

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
CAMPUS FRANCISCO BELTRÃO
CURSO DE LICENCIATURA EM INFORMÁTICA

Caroline de Souza Campanha

**Aplicação Web para Programação Visual
usando *Raspberry Pi***

Francisco Beltrão, Paraná

2017

Caroline de Souza Campanha

**Aplicação Web para Programação Visual usando
*Raspberry Pi***

Trabalho de Conclusão de Curso, apresentado a Universidade Tecnológica Federal – Campus Francisco Beltrão, como parte das exigências para a obtenção do título de Licenciada em Informática.

Orientador: Prof. Msc. Wellton Costa de Oliveira

Francisco Beltrão, Paraná

2017

Caroline de Souza Campanha

Aplicação Web para Programação Visual usando *Raspberry Pi*/ Caroline de Souza Campanha. – Francisco Beltrão, Paraná, 2017-

47 p. : il. (alguma cor.) ; 30 cm.

Orientador: Prof. Msc. Wellton Costa de Oliveira

monografia – , 2017.

1. Programação Visual. 2. Robótica. I. Prof. Msc. Wellton Costa de Oliveira. II. Universidade Tecnológica Federal do Paraná. III. Campus Francisco Beltrão. IV. Aplicação Web para Programação Visual usando *Raspberry Pi*

CDU 02:141:005.7

Caroline de Souza Campanha

Aplicação Web para Programação Visual usando *Raspberry Pi*

Trabalho de Conclusão de Curso, apresentado a Universidade Tecnológica Federal – Campus Francisco Beltrão, como parte das exigências para a obtenção do título de Licenciada em Informática.

Trabalho aprovado. Francisco Beltrão, Paraná, 31 de Outubro de 2017.

Prof. Msc. Wellton Costa de Oliveira
UTFPR (Orientador)

Prof. Doutor Eng. Francisco Antonio Fernandes Reinaldo
UTFPR (Presidente da Banca)

Prof^a. Doutora Maici Duarte Leite
UTFPR (Membro Banca)

Folha de Aprovação assinada encontra-se arquivada na Coordenação do Curso.

Dedico este trabalho aos meus pais,
João Pedro e Rita, e a minha irmã Karine
que sempre me apoiaram e acreditaram
em meu potencial. Obrigado! Amo vocês.

AGRADECIMENTOS

Nenhuma batalha é vencida sozinha. No decorrer dessa luta algumas pessoas estiveram junto me acompanhando ao longo desse caminho, me incentivando a buscar as minhas vitórias para que assim eu pudesse conquistar meu sonho.

Agradeço primeiramente à minha família, pois acredito que sem o apoio deles seria muito difícil vencer esse desafio, sou eternamente grata pela admiração, incentivo e carinho de vocês.

Muito Obrigado também ao meu namorado, André Luza, que compartilho comigo esse momento, foi muito paciente em meus momentos de ausência, e esteve o tempo todo ao meu lado, dando apoio moral, revisando o texto comigo e até mesmo auxiliando na montagem do protótipo robótico. Você é essencial para mim.

Agradeço ao meu orientador Prof. Msc. Wellton Costa de Oliveira, pela sabedoria com que me guiou nesta trajetória me dando todo o suporte necessário, não medindo esforços para que meu tão almejado projeto fosse desenvolvido.

Aos meus amigos e colegas da universidade, que torceram por mim. Por termos compartilhados vários momentos de alegria e de ansiedade até a entrega final de nossos trabalhos. Vocês fazem parte dessa história também, e lembrarei de todos com muito carinho.

A Coordenação do Curso, pela cooperação, e demais professores que passaram por essa trajetória compartilhando conhecimentos, incentivando - nos a buscar sempre o melhor para que assim pudessemos chegar vitoriosos ao fim dessa conquista.

Gostaria de deixar registrado também, o meu reconhecimento a dois grandes amigos e colegas de trabalho Alex P. dos Santos e Jean Detoni pelos esclarecimentos aos termos técnicos, críticas positivas, pela paciência de me explicarem, instruírem e darem sugestões. Vocês são demais.

Agradeço a Overleaf pela oportunidade em utilizar $\text{\LaTeX}2\epsilon$ durante orientação e desenvolvimento de meu trabalho.

Enfim, agradeço a todas as pessoas que fizeram parte dessa importante fase de minha vida, peço desculpas àquelas que não se fizeram presentes entre estas palavras, mas podem ter certeza que fazem parte do meu pensamento e de minha eterna gratidão. Meu muito obrigado a todos que de alguma forma contribuíram para a realização e concretização deste trabalho, desse sonho.

*Ensinar não é transferir conhecimento,
mas criar possibilidades para a
sua produção ou a sua construção.
(Paulo Freire)*

RESUMO

No mundo contemporâneo, as tecnologias computacionais, tais como computadores de mesa, *smartphones*, internet, *web*, etc, são vistas como novas transmissoras de informação global. Porém, chegou-se a um ponto que utilizar tais tecnologias somente para transmissão de informação não se faz mais suficiente. Para modificar esses conceitos já pré-estabelecidos temos que rever e aprimorar os conceitos pedagógicos de ensino aprendizagem com relação ao uso das tecnologias no âmbito educacional. Nossa sociedade vem de um tradicional sistema de ensino conservador, onde fazer a inserção de novos métodos integrando as novas tecnologias é um grande desafio para educadores. A temática da utilização da informática na educação vem sendo estudada por especialistas que buscam explicar a importância do uso das tecnologias aliadas aos recursos pedagógicos e como eles podem auxiliar no desenvolvimento do conhecimento dos estudantes. Uma das formas de utilizar as tecnologias para auxiliar no processo ensino-aprendizagem é através do ensino de lógica de programação visual que pode ser bem aplicada nesse contexto, pois através de um problema proposto as crianças podem desenvolver possíveis soluções. Descrevendo, ordenando, criando sequências lógicas metaforicamente através de ilustrações, visualizando a sua resolução, esses ambientes são mais propícios a motivar e despertar o interesse das crianças. Um sistema *web*, que possibilite introduzir o conceito de lógica de programação visual através de uma interface atrativa com funções e um protótipo robótico físico construído usando *Raspberry PI* e outros componentes, que seja capaz de interpretar e reproduzir os movimentos programados. Pode introduzir o conhecimento mesmo que de forma lacônica, visando a construção sequencial de instruções finitas para solucionar determinados problemas. A união da robótica com a lógica de programação visual pode ser um elemento facilitador no processo ensino-aprendizagem.

Palavras-chave:Ensino. Lógica de Programação. Linguagem programação Visual. Robótica. *Raspberry Pi*.

ABSTRACT

In the contemporary world, computer technologies, such as desktop computers, smartphones, the internet, the web, etc., are seen as new transmitters of global information. However it arrived - if in a point that use the technologies just as a form as knowledge transmitters is not enough anymore. And to modify those preconceived notions we have to review and improve the pedagogical concepts of teaching-learning in relation to the use of the technologies in the educational field. Our society comes from a traditional system of conservative teaching in which to insert new methods, and integrating the new technologies, is a great challenge for the educators. The thematic in use the computing in the education is being studied by specialists that research to explain the importance of the use of the technologies along the pedagogical resources, and how they can help in the development of the children's knowledge. One of the ways in use the technology to help in the process of the teaching-learning is through the logical teaching of the programming. The visual language of programming can be well applied in this context, because through a problem proposed to the children, possible solutions can be developed describing, ordering, creating logical sequences metaphorically through illustrations, viewing the resolution, these environments are better to motivate and wake up the children's interest. A web system, that make it possible to introduce the logical concept of programming in a visual form through of an attractive interface with functions and a robotic physical prototype, built using Raspeberry PI and other components, that will be able to interpret and reproduce the programmed moves. It can introduce the knowledge, but at the same time laconic, viewing the sequential construction of infinite instructions to solve determinate problems. The union of the robotic with the logic of visual programming, can be a facilitator element in the teaching-learning process.

Keywords: Education. Logic programming. Visual language programming. Robotics. *Raspberry Pi.*

LISTA DE ILUSTRAÇÕES

Figura 1 – Diagrama <i>Raspberry Pi</i> Model B+	21
Figura 2 – Diagrama GPIO <i>Raspberry Pi</i> Model B+	22
Figura 3 – Visão Geral do Projeto	28
Figura 4 – Kit chassi para robótica	29
Figura 5 – Diagrama Ponte H L298N	30
Figura 6 – Esquema elétrico	31
Figura 7 – Função em Python	32
Figura 8 – Código em PHP para interpretar os comandos provenientes do <i>JavaScript</i> e sua consequente chamada do Python	34
Figura 9 – Protótipo do <i>Front-End</i> da aplicação web	35
Figura 10 – Código fonte do <i>Front-end</i> da aplicação web desenvolvida (Parte1)	35
Figura 11 – Código fonte do <i>Front-end</i> da aplicação web desenvolvida (Parte 2)	36
Figura 12 – Protótipo robótico final	39
Figura 13 – Interface versão desktop	40
Figura 14 – Interface versão mobile	40
Figura 15 – Botão Frente	41
Figura 16 – Botões Para Frente, Direita, Esquerda e Voltar	41
Figura 17 – (A) Seta Seleccionada; (B) Seta removida	42
Figura 18 – Botão Recomeçar	42
Figura 19 – Projeto final	43

SUMÁRIO

1	INTRODUÇÃO	12
1.1	CONSIDERAÇÕES INICIAIS	12
1.2	OBJETIVO GERAL	14
1.3	OBJETIVOS ESPECÍFICOS	14
1.4	JUSTIFICATIVA	14
1.5	ESTRUTURA DO TRABALHO	15
2	REFERENCIAL TEÓRICO	16
2.1	O DESENVOLVIMENTO DO CONHECIMENTO	16
2.2	LINGUAGEM DE PROGRAMAÇÃO VISUAL	17
2.3	ROBÓTICA PEDAGÓGICA	18
2.4	ROBÓTICA COM <i>RASPBERRY PI</i>	19
2.4.1	<i>O Raspberry Pi</i>	20
2.4.2	GPIO do <i>Raspberry Pi</i>	21
2.5	<i>PYTHON</i> E APLICAÇÕES WEB	22
2.5.1	Linguagem de programação Python	23
2.5.2	Aplicações Web	23
2.5.2.1	HTML	23
2.5.2.2	CSS	24
2.5.2.3	Javascript	24
2.5.3	Frameworks	25
2.6	TRABALHOS RELACIONADOS	26
3	METODOLOGIA	28
3.1	VISÃO GERAL	28
3.2	<i>HARDWARE</i> - PARTE FÍSICA	29
3.2.1	Parte Mecânica	29
3.2.2	Esquema Elétrico e Eletrônico	30
3.3	<i>SOFTWARE</i> - PARTE LÓGICA	31
3.3.1	Python e o GPIO	31
3.3.2	Servidor Web e Liguagem de Programação (Back-end)	33
3.3.3	Interface Gráfica Web (Front-end)	35
4	RESULTADOS E DISCUSSÃO	39
4.1	UTILIZAÇÃO DO SISTEMA	39
4.2	DEMONSTRAÇÃO DO SISTEMA	40

4.3	DISCUSSÃO	43
5	CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS	44
5.1	TRABALHOS FUTUROS	45
	REFERÊNCIAS	46

1 INTRODUÇÃO

Esta seção inicia-se com uma breve explanação do tema principal: Programação Visual. Após isso delimita-se os objetivos do trabalho e a justificativa pela qual este trabalho foi desenvolvido. Na última seção, a estrutura completa do trabalho apresentado.

1.1 CONSIDERAÇÕES INICIAIS

Nossa sociedade vem de um tradicional sistema de ensino conservador, onde fazer a inserção de novos métodos integrando as novas tecnologias é um grande desafio para educadores. A temática da utilização da informática na educação vem sendo estudada por especialistas que buscam explicar a importância do uso das tecnologias aliadas aos recursos pedagógicos e como eles podem auxiliar no desenvolvimento do conhecimento das crianças.

Muitas pessoas, inclusive educadores, acabam vendo as tecnologias, em específico os computadores, como transmissores de informação. Porém chegou - se em um ponto que utilizar as tecnologias somente como forma de transmissão de informação não é mais suficiente. E para modificar esses conceitos já pré estabelecidos tem-se de rever e aprimorar os conceitos pedagógicos de ensino-aprendizagem.

ALTOÉ e FUGIMOTO (2009) salientavam que:

“A inserção das tecnologias na educação coloca em xeque o modelo de educação fundamentado em teorias de ensino e de aprendizagem tradicional. Entretanto, para que ocorra uma ruptura significativa com esse modelo ultrapassado é necessário que os professores assumam uma postura reflexiva em relação à sua formação.”

Um método de utilização do computador como recurso para desenvolvimento do conhecimento pode ser o ato de programar. Essa habilidade que até então se restringia a profissionais da tecnologia ganha espaço e incentivo para atingir pessoas de todos os níveis intelectuais.

Uma forma de incentivar as crianças a aprenderem lógica de programação é proporcionando atividades com materiais e recursos intuitivos, em ambientes que não saiam de seu convívio habitual. Linguagem de Programação Visual podem ser bem aplicadas neste contexto, pois através de um problema proposto os estudantes podem desenvolver possíveis soluções descrevendo, ordenando, criando sequências lógicas metaforicamente através de ilustrações, visualizando a sua resolução. Esses ambientes são mais propícios a motivar e despertar o interesse dos alunos.

Para a mestre em Psicologia de Crianças e Adolescentes e fundadora do *Playground* da Inovação, Fernanda Furia ¹, a programação é um dos passos para a alfabetização digital e será cada vez mais uma habilidade para toda a vida. pois desenvolve capacidades como resolução de problemas, raciocínio lógico, relação entre causa e consequência, raciocínio matemático, linguística, etc..(FURIA, 2014)

Programas e aplicativos que incentivam a programação visual vêm se popularizando como o *Scratch*² e o *CODE*³, apresentando métodos de ensino orientado ao *design* para interface. A união da programação com a robótica está em alguns brinquedos programáveis como os da linha *Lego Mindstorms*®⁴, que possui uma interface dinâmica e atrativa, porém o alto custo para aquisição de tais *kits* torna-o restrito para pessoas com baixo poder aquisitivo. Por outro lado, computadores de placa única *single boards*/, tais como *Arduino*⁵ ou *Raspberry Pi* ⁶ ganham espaço nos quesitos lógica de programação e robótica, tornando mais simples a construção de modelos robóticos, mas geralmente não apresentam uma interface com características chamativas. A linguagem de programação nativa utilizada por esses modelos se torna complexas para pessoas com baixa idade. As crianças podem obter mais benefícios ao aprender Lógica de Programação.

“Aprender a escrever programas estende sua mente, e ajuda a pensar melhor, cria uma maneira de pensar sobre coisas que eu acho que são úteis em todas as áreas” (Bill Gates)(JUNIOR, 2015)

Nesta temática, Seymour Papert desenvolveu estudos sobre uma filosofia educacional denominada construcionismo, com base nas teoria construtivista de Jean Piaget buscou demonstrar que o computador pode servir para o desenvolvimento intelectual da criança além de apresentar uma nova concepção de ensino.

Dando seguimento a contextualização do projeto, a corrente pesquisa apresenta um ambiente capaz de traduzir programas, representados por setas de direção, em reações no modelo robótico físico. Para isso, o ambiente provê uma interface em que elementos (setas de direção), quando ordenados possibilitam a construção de pequenos programas. Cada programa, ao ser executado, envia instruções para componentes eletrônicos conectados a um minicomputador *Raspberry Pi*, de maneira que o aluno observe o resultado do experimento por meio do comportamento apresentado pelo modelo robótico.

¹ Os estudos e abordagens de Fernanda Furia, tornaram – se um Projeto de Lei Nº 933, de 2016 do Estado de São Paulo e foi apresentado a Assembleia Legislativa do Estado. Tendo como Ementa Tornar obrigatório o oferecimento opcional do ensino da "Linguagem de Programação de Códigos" pelas escolas públicas e privadas do Estado. O projeto foi aprovado com parecer favorável do Deputado e seus favoráveis.

² <<https://scratch.mit.edu/>>

³ <<https://code.org/>>

⁴ <<https://www.lego.com/en-us/mindstorms>>

⁵ <<https://www.arduino.cc/>>

⁶ <<https://www.raspberrypi.org/>>

1.2 OBJETIVO GERAL

Desenvolver uma aplicação *web* para programação visual, que possibilite a criação de pequenos programas para controle básico de movimento de um robô.

1.3 OBJETIVOS ESPECÍFICOS

Pesquisar e estudar sobre de que forma o ensino de programação para crianças pode influenciar no seu desenvolvimento, e quais benefícios podem agregar ao processo de ensino aprendizagem.

Fazer o levantamento de requisitos de melhores soluções para a prototipação do projeto visando que o público alvo serão crianças, definindo assim a arquitetura para o desenvolvimento do projeto.

Montar um veículo robótico que seja capaz de interpretar as funções de movimento informadas pela página *web* desenvolvido e hospedada no *Raspberry Pi*.

1.4 JUSTIFICATIVA

Acredita-se que ensinar lógica de programação para crianças desperta o interesse pelas áreas da ciência exatas e tecnologia, o que pode contribuir para a escolha de sua futura formação acadêmica e futura profissão. Esse assunto vem sendo estudado em projetos de pesquisas que incentivam o ensino de lógica de programação na área da Informática na Educação.

Papert em seu livro LOGO : Computadores e Educação fez a seguinte indagação:

Se programar for vista sob uma perspectiva adequada, não há de muito surpreendente no fato de que isso aconteça. Programar significa, nada mais, nada menos, que comunicar-se com o computador numa linguagem que tanto ele quanto o homem podem "entender". E aprender línguas é uma das coisas que as crianças fazem bem. Toda criança normal aprende a falar. Por que então não deveria aprender a "falar" com um computador?(PAPERT, 1985)

No Brasil, a Sociedade Brasileira de Computação (SBC), através da sua Comissão de Especialistas de Ensino de Computação e Informática e do GT3 (Grupo de Trabalho de Licenciatura em Computação), tem fomentado as discussões sobre a importância de introduzir o pensamento computacional e algorítmico nas escolas de ensino médio, para que se possa desenvolver competências e fomentar o interesse pela área, aumentando o número de profissionais no país (JÚNIOR et al., 2005).

Um protótipo de sistema *web*, que possibilite introduzir o conceito de lógica de programação de forma visual através de aplicações com funções e um robô capaz de

interpretar e reproduzir os movimentos programados. Pode introduzir o conhecimento mesmo que de forma lacônica, visando à construção sequencial de instruções finitas para solucionar determinados problemas. A união da robótica com a lógica de programação visual pode ser um elemento facilitador no processo ensino-aprendizagem.

1.5 ESTRUTURA DO TRABALHO

Neste capítulo, foi apresentada a introdução ao trabalho, com a contextualização, seus objetivos, justificativa e organização;

No capítulo 2, será abordada a conceituação do Desenvolvimento do Conhecimento e da Programação Visual aplicada a crianças e referencial teórico de estudos relacionados ao tema principal

O capítulo 3 aborda a metodologia do trabalho, apresentando quais foram os materiais e métodos utilizados e detalhando cada etapa do projeto;

A apresentação dos resultados e a discussão do trabalho serão abordadas no capítulo 4;

E no capítulo 5, a conclusão do trabalho e sugestões para trabalhos futuros.

2 REFERENCIAL TEÓRICO

O presente capítulo apresenta os trabalhos correlatos e a fundamentação teórica envolvida na concepção do projeto.

2.1 O DESENVOLVIMENTO DO CONHECIMENTO

Ao observar o contexto do modelo de ensino da educação tradicional, verifica-se que o professor era visto como o transmissor do conhecimento, o ato de aprender era sinônimo de memorização do conteúdo. Assim o computador na educação a algum tempo atrás também repassava essa visão, de que era uma máquina de ensinar e ou transmissora do conhecimento.

Em uma sociedade denominada atualmente como “sociedade do conhecimento”, novas demandas sobre o uso da informática na educação vem sendo pesquisadas e estudadas, possibilitando novos conceitos para o uso do computador.

A teoria epistemológica desenvolvida por Jean Piaget procurava explicar o desenvolvimento do conhecimento, onde defendia que há uma relação evolutiva entre o sujeito e o meio onde ele vive, ou seja, criança reconstrói suas ações e ideias quando se relaciona com novas experiências em diferentes ambientes. Para Piaget em sua perspectiva construtivista, o começo do conhecimento é a ação do sujeito sobre o objeto, ou seja, o conhecimento humano se constrói na interação homem-meio, sujeito-objeto. Uma ação real resulta em conhecimento quando a ação consiste em transformações a fim de se obter o entendimento de algo, em resumo Piaget acreditava que a aprendizagem ocorria baseando - se na experiência vivida no mundo real (ARGENTO, 2009).

A teoria do construcionismo, filosofia educacional idealizado por Seymour Papert baseou-se na teoria epistemológica desenvolvida por Jean Piaget. Que propunha o uso do computadores como uma ferramenta indispensável para a aprendizagem relacionando teoria e prática de informática aos princípios piagetianos de aprendizagem. Dessa forma têm-se a teoria que o computador permite à criança construir por ela mesma, suas estruturas intelectuais.

Papert (1985) preconizava que educar consiste em criar situações para que os aprendizes se engajem em atividades que alimentam este processo construtivo. Educar, portanto, é principalmente dar condições para que os alunos construam, mas não se resume a isso. Segundo Maltempo (2004) o Construcionismo postula que o aprendizado ocorre especialmente quando o aprendiz está engajado em construir um produto de significado pessoal (por exemplo, um poema, uma maquete ou um website), que possa ser mostrado a

outras pessoas. Portanto, ao conceito de que se aprende melhor fazendo, o Construcionismo acrescenta: aprende-se melhor ainda quando se gosta, pensa e conversa sobre o que se faz.

É nessa temática de construir o conhecimento que enfatiza-se a importância e relevância de ensinar lógica de programação e robótica para crianças, não sintaxes complexas de linguagens, e protótipos revolucionários de robôs, mas pequenos programas em que elas possam criar algo ou modificar. Não somente os computadores mas todas as tecnologias imensuráveis que surgem dia após dia, se ensinadas e incentivadas a serem usadas podem ser aliadas no desenvolvimento da construção do conhecimento.

2.2 LINGUAGEM DE PROGRAMAÇÃO VISUAL

Aprender a programar pode ser muito difícil para pessoa de todas as idades. Além dos desafios da aprendizagem para formar soluções estruturadas para os problemas necessita-se da compreensão de uma sintaxe de comandos que talvez possam ser um pouco confusas. Esses desafios apresentam dificuldades que muitas vezes podem ser desanimadoras para programadores iniciantes.

Segundo Kelleher e Pausch (2005), desde o início de 1960 os pesquisadores construíram uma série de linguagens e ambientes de programação com a intenção de fazer a programação acessível a um maior número de pessoas. Também afirmam que é preciso simplificar a sintaxe das linguagens, para que estas fiquem mais próximas da linguagem natural.

Os autores definem a programação como “o ato de reunir um conjunto de símbolos que representam ações computacionais”. Usando esses símbolos, os usuários podem expressar as suas intenções para o computador e, dado um conjunto de símbolos, um usuário os compreende pode prever o comportamento do computador.

O ensino de programação com ênfase na visualização permite que o aluno mantenha o foco no raciocínio lógico, e não se preocupe inicialmente com a sintaxe das linguagens, como ocorre ao utilizar as linguagens de programação estruturadas.

Sudol (2009) afirma que as linguagens de programação visuais são destinadas a fornecer metáforas para programadores. Estas metáforas são frequentemente relacionadas às atividades do mundo real. As metáforas proporcionam o contato com a abstração de conceitos e o decorrer de seu uso possibilita ao usuário torna-se capaz de propor soluções aos problemas mais facilmente e compreender conceitos de programação.

O autor ressalta em sua dissertação que as VLPs (*visual language programming*) ajudam a criar um ambiente de programação mais natural e familiar para os usuários, e apresentam como competências:

- O uso de metáforas que são de fácil compressão;
- Reduzir a carga cognitiva sobre os estudantes que aprendem sua primeira língua;
- Criar um código que é facilmente lido e compreendido;
- É utilizável por professores;
- Pode ser integrada com o currículo atual de ensino;
- Apoio para os estudantes na transição para linguagens de programação estruturadas;
- Faz com que programar se torne divertido.

Os elementos visuais utilizados nas VLPs são muitas vezes bastante básicos, principalmente blocos que podem ser usados na condição ou no lado da instrução para serem combinados e construir uma sequência finita e lógica. O aspecto visual simples ajuda a descobrir o que pode ser feito e a reconhecer as regras existentes para a construção do programa.

A programação visual tende a ser mais acessível principalmente por simplificar a mecânica da programação e oferecer apoio aos aprendentes, por proporcionar aos alunos motivação para aprender a programar. Os iniciantes precisam sentir que eles podem programar. No entanto, a dificuldade não é o único motivo pela qual as pessoas hesitam em aprender. Existem vários fatores sociológicos como o fato de alunos que não verem relevância em aprender programação ou definem que as áreas da ciência da computação são campos de atuação restrita a determinados perfis, aos quais eles auto se julgam não se encaixar.

Tentar quebrar algumas dessas barreiras sociológicas em relação a programação, que vise apoiar os alunos fornecendo motivos interessantes para eles desenvolverem gosto pela programação, criando ambiente favorável é um dos objetivos das linguagens de programação visual. Que potencialmente pode atrair grupos mais diversificado de pessoas para as áreas da ciência da computação.

2.3 ROBÓTICA PEDAGÓGICA

A união entre a robótica e educação é uma realidade em vários ambientes em nossa sociedade, porém, uma realidade que não é para todos, mas tem tudo para dar certo. Primeiro, o robô, como elemento tecnológico, possui uma série de conceitos científicos cujos princípios básicos são abordados pela escola. Segundo, pelo fato de que os robôs mexem com o imaginário infantil, criando novas formas de interação, e exigindo uma nova maneira de lidar com símbolos (SILVA; GUERREIRO; GONÇALVES, 2009).

SILVA, GUERREIRO e Gonçalves (2009) define Robótica Pedagógica ou Robótica Educacional como sendo um ambiente de aprendizagem em que o professor ensina ao aluno a montagem, automação e controle de dispositivos mecânicos que podem ser controlados pelo computador. Para Papert (1994) um ambiente de Robótica Pedagógica deve fazer emergir, no contexto escolar de aprendizagem, ideias tecnológicas para permitir que as crianças se apropriem delas.

Conforme Rodrigues (2014), a robótica pode proporcionar ao aluno a oportunidade de vivenciar experiências semelhantes às que ele tem na vida real. Os robôs podem ser aplicados em várias matérias.

De acordo Zilli et al. (2004), a robótica educacional é uma ferramenta que permite ao professor demonstrar na prática muitos dos conceitos teóricos, às vezes de difícil compreensão, motivando o aluno, que a todo momento é desafiado a observar, abstrair e inventar. Na robótica pedagógica o aluno adquire o conhecimento através de suas próprias observações, esse aprendizado com o esforço tem muito mais significado para o jovem.

Segundo SILVA, GUERREIRO e Gonçalves (2009), a robótica pedagógica envolve um processo de motivação, colaboração, construção e reconstrução. Para isso, faz-se necessário a utilização de conceitos de diversas disciplinas para a construção de modelos, levando os alunos a uma rica vivência interdisciplinar. Ainda segundo o autor, o robô como ferramenta de trabalho possibilita a criação de novas formas de interação com o mundo. A aprendizagem é fundamentalmente uma experiência social, de interação pela linguagem e pela ação. Essa interação deve favorecer a cooperação e autonomia, assegurar a centralidade do indivíduo na construção do conhecimento e possibilitar resultados de ordem cognitiva, afetiva e de ação.

Além de propiciar ao instrutor o conhecimento da tecnologia atual, os usuários dessa metodologia desenvolvem competências como raciocínio lógico, habilidades manuais e estéticas, relações interpessoais, utilização de conceito aprendidos em diversas áreas do conhecimento para o desenvolvimento de projetos, investigação e compreensão, representação e comunicação, trabalho com pesquisa, resolução de problemas por meio de erros e acertos, aplicação das teorias formuladas a atividades concretas, utilização da criatividade em diferentes situações e capacidade crítica (ZILLI, 2002).

2.4 ROBÓTICA COM *RASPBERRY PI*

Esta seção tem como objeto fazer um levantamento sobre o que é o minicomputador *Raspberry Pi*, e partes eletrônicas e mecânicas que são comuns atualmente e que foram utilizadas para a construção de um protótipo robótico.

2.4.1 O Raspberry Pi

O *Raspberry Pi* é um microcomputador desenvolvido pela *Raspberry Pi Foundation* no Reino Unido. O propósito do seu desenvolvimento é reprodução de um computador simples e de baixo custo para que ele pudesse chegar às escolas. Assim, crianças teriam acesso a fundamentos de programação e teriam também um entendimento mais aprofundado sobre a natureza dos computadores e o seu funcionamento (RICHARDSON; WALLACE, 2013).

O microcomputador, é baseado em SOC (*System on Chip*), onde todos os componentes do computador dispostos em um circuito integrado, uma placa única, do tamanho de um cartão de crédito. Que integra um processador do tipo ARM a 700MHz e uma unidade de processamento gráfico. Por não apresentar incluso um disco rígido com memória não volátil, possui ainda por uma entrada de cartão SD para armazenamento de dados. Faz-se necessário um cartão de memória SD com no mínimo com 4GB para instalar o sistema e as demais aplicações.

Estão disponíveis hoje três modelos A, B e B+ a diferença entre os modelos está relacionada a capacidade de memória RAM e periféricos disponíveis. Ele pode ser utilizado como computador e como um microcontrolador, pela sua capacidade de integração com a eletrônica, por isso os componentes são dispostos de forma visíveis, para que o usuário possa observar a arquitetura deste pequeno sistema e desenvolver integração com outros dispositivos como o *Arduino*.

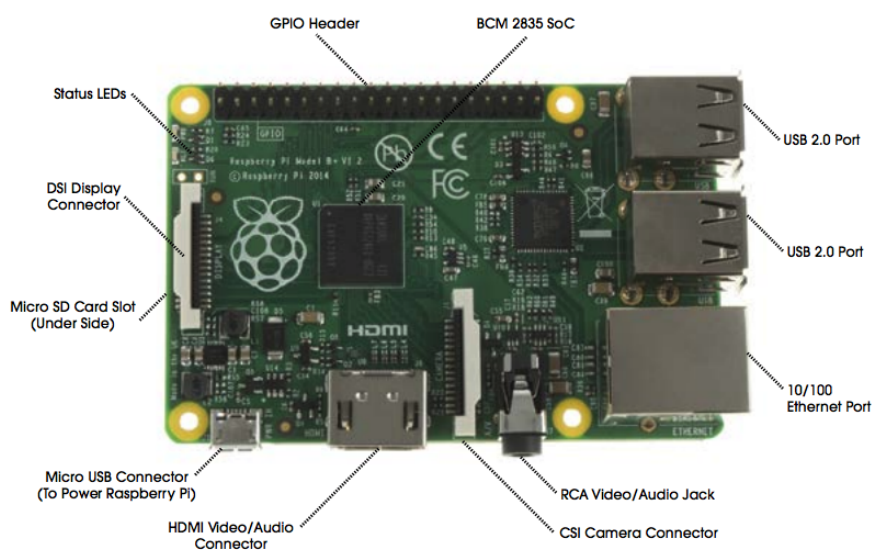
O Seu sistema operacional oficial é o *Raspbian* uma distribuição *Linux* baseada em *Debian*, porém, existem sistemas operacionais de terceiros que também são compatíveis com o microcomputador, como o *Ubuntu Mate*, *Snappy Ubuntu Core*, *Windows 10 IOT Core*, *Pinet*, *Risc OS* entre outros.

O *Raspbian*¹, apresenta disposto em seu menu iniciar os tipos de aplicação ou ferramenta (acessórios, educação, Internet, programação, ferramentas do sistema, preferências, executar e logout). Engloba o navegador Midori, além de diversas ferramentas de programação como o *Python*, o *Scratch* e o *Squeak* que permitem o ensino de algoritmos através de métodos práticos e instigantes.

No projeto utilizou-se o *Raspberry Pi 3 model B+* apresentado na Figura 1, por conter *wi-fi* integrado, assim não foi preciso adquirir um adaptador, e por conter um GPIO estendido de 40 pinos.

Cardoso, Costa e Sousa (2013) destacam que esse minicomputador é um equipamento bastante interessante e com elevado potencial, principalmente, no ensino e no desenvolvimento da arquitetura dos computadores, das linguagens de programação e até mesmo em nível do entretenimento.

¹ <<https://www.raspberrypi.org/downloads/>>

Figura 1 – Diagrama *Raspberry Pi* Model B+

Fonte: <https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>

Richardson e Wallace (2013) descreve que o surgimento do *Raspberry Pi* se deu sem fins lucrativos, com a finalidade de proporcionar a crianças e jovens, sobre tudo aqueles que não possuíam condições para aquisição de computador, um primeiro contato com dispositivos que proporcionam e instigaram o interesse em aprender programação.

Nesse novo cenário interativo da educação, o *Raspberry Pi* revela-se um dispositivo com características inovadoras, que apresenta uma contribuição valiosa no ensino aprendizagem de programação, “poderá resolver grande parte dos problemas ancorados à programação, bem como motivar cada vez mais as crianças e jovens a gostarem de informática”(CARDOSO; COSTA; SOUSA, 2013).

2.4.2 GPIO do *Raspberry Pi*

As portas do GPIO são a comunicação entre o *Raspberry Pi* e o mundo exterior, elas permitem a criação de protótipos físicos. Basicamente é constituída por pinos, no caso do *Raspberry Pi* B+ utilizado apresentado na Figura 1, são especificamente 40 pinos, mas há modelos que possuem 26 pinos como é o caso do *Raspberry Pi* B.

Abaixo a figura 2 apresenta um guia para identificação dos pinos de um *Raspberry Pi* B+:

Esses pinos são os responsáveis por interpretar sinais digitais, essas instruções podem ser enviadas através de um computador, por exemplo, como um dado de entrada e a saída pode ser representada por outros componentes como um *led*, relês ou motores de corrente contínua como os utilizado no projeto.

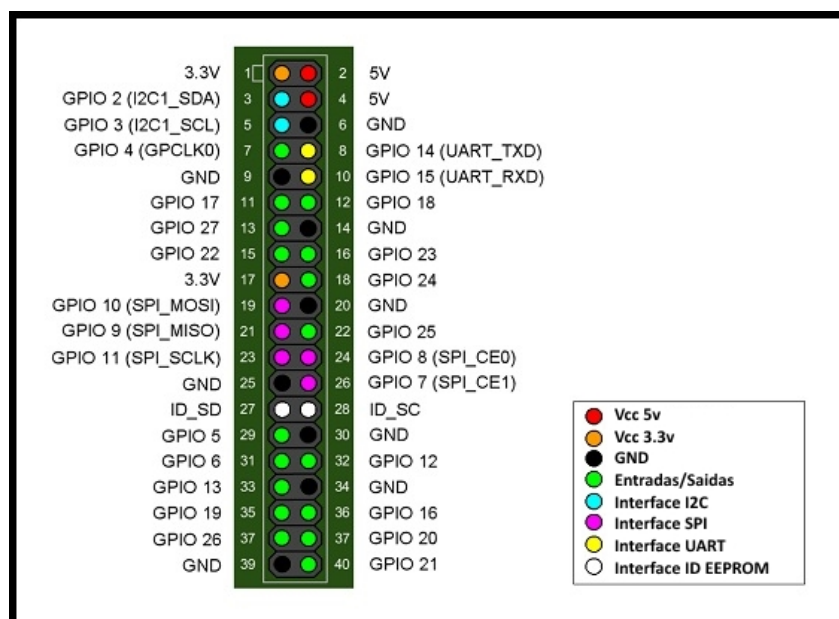


Figura 2 – Diagrama GPIO *Raspberry Pi* Model B+

Fonte: <https://www.rs-online.com/designspark/introducing-the-raspberry-pi-b-plus>

Quando usamos o *Raspberry Pi* como saída a interpretação que ele faz é que cada pino pode ligar ou desligar, ou ir alto ou baixo em termos de computação. Quando o pino é alto, ele produz 3.3 volts, quando o pino é baixo, ele está desligado. As entradas são um pouco mais complicadas por causa da forma como os dispositivos digitais funcionam. o Pi pode ficar confuso e não interpretar corretamente a instrução de entrada realizada. Para isso existem bibliotecas capazes de minimizar esses possíveis erros e informando corretamente o sinal digital para o PI interpretar.

O GPIO é composto por duas portas de 5 *Volts*, qualquer coisa que conectado a esses pinos receberam essa energia. Uma porta que libera 3.3 *Volts* de energia também para conexão de componentes. Oito portas GND (*graduated neutral density filter*), neutros, sem voltagens utilizados para completar circuito. Dezessete GPIO para entrada e saída de dados. Duas portas para interface I2C (Circuito Inter-Integrado), para circuitos integrados montados em uma mesma placa. Cinco portas SPI para comunicação serial *Full Duplex* síncrono, para comunicação externa com componente em casos que forem implementado esse tipo de protocolo. Duas portas UART, utilizam o protocolo RS-232 para envio e recebimento de sinal digital. E duas portas ID SC e ID SD/DNC, que trabalham através de tensão elétrica para armazenagem de dados.

2.5 PYTHON E APLICAÇÕES WEB

Esta seção apresenta conceitos da linguagem de programação Python e as tecnologias que Aplicações web utilizam para serem implementadas.

2.5.1 Linguagem de programação Python

*Python*² é uma linguagem *open-source* (código livre) de alto nível, interpretativa, de *script*, imperativa, orientada a objeto e funcional, a sua idealização é poder proporcionar a criação de aplicativos e projetos de forma simples, rápida e fácil.

Python é uma linguagem expressiva, em que é fácil traduzir o raciocínio em um algoritmo. Por ser uma linguagem interpretativa não necessita que os scripts sejam compilados, os mesmos são diretamente executados.

Bogdanchikov, Zhaparov e Suliyev (2013) defende que a linguagem de programação *Python* é a mais adequada para ser aprendida como primeira linguagem de programação. Porque tem ferramentas poderosas que refletem a maneira como as pessoas pensam e a forma como eles implementar o código. Além disso, apresenta a minimização de palavras-chave adicionais, que estão relacionadas com a escrita correta da sintaxe dos códigos. Esses detalhes podem apresentar um maior rendimento do que no ensino de línguas C++ ou Java, que têm uma grande quantidade de especificidades de língua.

Python pode ser uma opção de aplicação para o ensino dos conceitos básicos de programação para crianças e jovens iniciantes, por proporciona a visualização estimulante do código fonte para compreensão dos programas.

2.5.2 Aplicações Web

Geralmente, aplicações Web, se dividem em duas partes: Front-end, que é a interface gráfica e tudo que o usuário vai ver e Back-end, é a parte que fica no servidor e tem acesso ao banco de dados em determinadas ocasiões.

No Front-end (interface gráfica), as aplicações web são formadas por HTML, CSS e Javascript.

2.5.2.1 HTML

O HTML (*Hyper text Markup Language* ou linguagem de marcação de Hipertext) é uma linguagem de programação que os navegadores de internet conseguem interpretar e fazer a exibição do conteúdo. Para que essas informações sejam exibidas de maneira organizada, é necessário formatar cada trecho delimitando-os dentro de uma marcação, que chamamos de *tags*. As *tags* influenciam na maneira como cada texto, ou parte, será exibido. Sendo assim o HTML é responsável por exibir, organizar e dar significado às informações das páginas *web* (MILETTO; BERTAGNOLLI, 2014).

² <<https://www.python.org/>>

2.5.2.2 CSS

O HTML não apresenta nenhum atrativo estético, foi desenvolvido basicamente para apresentar trabalhos, monografias redigidas e formatadas por metodologias de trabalhos científicos. Porém com o a evolução da *web* foi necessário desenvolver métodos de aplicações de elementos gráficos para deixar as páginas *web* mais atrativas, tendo em vista que ele contava apenas com algumas tags que possibilitam estilizar usualmente as páginas, porém esse método se tornou impraticável.

Surge então o CSS(*Cascading Style Sheets*), responsável por deixar as páginas web mais bonitas e atrativas, customizando o designer dos elementos das páginas. O CSS é uma linguagem, separada do HTML, com o objetivo único de cuidar da estetização da página, ele separa as coisas, regras de estilo não aparecem mais no HTML, apenas no CSS. Ele descreve como os elementos HTML devem ser exibidos na tela, podendo assim controlar o *layout* de várias páginas da *Web* de uma só vez(MILETTO; BERTAGNOLLI, 2014).

O CSS pode ser adicionado aos elementos HTML de 3 maneiras:

Inline (na linha) - usando o atributo de estilo em elementos HTML: é usado para aplicar um estilo exclusivo em linha usa o atributo de estilo de um elemento HTML.

Interno - usando um elemento *style* na seção *head*: é usado para definir um estilo para uma única página HTML.

Externo - usando um arquivo CSS externo: um arquivo externo armazena os estilos a serem usados em um arquivo com a extensão .css, essa opção pode ser usada para definir o estilo para muitas páginas HTML. Para usar essa opção é preciso adicionar um link na seção *head* da página HTML.

2.5.2.3 Javascript

O JavaScript é a linguagem de programação que permite a execução de *scripts* contidos nas páginas dentro do próprio navegador. Suportada por todos os navegadores, a linguagem é responsável por praticamente qualquer tipo de dinamismo a ser executada nas páginas, ou seja através dessa linguagem podemos controlar ações e aplicação das páginas(MILETTO; BERTAGNOLLI, 2014). É uma linguagem interpretativa, que não precisa ser compilada para ser executada.

Para ser executado ela deve ser enviado junto com o HTML para o navegador. É possível fazer um arquivo externo com a extensão .js e importá-lo na sintaxe do HTML (linha 59 da figura 9 apresenta essa requisição). Isso permite que o *script* em JavaScript possa ser utilizado por mais de uma página.

- **Ajax**³: permite que as aplicações *web* façam rapidamente atualizações incrementais para a interface do usuário sem recarregar a página inteira do navegador. Isso torna a aplicação mais rápida e sensível às ações do usuário.

As vantagens de utilizar estes recursos está na maior facilidade para a detecção de erros, por serem peças mais concisas de software; concentração na abstração de soluções do problema que estamos tratando; eficiência na resolução dos problemas, otimização de recursos e de tempo.

A próxima subseção apresenta em detalhes as funções básicas desenvolvidas em JavaScript associadas aos botões dipostos nas interface.

2.5.3 Frameworks

Podemos definir *frameworks* como sendo uma abstração que une códigos comuns entre vários projetos de *software* provendo uma funcionalidade genérica. Um *framework* pode atingir uma funcionalidade específica, por configuração, durante a programação de uma aplicação (MULLER, 2014).

Ou seja, é um conjunto de classes que podemos utilizar no desenvolvimento de uma aplicação. Essas classes podem ser criadas ou podemos encontrar alguns pacotes prontos na internet, que auxiliam e facilitam o trabalho de desenvolver.

Abaixo a descrição dos principais *frameworks* utilizados no desenvolvimento da interface.

- **Bootstrap**⁴: O *Bootstrap* é um *framework front-end* para desenvolvimento de sites responsivos para *web*. Ele é um conjunto de ferramentas de código aberto para desenvolvimento com HTML, CSS e JS apresenta recursos como formas, botões, estilos de navegação, ícones e outros componentes de interface que podem facilitar e agilizar o trabalho de se desenvolver uma página *web*.

- **jQuery**⁵: O *jQuery* é uma grande biblioteca que contém diversas funções que facilitam o desenvolvimento. A sintaxe do *jQuery* foi projetada para facilitar a navegação em um documento, selecionar elementos, criar animações , manipular eventos e desenvolver aplicativos do *Ajax* . *jQuery* também fornece recursos para desenvolvedores para criar *plug-ins* em cima da biblioteca de JavaScript. A abordagem modular da biblioteca *jQuery* permite a criação de dinâmicas e poderosos aplicativos e páginas *web*.

³ <https://www.w3schools.com/xml/ajax_intro.asp>

⁴ <<http://getbootstrap.com/>>

⁵ <<https://jquery.com/>>

2.6 TRABALHOS RELACIONADOS

CANO et al. (1998) define quatro perspectivas de utilização do computador aplicáveis no processo geral de ensino-aprendizagem 1) Aprender informática; 2) usar recursos informáticos projetados especificamente para o ensino (Softwares educacionais); 3) usar aplicações informáticas de base para o tratamento de informação (Internet); 4) aprender a programar. Aprender a programar pode ter como objetivo, aprender uma linguagem de programação para fins profissionais, ou utilizar a programação como um meio para o desenvolvimento de determinados processos cognitivos, para desenvolvimento de habilidades de análise, resolução de problemas, etc.

Gomes e Ghedin (2011) em seu estudo sobre as principais ideias da teoria do desenvolvimento cognitivo de Jean Piaget afirmavam que a inteligência da criança, no período sensório-motor, é prática, sustentada pela manipulação de objetos concretos. Assim essa inteligência é uma adaptação das ações com relação ao meio, ou seja, a forma que a criança realiza uma ação utilizando os objetos externos para aprender resulta no processo de evolução da inteligência. A ação é indispensável para a construção do conhecimento.

Segundo Bossuet e Fischer (1985) LOGO é uma linguagem que permite demonstrar os processos mentais empregados por um indivíduo para resolver os problemas que se lhe apresentam e aos quais ele propõe uma solução, num contexto de ação sobre o mundo exterior. A linguagem LOGO aparece como um excelente recurso para aqueles que iniciam na arte de programar foi desenvolvida no laboratório de Inteligência artificial do MIT – *Massachusetts Institute of Technology* sobre a direção do professor e pesquisador Seymour Papert.

O LOGO é uma linguagem de programação interpretativa e interativa. Que envolve uma tartaruga gráfica em sua interface que responde imediatamente aos comandos inseridos pelo usuário. O programa nas versões mais antigas exibia comandos para pintar e desenhar. A tartaruga movimentava-se produzindo traços na tela de forma com que as crianças rapidamente pudessem visualizar o desenho que desejam. Ao decorrer dos anos novas versões foram surgindo e professores começaram a utilizar o LOGO para trabalhar outras habilidades como geometria, noções de tempo e planejamento.

Por propiciar trabalhos lúdicos como desenhar, o programa chegou a ser utilizado por psicólogos como instrumento auxiliar para tratar crianças com problemas e/ou distúrbios de aprendizagem, que não se remetiam a deficiência de inteligência mas ocorriam por outros fatores de ordem emocional, cognitiva ou motora. O programa propiciava a organização do pensamento dando maiores condições para o desenvolvimento do pensamento operatório.

Atualmente, o LOGO tem comandos para controlar portas paralelas de microcomputador, fazendo com que seus pinos de I/O's (*Input/Output* – Entrada e Saída) trabalhem com níveis lógico de 0 e 1, permitindo desenvolver projetos de robótica, que

possam controlar robôs e mecanismos de desenhos, gerando uma interação conhecimento adquirido e demonstrado e o mundo físico.

No mesmo seguimento, Saleiro et al. (2013) desenvolveu um sistema que baseá-se em um microcomputador *Raspberry Pi* para controlar múltiplos robôs. Com interface baseada em *blockly*⁶, possibilita o encaixe dos blocos para desenvolver programação de orientação para a locomoção dos robôs; este projeto foca-se em alunos do ensino fundamental e possui instruções simples como “seguir em frente” e “girar 90 graus” e pode ser utilizado pelas diversas áreas da educação básica, adaptando-o ao conteúdo do exercício proposto.

Entretanto, a aplicação utilizada no experimento de Saleiro não está disponível ao público, e os robôs, mesmo sendo feitos com materiais acessíveis, dependem de uma placa impressa por encomenda (HEINEN et al., 2015).

O estudo da união de *Raspberry Pi* e *blocklys* também foi abordado em artigo e trabalhos acadêmicos desenvolvido por Heinen et al. (2015), o projeto nomeado de RASPIBLOCOS tem como objetivo a construção de uma ferramenta que possibilitasse o desenvolvimento de lições de programação estimulantes e dirigidas à experimentação.

Com uma interface baseada em *blocklys*, blocos coloridos encaixáveis permite desenvolver pequenos programas, quais as instruções são enviadas através de componentes eletrônicos para um microcomputador *Raspiberry Pi* que efetuava a interpretação dos movimentos resultando em uma reação visual.

O autor aponta que o uso do *Raspberry Pi* como hospedeiro da aplicação possibilita que toda a programação seja desenvolvida em uma mesma linguagem, no caso, *JAVAScript*. A distribuição do projeto com código aberto volta-se ao objetivo de desenvolver lições de programação para a graduação criando modelos e blocos gradativamente mais complexos.

⁶ <<https://blockly-games.appspot.com/>>

3 METODOLOGIA

O trabalho desenvolvido segue o modelo experimental e busca desenvolver uma aplicação web para programação visual, e um modelo robótico (um carrinho) capaz de interpretar os comandos instruídos em uma interface web através de programação visual.

3.1 VISÃO GERAL

A Figura 3 mostra a representação gráfica dos os componentes utilizados e nos da uma visão geral do projeto proposto.

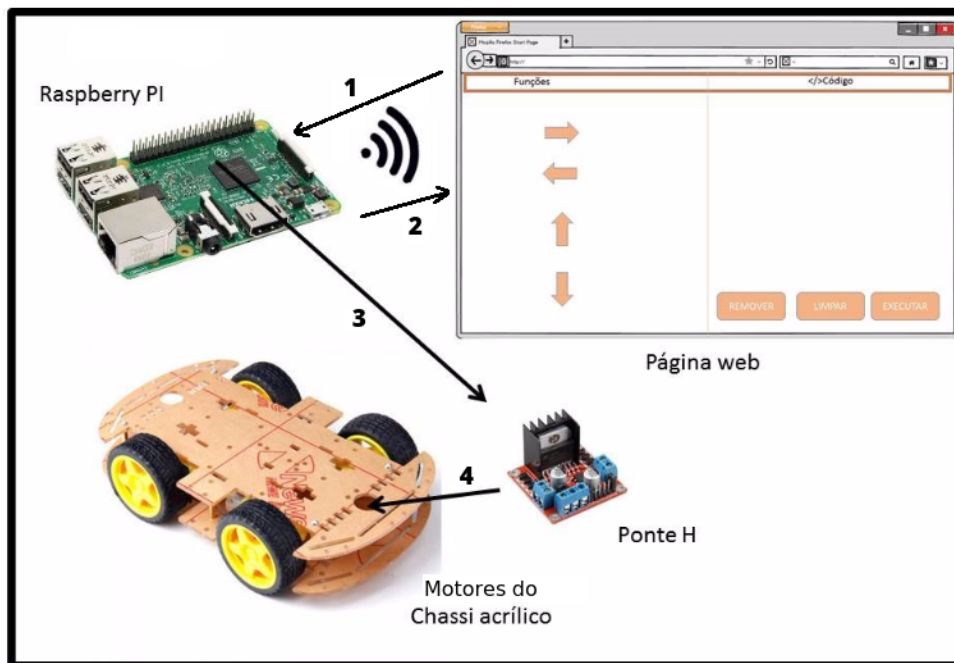


Figura 3 – Visão Geral do Projeto

Fonte: Imagem reproduzida pelo autor

A etapa 1 da figura representa uma requisição ao endereço *ip* do servidor web do *Raspberry Pi* feito através de um navegador web qualquer, seja de um desktop ou de um dispositivo *mobile*. Esta requisição é possível através de uma conexão *wifi* entre o dispositivo o *Raspberry Pi*, sendo este o Ponto de Acesso que fornecerá via *DHCP* um *ip* para o cliente. A etapa 2 representa a resposta do servidor web do *Raspberry Pi* contendo os códigos do front-end da página junto com imagens e arquivos necessários. Quando o usuário da página web clicar no botão executar, ele chamará a etapa 3 do processo que é o envio do comando solicitado pelo usuário para o servidor web do *Raspberry Pi* que automaticamente chamará uma função de ativação do componente eletrônico ponte-H

(que será explicado na próxima seção) que ativará os motores do chassi do carrinho (etapa 4), fazendo-o com que ande de acordo com o que foi solicitado pelo usuário da página web.

O robô deverá ser capaz de interpretar e reproduzir os movimentos programados com as funções para controle de movimento para frente, para a direita, para a esquerda, para trás apresentados na interface. A interface apresenta também a opção de remover o comando, e limpar para iniciar as instruções novamente.

3.2 HARDWARE - PARTE FÍSICA

Esta seção apresenta como a parte física do projeto é estruturada. Em suma, o projeto consiste em um chassi de plástico com quatro rodas e quatro motores elétricos de corrente contínua, estruturado como um carro, uma ponte H para interligar o GPIO do *Raspberry Pi* e os motores elétricos. A seguir, será explicado cada parte dessa detalhado, separado em parte mecânica e esquema elétrico.

3.2.1 Parte Mecânica

O *kit* adquirido para a montagem do veículo robótico é composto por 4 motores DC (*Direct Current*), 3-6 *Volts* com caixa de redução 48:1 e tração nas 4 rodas 4WD. O chassi de acrílico possui duas plataformas que permitem espaçamento entre os motores. Ele é um dos mais indicados para aplicação robótica por conter uma plataforma capaz de acoplar componentes eletrônicos e permitir a adição de outras partes mecânicas se for necessário. Representados naa figura 4.



Figura 4 – Kit chassi para robótica

Fonte: Imagem reproduzida pelo autor

3.2.2 Esquema Elétrico e Eletrônico

O esquema elétrico e eletrônico é composto basicamente de uma uma Ponte H para fazer com que os motores funcionem de acordo com um sinal elétrico chegue nele, provido pelo *Raspberry Pi* e uma bateria de 9V.

Os módulos ponte H possuem circuitos básicos para o uso dos CI (circuitos integrados), são utilizados em aplicações de robótica quando a necessidade de utilizar motores DC (*direct current* ou corrente contínua), por demandarem da utilização de corrente elétrica com tensão maior que a fornecida por componentes como o *Raspberry Pi*, os motores não devem ser ligados diretamente na placa pois podem danificá - la. Através dela também é possível controlar a sentido de giro do motor.

A Ponte H utilizada foi a L298N, a figura 5 apresenta seu esquema.

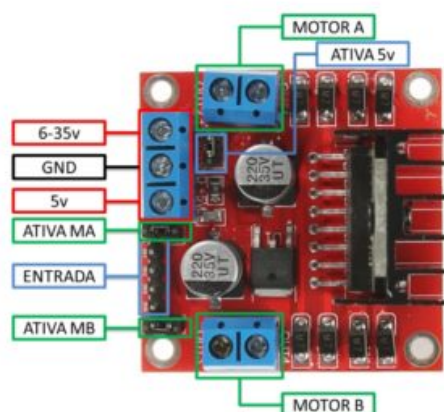


Figura 5 – Diagrama Ponte H L298N

Fonte: <https://portal.vidadesilicio.com.br/modulo-ponte-h-l298n-arduino/>

Possui duas entradas para conectar motor A e B, uma entrada de 6-35 Volts para alimentação da placa. Uma entrada de 5 Volts para quando não há alimentação com mais de 6 Volts. Um GND neutro para fechamento de circuito. Ativa MA pode ser utilizado para controlar a velocidade do motor A e a Ativa B utilizada para controlar a velocidade do motor B, quando removido o jumper esse controle pode ser feito através da voltagem utilizada para alimentação onde 0 Volts é mínimo (parado) e 5 Volts é a velocidade máxima. São quatro entradas que podemos chamar de IN. A IN1 e a IN2 controlam as ações realizadas pelo motor 'A' enquanto a IN3 e a IN4 controlam as ações realizadas pelo motor 'B'.

Também foi utilizado uma bateria 9V para alimentar a ponte H, um carregador sem fio para alimentar o *Raspberry Pi* e cinco jumpers fêmea-fêmea utilizados para fazer a ligação entre os pinos do GPIO e as entradas da ponte H.

O esquema de ligações de componentes descritos é apresentado na figura 6.

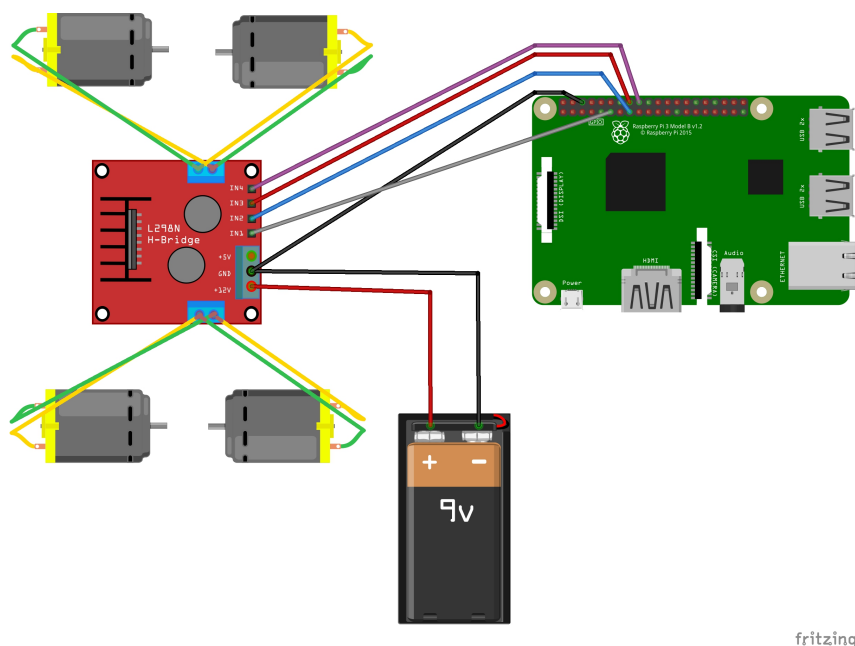


Figura 6 – Esquema elétrico

Fonte: Imagem reproduzida pelo autor

No projeto foi utilizado quatro motores de corrente contínua, que acompanhavam o chassi de acrílico adquirido, a ponte H tem entrada somente para dois, então foi necessário utilizar uma saída para cada dois motores. Na montagem do chassi esses motores ficaram dispostos de frente um para o outro. Sendo assim foi necessário, inverter o sentido de rotação dos mesmo para que os motores que ficaram do mesmo lado rodassem no mesmo sentido. Então foi alterado a ligação feita na porta de saída da ponte H, fio verde de um motor com fio amarelo de outro motor, como representado na figura, na ligação física esses fios são vermelho e preto.

3.3 SOFTWARE - PARTE LÓGICA

Esta seção mostra como as partes lógicas foram desenvolvidas. Mostrando sobre python, interação com GPIO, a interação entre PHP e python, o javascript para enviar para o PHP a programação feita pelo HTML e CSS que, por fim, implementam a interface da aplicação web para interação com o usuário. A seguir será explicado de maneira mais detalhada cada parte citada.

3.3.1 Python e o GPIO

A linguagem *Python* é uma linguagem nativa do *Raspberry Pi*, o sistema consegue interpretá-la diretamente no terminal ou em alguns aplicativos que vem junto com o

sistema *RaspBian*. Sendo assim ela consegue interagir diretamente com as portas GPIO do *Raspberry Pi*. O `RPi.GPIO` é pacote da linguagem *Python* que possui licença MIT, e fornece uma classe que permite controlar o GPIO do *Raspberry Pi*, e em poucas linhas foi possível configurar o modo, pino, direção da ação desejada.

A figura 7 apresenta uma função utilizando o pacote `RPi.GPIO` utilizado para interagir com o GPIO do *Raspberry Pi*, e logo abaixo a suas descrições.

```
import RPi.GPIO as gpio
import time

def init():
    gpio.setmode(gpio.BCM)
    gpio.setup(17, gpio.OUT)
    gpio.setup(22, gpio.OUT)
    gpio.setup(23, gpio.OUT)
    gpio.setup(24, gpio.OUT)

def frente(tf):
    init()
    gpio.output(17, True)
    gpio.output(22, False)
    gpio.output(23, True)
    gpio.output(24, False)
    time.sleep(tf)
    gpio.cleanup()

frente(1)
```

Figura 7 – Função em Python

Fonte: Imagem reproduzida pelo autor

import RPi.GPIO as gpio: importa a função e renomeia para que a mesma possa ser referida apenas como `gpio` pelo resto do *script*.

import time: importa a função de tempo, que será setado para a execução de cada comando.

def init(): declaração de uma função de inicialização.

gpio.setmode(gpio.BCM): define que os pinos utilizado serão referenciados pela numeração lógica do *Raspberry Pi* ou também chamado de numeração "*Broadcom SOC Channel*", estes números são identificados após "GPIO"no esquema de pinos do GPIO do *Raspberry Pi*, esse esquema lógico pode mudar de acordo com o modelo da placa do *Raspberry Pi*. Se fossemos referenciar os pinos pela sua posição física na placa deveria ser utilizado o comando `gpio.setmode(gpio.BOARD)`.

gpio.setup(17, gpio.OUT): configura o pino, posição lógica definida anteriormente, e a direção. Isso acontece com os quatros pinos utilizados.

def frente(tf): Definindo as funções de movimento.

init(): chama a função de inicialização definida anteriormente.

gpio.output(17, True)

gpio.output(22, False)

gpio.output(23, True)

gpio.output(24, False): definem a saída da ação, essas funções definidas como no exemplo, vão fazer com que o movimento dos quatro motores sejam para o mesmo lado, como para a frente.

time.sleep(tf): Define que essa função será executada pelo tempo determinado.

gpio.cleanup(): No final do programa é utilizado um comando para limpar os recursos usados. Isso só irá limpar os canais GPIO que o *script* usou.

frente(1): executa o programa e define o tempo, nesse exemplo a ação irá durar 1 (um) minuto;

O pacote de RPi.GPIO foi utilizado para desenvolver as quatro funções de movimento principais do projeto. Sendo elas, `def frente(tf)`, `def tras(tf)`, `def virar-direita(tf)` e `def virar-esquerda(tf)`. Ao receber esses comandos em linguagem *Python* o GPIO do *Raspberry Pi* interpreta e libera os dados de saída pelos pinos definidos, estes pinos estão interligados com os pinos de entrada da Ponte H, e ficam responsáveis por controlar a tensão e enviar o comando para os motores de corrente contínua, que estão ligados nas portas de saída da ponte H e executam os comandos.

3.3.2 Servidor Web e Liguagem de Programação (Back-end)

Para que um computador externo acesse o *Raspberry Pi*, é necessário ter uma rede com uma faixa de IP necessária para que eles consigam se conectar. Para isso, a ideia é transformar o *Raspberry Pi* (que já possui uma placa *Wifi* nativa em um Ponto de acesso que libera IP automaticamente usando DHCP. A ação de ativação e liberação do sinal de *wifi*, ocorre devido a configuração dos pacotes *hostapd* e *dnsmasq*¹ que permitem configurar e usar o *Raspberry pi* como um ponto de acesso.

Ao acessar o ip do *Raspberry pi* a intranet, através de um navegador (Chrome, Firefox, Edge...), este se comunica com um servidor, fazendo requisição e recebendo dados dos arquivos hospedados neles como resposta. Um servidor *Web* (ou um conjunto de servidores) é responsável por processar todas essas informações de requisição ou transferência de dados solicitados pelos usuários e exibi-la na tela.

O servidor web utilizado no projeto foi instalado no *Raspberry PI*, ele hospeda a aplicação para que assim possamos acessa-la de um computador ou de um celular através de

¹ <<https://www.raspberrypi.org/documentation/configuration/wireless/access-point.md>>

um navegador. O *software* de servidor *web* utilizado foi o Apache footnote<www.apache.org> que é distribuído pela *The Apache Software Foundation* sob a licença GNU, ou seja, é gratuito e pode ser estudado e modificado através de seu código fonte por qualquer pessoa (ALECRIM, 2006).

O Apache consegue interpretar várias linguagens de programação dentre elas a linguagem PHP(*Personal Home Page*)² que foi utilizada como intermediária da requisição web em JavaScript com os comando de movimento desenvolvida em *python*. O PHP é uma linguagem de programação interpretativa usada originalmente apenas para o desenvolvimento de aplicações *web*, e também é livremente distribuída.

A figura 8 exibe a função desenvolvida em PHP. Ela recebe um *array*(conjunto de elementos) de comandos armazenado pelo *JavaScript* em uma variável. Cada comando armazenado nessa variável é separado por ",". Para percorrer esse *array* e associar o comando recebido da interface com o comando que deve ser executado em *python* é o utilizado o laço de repetição "*foreach*", em uma estrutura de "se", "senão se"ele faz essa relação de comparação entre os comandos correspondentes definidos e chama o comando *python* para ser executado.

```
1 <?php
2 include("Ssh.class.php");
3 if($_POST != null){
4     $com = $_POST["comandos"];
5     $ssh = new Ssh("root", "127.0.0.1", "22", "rasp##");
6
7     $ssh->conectar();
8     $COM = explode(",", $com);
9
10    foreach($COM as $COM1){
11        if(strpos($COM1, "up") !== FALSE){
12            $ex = "python /home/pi/movimento/frente.py";
13        }else if(strpos($COM1, "down") !== FALSE){
14            $ex = "python /home/pi/movimento/tras.py";
15        }else if(strpos($COM1, "left") !== FALSE){
16            $ex = "python /home/pi/movimento/esquerda.py";
17        }else if(strpos($COM1, "right") !== FALSE){
18            $ex = "python /home/pi/movimento/direita.py";
19        }
20        $ssh->executarComando($ex);
21        echo $ssh->pegarSaida();
22    }
23    $ssh->desconectar();
24 }
25 }
```

Figura 8 – Código em PHP para interpretar os comandos provenientes do *JavaScript* e sua consequente chamada do Python

Fonte: Imagem reproduzida pelo autor

Foi incluído uma classe de acesso seguro *Ssh* (*Secure Shell*). Essa classe permite acessar os GPIO do *Raspberry Pi* como usuário *root*(administrador) e manipula-los de forma segura, e assim poder executar os comando enviado em *python*.

² <https://secure.php.net/manual/pt_BR/intro-what-is.php>

3.3.3 Interface Gráfica Web (Front-end)

Até este ponto foi demonstrado como foi construída toda a parte interna do robô. Para que a aplicação tenha uma interação com o usuário para que ele escolha a programação que quiser que o carrinho realize, é necessária construção de um módulo de interface gráfica com o usuário. Esta parte é o que chama-se de *Front-end*, a parte visual de um aplicação web.

Para isso, um protótipo inicial foi feito para dar base à implementação a esta interface gráfica e pode ser visto na Figura 9.

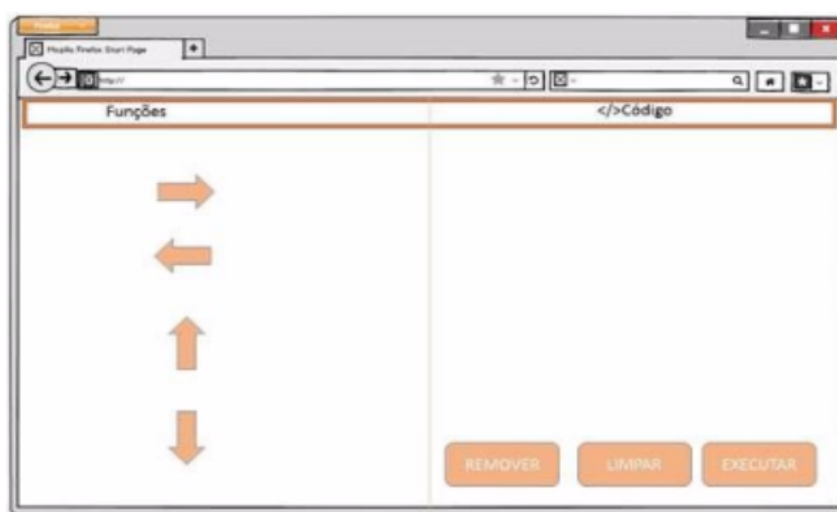


Figura 9 – Protótipo do *Front-End* da aplicação web

Fonte: Imagem reproduzida pelo autor

Para desenvolver a interface gráfica da página *web* foram utilizadas as linguagens HTML, CSS (com Bootstrap) e JavaScript (com jQuery). As figuras 10 e 11 exibem a página *web* do projeto em HTML com aplicação dos recursos do CSS.

```
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
    <meta name="description" content="">
    <meta name="author" content="">
    <title>RaspRobo</title>
    <!-- Bootstrap core CSS -->
    <link href="css/bootstrap.min.css" rel="stylesheet">
    <!-- Custom styles for this template -->
    <link href="css/style.css" rel="stylesheet">
    <link href="css/font-awesome.min.css" rel="stylesheet">
  </head>
  <body background="fundo.jpg">
    <nav class="navbar navbar-expand-md navbar-dark fixed-top" style="padding-top:2%;">
      <b style="color: #4B0082; font-size: 44px; font-family: sans-serif;"> RaspRobo</b>
    </nav>
    <div class="container" background="fundo.jpg" style="padding-top:3%;">
      <div class="content">
        <div class="row">
          <div class="col-md-3">
```

Figura 10 – Código fonte do *Front-end* da aplicação web desenvolvida (Parte1)

Fonte: Imagem reproduzida pelo autor

```

        <button type="button" style="background-color:#00BFFF" class="btn btn-secondary btn-block btn-lg" id="up"><i
class="fa fa-arrow-circle-up" aria-hidden="true"></i> Para Frente</button>
        <button type="button" style="background-color:#00BFFF" class="btn btn-secondary btn-block btn-lg" id="right">
<i class="fa fa-arrow-circle-right" aria-hidden="true"></i> Direita</button>
        <button type="button" style="background-color:#00BFFF" class="btn btn-secondary btn-block btn-lg" id="left">
<i class="fa fa-arrow-circle-left" aria-hidden="true"></i> Esquerda</button>
        <button type="button" style="background-color:#00BFFF" class="btn btn-secondary btn-block btn-lg" id="down">
<i class="fa fa-arrow-circle-down" aria-hidden="true"></i> Voltar</button>
        <hr>
        <button type="button" style="background-color:#000000" class="btn btn-secondary btn-block btn-lg"
id="executar"><i class="fa fa-play-circle" aria-hidden="true"></i> Executar</button>
        <button type="button" style="background-color:#FFA500" class="btn btn-secondary btn-block btn-lg"
id="limpar"><i class="fa fa-undo" aria-hidden="true"></i> Recomeçar</button>
        <button type="button" style="background-color:#FF0000" class="btn btn-secondary btn-block btn-lg"
id="remover"><i class="fa fa-trash-o" aria-hidden="true"></i> Remover</button>
    </div>
    <div class="col-md-9">
        <figure id="container-comandos" class="highlight" style="background-color: rgba(50, 50, 50, 0.1)"></figure>
    </div>
</div>
</div>
<!-- /.container -->
<script src="js/jquery.min.js"></script>
<!-- Bootstrap core JavaScript
===== -->
<!-- Placed at the end of the document so the pages load faster -->
<script src="js/popper.min.js"></script>
<script src="js/holder.min.js"></script>
<script src="js/anchor.min.js"></script>
<script src="js/bootstrap.min.js"></script>
<!-- IE10 viewport hack for Surface/desktop Windows 8 bug -->
<script src="js/ie10-viewport-bug-workaround.js"></script>
<script src="js/scripts.js"></script>
</body>
</html>

```

Figura 11 – Código fonte do *Front-end* da aplicação web desenvolvida (Parte 2)

Fonte: Imagem reproduzida pelo autor (cont.)

Com o HTML e CSS (utilizando o *framework bootstrap*) criou-se uma página de exibição, que apresentava informações mas não continha nenhuma ação. Para implementar ações aos botões foi necessário utilizar a linguagem JavaScript para programar as funções que foram dispostas no HTML e que seriam interpretadas pelo PHP para este fazer requisição de execução ao *python*, e este fará o acionamento dos motores respectivos aos movimento e a direção deles (frente ou trás).

Para isso, foram implementados alguns botões que foram associados às funções programadas em *JavaScript*, utilizando o *jQuery* com ações a serem executadas quando são clicados.

Ao clicar no botão "Para Frente", por exemplo, uma seta deve ser adicionada na área delimitada ao lado. Esta seta é a representação gráfica da direção que o protótipo robótico deve se locomover. A função programada para que isso aconteça é apresentado abaixo, e faz o robo ir para FRENTE.

```

$('button#up').click(function(){
    $('figure#container-comandos').append(returnHtmlComando('up'));
    $('span.comando').click(function(){
        $('span.comando').removeClass('text-danger');
        $(this).addClass('text-danger');
        comando_selecionado = $(this);
    });
});

```

Este é o trecho de código que é executado ao acionar o botão "Para Frente". Essa mesma ação se repete se clicarmos nos botões "Direita", "Esquerda" e "Voltar". Setas de direcionamento são adicionadas na área delimitada ao lado. Desta forma, pode-se criar uma sequência lógica de movimentos que deseja-se executar, como pode ser visto abaixo.

Este é o trecho de código jQuery que será executado para mandar o comando para virar o robo para a TRÁS.

```
$('#button#down').click(function(){
    $('#figure#container-comandos').append(returnHtmlComando('down'));
    $('#span.comando').click(function(){
        $('#span.comando').removeClass('text-danger');
        $(this).addClass('text-danger');
        comando_selecionado = $(this);
    });
});
```

Este é o trecho de código jQuery que será executado para mandar o comando para virar o robo para a ESQUERDA.

```
$('#button#left').click(function(){
    $('#figure#container-comandos').append(returnHtmlComando('left'));
    $('#span.comando').click(function(){
        $('#span.comando').removeClass('text-danger');
        $(this).addClass('text-danger');
        comando_selecionado = $(this);
    });
});
```

Este é o trecho de código jQuery que será executado para mandar o comando para virar o robo para a DIREITA.

```
$('#button#right').click(function(){
    $('#figure#container-comandos').append(returnHtmlComando('right'));
    $('#span.comando').click(function(){
        $('#span.comando').removeClass('text-danger');
        $(this).addClass('text-danger');
        comando_selecionado = $(this);
    });
});
```

A funcionalidade de remover uma seta não mais necessária ou limpar todas as setas de uma vez pelo usuário final também foram implementadas usando jQuery via JavaScript e estão descritas abaixo.

```
$('#button#remover').click(function(){
    if(comando_selecionado !== null){
        comando_selecionado.remove();
    }
});

$('#button#limpar').click(function(){
    $('#figure#container-comandos').html('');
});
```

Os comandos de movimentos inseridos na área delimitada criam o *array*. Ao clicar no botão "Executar" esse *array* é armazenado em uma variável, a função é enviada para o PHP. Este *array* é associado às funções em *python* como citado na figura 7. