitos foi efetuado normalmente, gerando os requisitos necessários para o esboço inicial da arquitetura. A identificação dos materiais para a construção dos blocos permitiu que a melhor escolha fosse realizada e a identificação da estrutura de hardware para o controle finalizou as etapas do trabalho, possibilitando que o objetivo geral fosse alcançado. A proposta inicial do trabalho em propor um protótipo de uma arquitetura de hardware para ensino de robótica a partir da programação em blocos foi cumprida. Foi identificado também o conjunto total das peças e suas respectivas confecções.

### 6.1 Trabalhos futuros

Como trabalhos futuros destacam-se:

- Implementação da arquitetura de hardware no mundo real.
- Desenvolvimento do software para captura, envio, reconhecimento, conversão e compilação do programa, permitindo a execução da programação e a sua respectiva visualização.
- Implementação do projeto completo e realização das tarefas com crianças de 6 a 12 anos.
- Análise e resultados obtidos com as tarefas aplicadas.

## Referências

- BRASIL, Scratch. **Você conhece o Scratch?**, 2014. Disponível em: <<http://www.scratchbrasil.net.br/index.php/sobre-o-scratch/73-conhece-scratch.html>>. Acesso em 8 de Julho de 2019.
- Code.org – **Sobre nós**. Disponível em: <<https://code.org/international/about>>. Acesso em 08 de Julho de 2019.
- CORDEIRO, Fillipe. **App Inventor: Guia de Criação de Apps**, 2018. Disponível em: <<https://www.androidpro.com.br/blog/desenvolvimento-android/app-inventor/>>. Acesso em 10 de Julho de 2019
- DevMedia. **Arquitetura de Software: Desenvolvimento orientado para arquitetura** 2008. Disponível em: <<https://www.devmedia.com.br/arquitetura-de-software-desenvolvimento-orientado-para-arquitetura/8033>>. Acesso em 26 de Julho de 2019.
- DevMedia. **Fundamentos de Arquitetura de Software**, 2008. Disponível em: <<https://www.devmedia.com.br/fundamentos-de-arquitetura-de-software/10581>>. Acesso em 26 de Julho de 2019.
- EXPOENTE. Disponível em: [http://www.expoente.com.br/educacional/informatica\\_BV.html](http://www.expoente.com.br/educacional/informatica_BV.html)  
Acesso em: 15 set. 2004.
- FERRARI, Dércio Fernando Moraes. **Desenvolvimento cognitivo: As implicações das teorias de Vygotsky e Piaget no processo de ensino e aprendizagem**, 2014
- FIGUEIRA, Anderson Marques da Silva. **Análise das técnicas de levantamento de requisitos para desenvolvimento de software nas empresas de Vitória da Conquista – BA**, 2012.
- FUTUREKIDS. Disponível em: <<http://www.futurekids.com.br/infoeduca.asp?pg=3>> Acesso em: 15 set. 2004.
- KOURI, Márcia Gatti. **Definição de requisitos para um sistema de monitoramento de veículos no transporte rodoviário de cargas**, 2007.
- LEITE, C. A. R.; LEITE, E. C. R.; PRANDI, L. R. - **A aprendizagem na concepção histórico cultural**. Akropolis. Umarama, n. 04, p.203-210, out/dez. 2009.
- LITWIN, Edith. - **Tecnologia educacional – política, histórias e propostas**. Porto Alegre: Artes Médicas, 1997.
- LOPES, Maísa Soares dos Santos. **Ambiente colaborativo para ensino aprendizagem de programação integrando laboratório remoto de robótica**, 2017.
- OLIVEIRA, M. K. de. - **Teorias psicogenéticas em discussão**. 5. Ed. São Paulo: Sumus, 1992.
- PAPERT, Seymour. - **A Máquina das Crianças: repensando a escola na era da informática**. Artes Médicas. Porto Alegre. 1994.
- RIBEIRO, Célia; COUTINHO, Clara; COSTA, Manuel F. - **A Robótica Educativa como Ferramenta Pedagógica na Resolução de Problemas de Matemática no Ensino Básico**. Braga, Portugal. p.440-445, 2011.



SILVA, Akynara Aglaé R. S. **Robótica e educação: Uma possibilidade de inserção sócio-digital**, 2010.

SILVA, Alzira Ferreira da et al. - **Utilização da teoria de Vygotsky em Robótica Educativa**. Natal.

SILVA, Alzira Ferreira da. **RoboEduc: Uma Metodologia de Aprendizado com Robótica Educacional**, 2009

SOMMERVILLE, Ian – **Engenharia de Software**. Tradução: Selma Shin Shimizu Melnikoff, Reginaldo Araki, Edilson de Andrade Barbosa. 8. ed. São Paulo: Person Addison-Wesley, 2007.

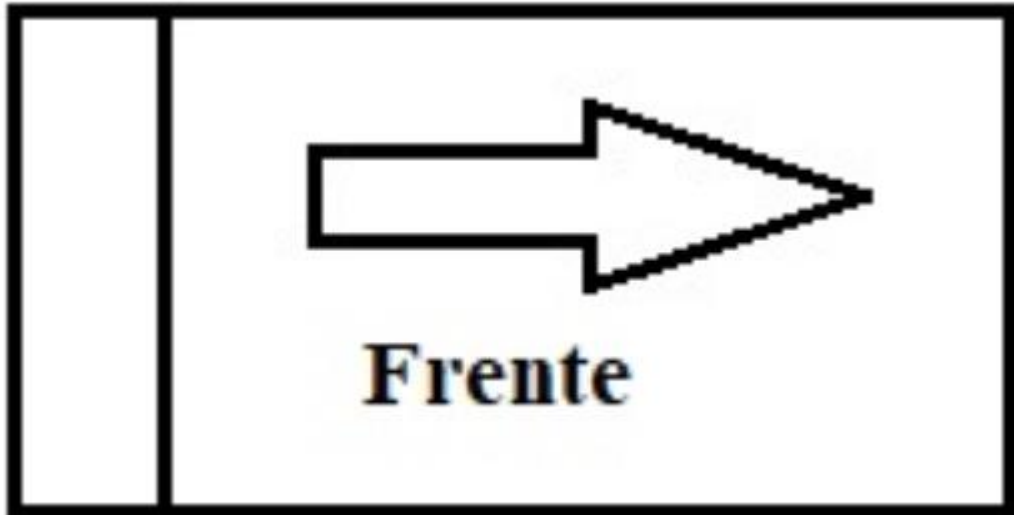
YUS, Rafael. - Comunidades de aprendizagem. **Revista Pátio**. Ano VI. Nº 24. Disponível em:  
<[www.artmed.com.br/pationline/fr\\_conteudo\\_patio.php?codigo=604&secao=334&pai=333](http://www.artmed.com.br/pationline/fr_conteudo_patio.php?codigo=604&secao=334&pai=333)>  
Acesso em: 14 set. 2004.

Zanetti, H.A.P.; Bonacin, R. **Uma Metodologia Baseada em Semiótica para Elaboração e Análise de Práticas de Ensino de Programação com Robótica Pedagógica**. In: Anais do Simpósio Brasileiro de Informática na Educação. p.1233-1242, 2014;

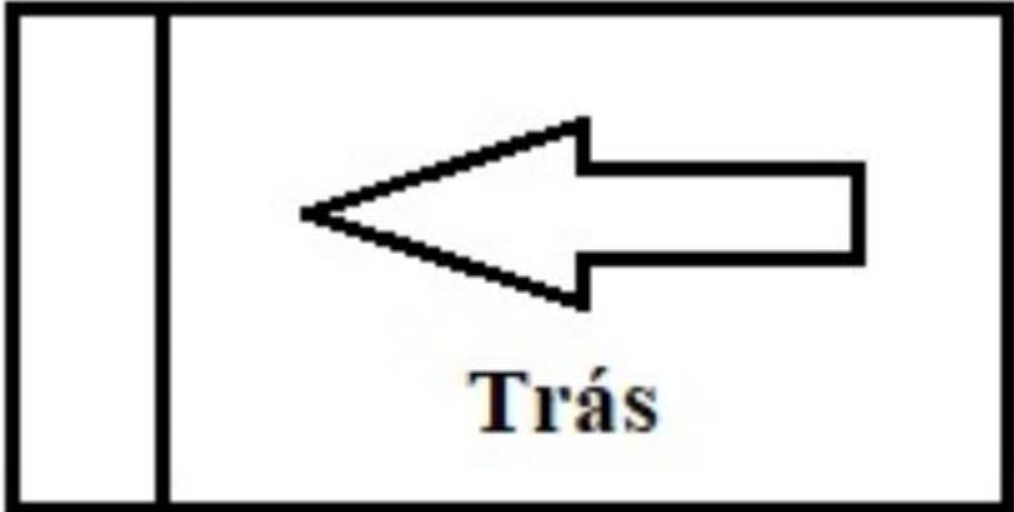
ZILLI, Silvana do Rocio. **A robótica educacional no ensino fundamental: Perspectivas e prática**, 2004.

## Apêndices

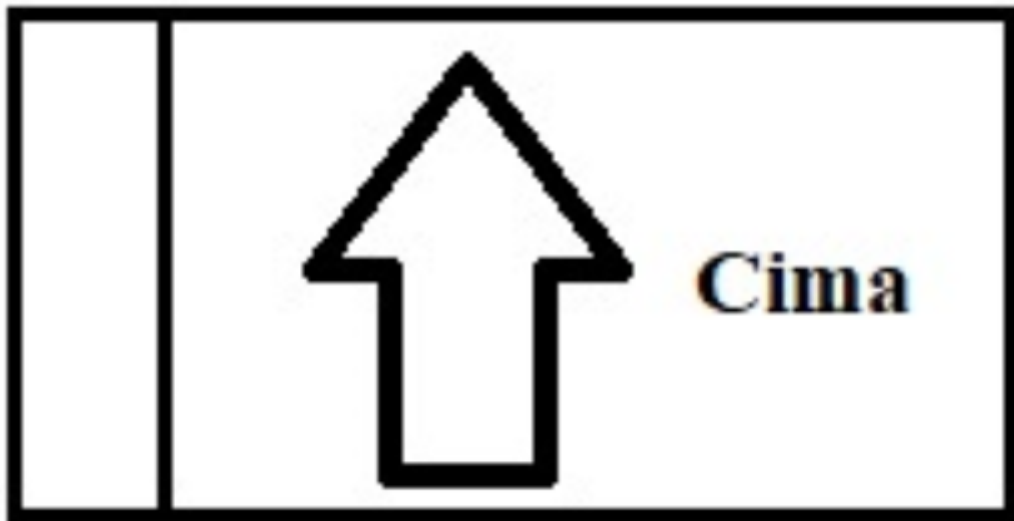
### Apêndice A – Blocos do dispositivo físico.



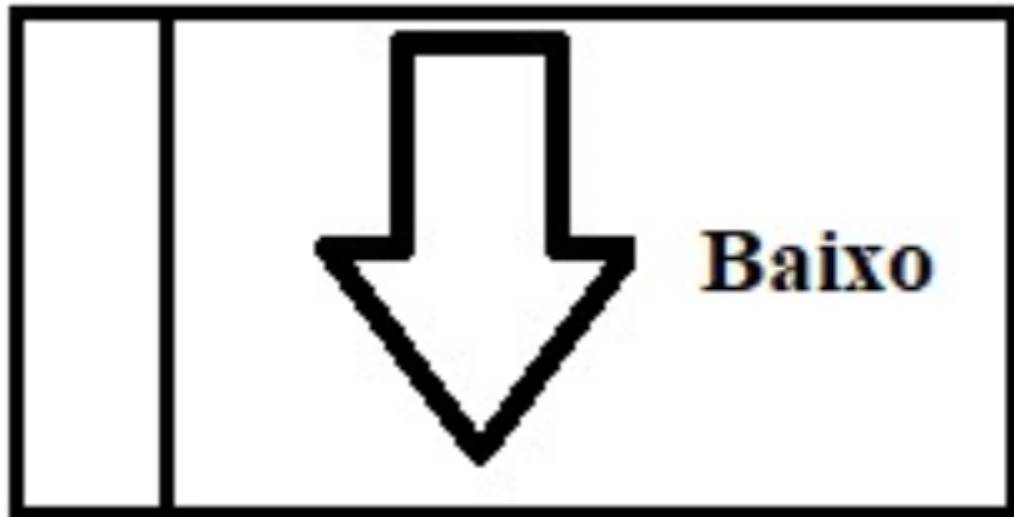
*Figura 20 – Bloco Frente. Fonte: O autor.*



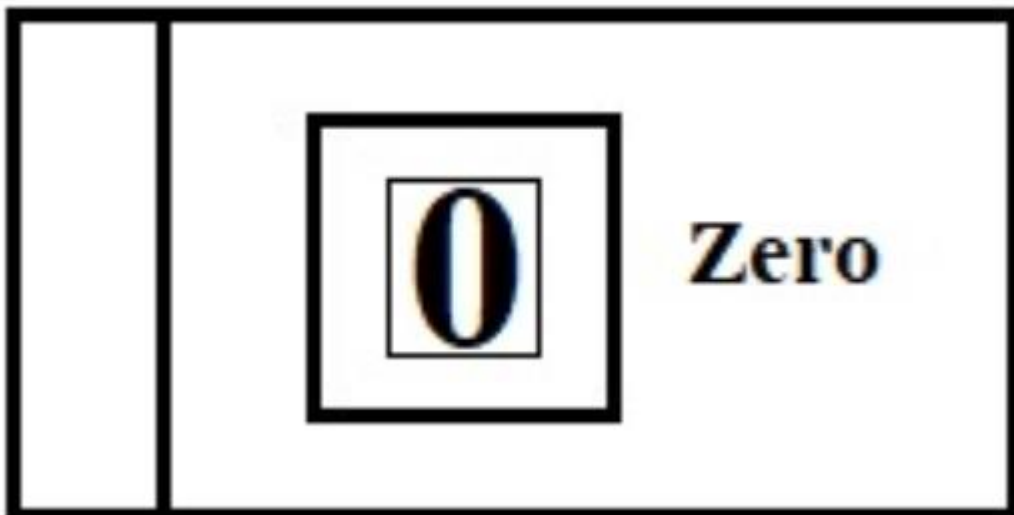
*Figura 21 – Bloco Trás. Fonte: O autor.*



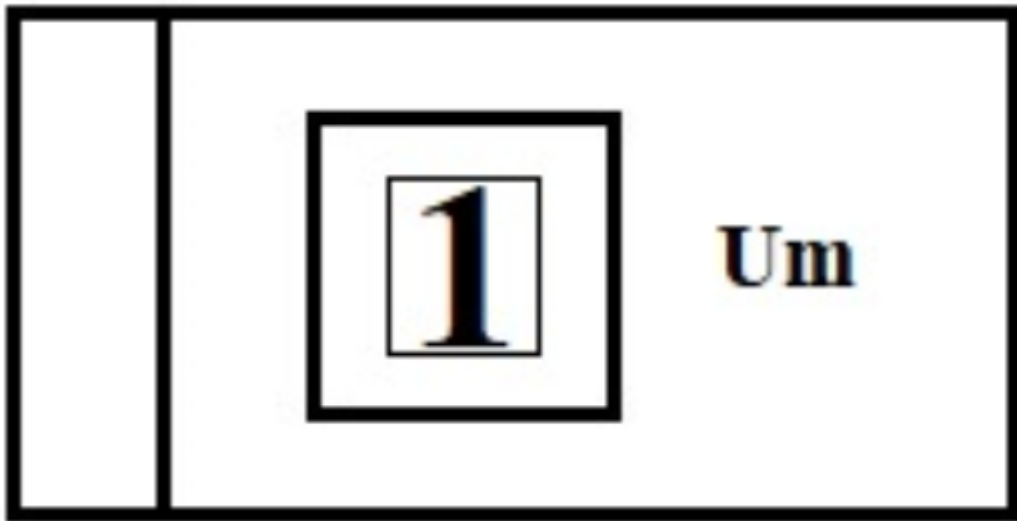
*Figura 22 – Bloco Cima. Fonte: O autor.*



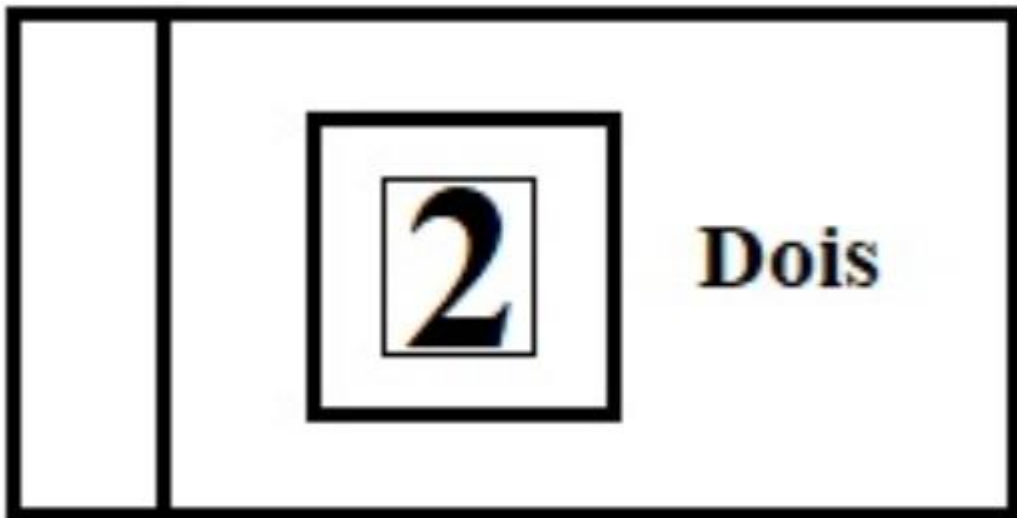
*Figura 23 – Bloco Baixo. Fonte: O autor.*



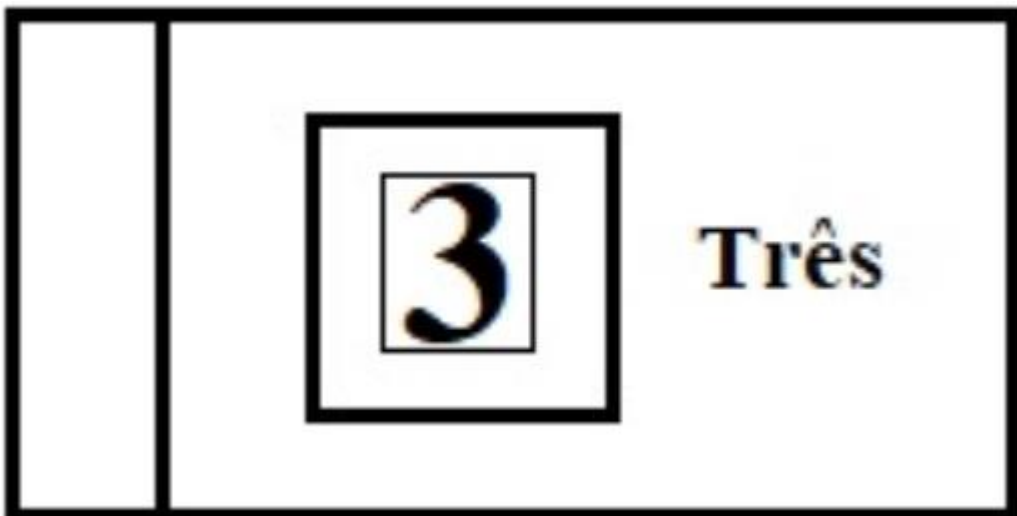
*Figura 24 – Bloco número 0 (Zero). Fonte: O autor.*



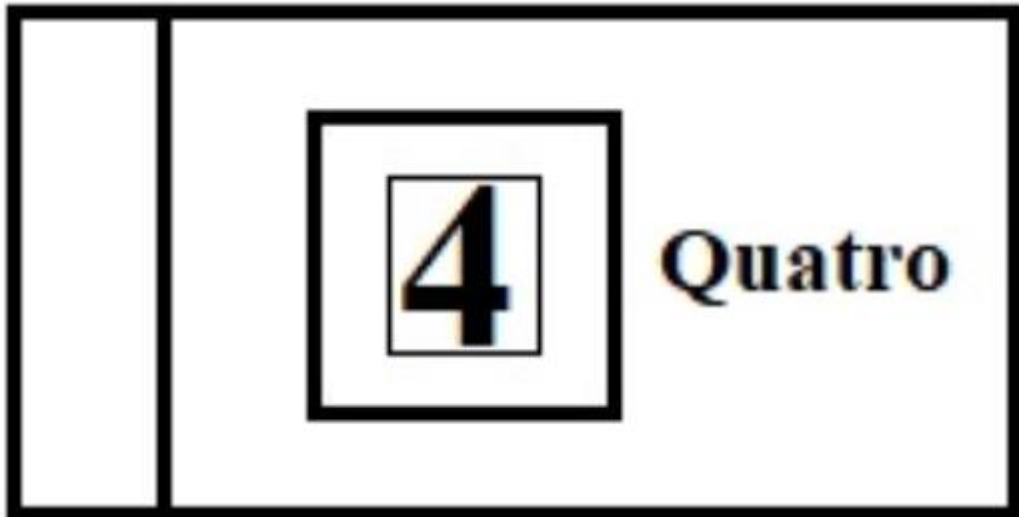
*Figura 25 – Bloco número 1 (Um). Fonte: O autor.*



*Figura 26 – Bloco número 2 (Dois). Fonte: O autor.*



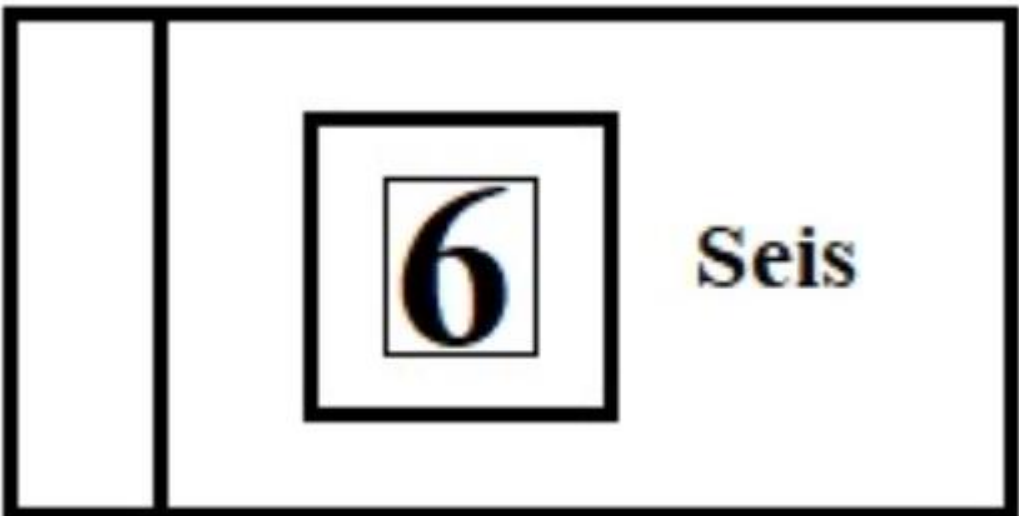
*Figura 27 – Bloco número 3 (Três). Fonte: O autor.*



*Figura 28 – Bloco número 4 (Quatro). Fonte: O autor.*



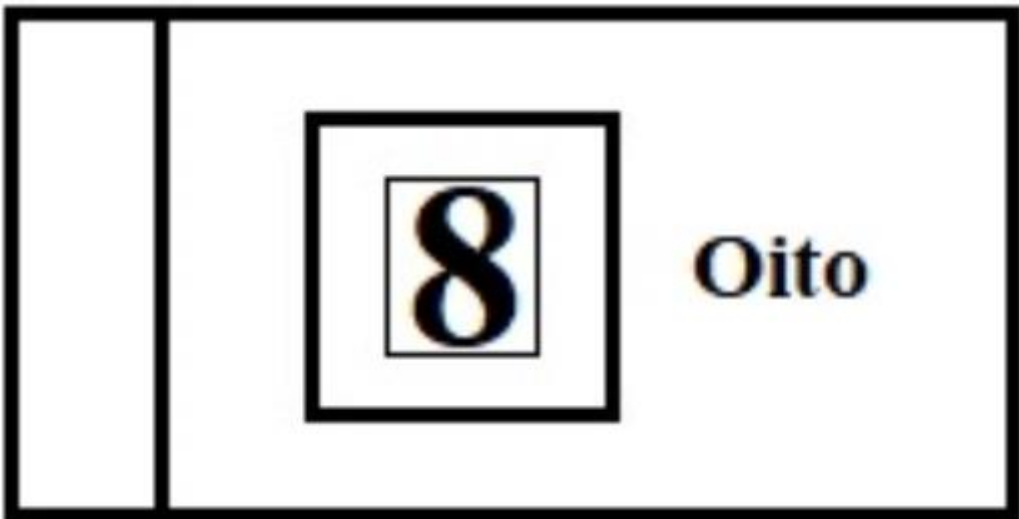
*Figura 29 – Bloco número 5 (Cinco). Fonte: O autor.*



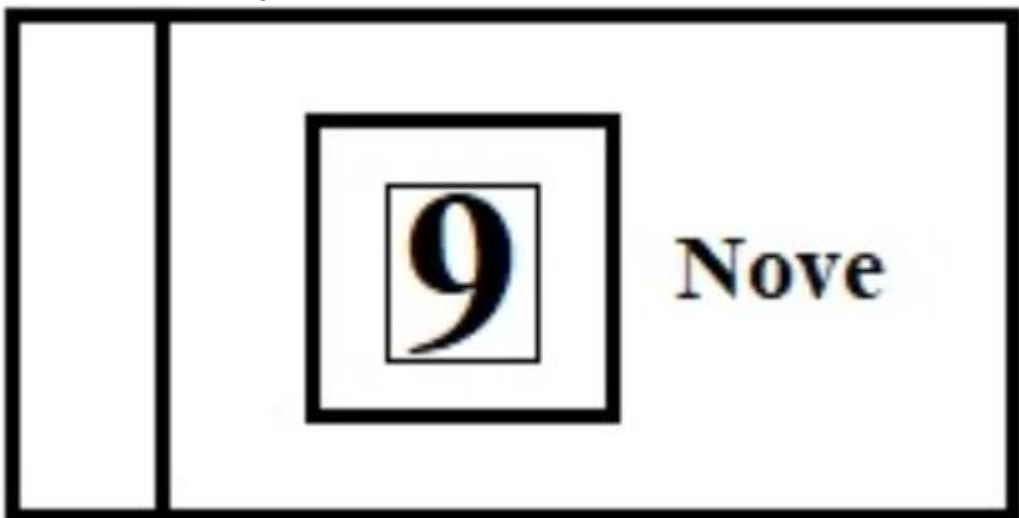
*Figura 30 – Bloco número 6 (Seis). Fonte: O autor.*



*Figura 31 – Bloco número 7 (Sete). Fonte: O autor.*



*Figura 32 – Bloco número 8 (Oito). Fonte: O autor.*



*Figura 33 – Bloco número 9 (Nove). Fonte: O autor.*

	<b>SE</b>
--	-----------

*Figura 34 – Bloco Se. Fonte: O autor.*

	<b>SENÃO</b>
--	--------------

*Figura 35 – Bloco Senão. Fonte: O autor.*

	<b>ENQUANTO</b>
--	-----------------

*Figura 36 – Bloco Enquanto. Fonte: O autor.*



*Figura 37 – Bloco Início. Fonte: O autor.*

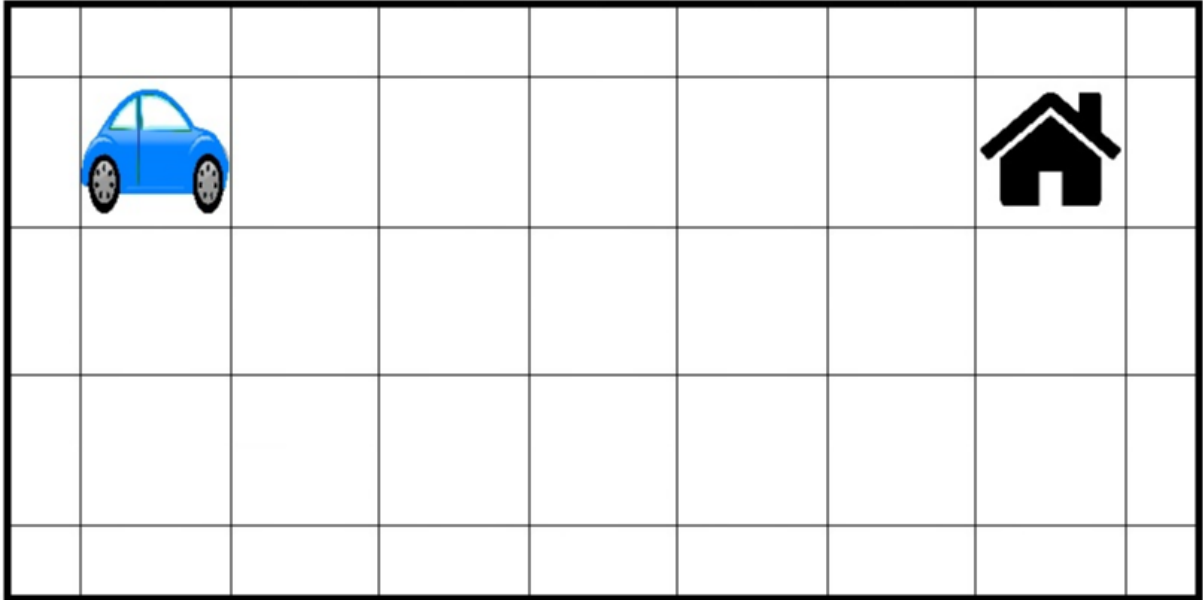


*Figura 38 – Bloco Fim. Fonte: O autor.*



## Apêndice B – Exemplo de exercício com resposta.

1 – Utilizando os blocos, elabore uma programação que faça o carrinho chegar até a casa.



Resposta:

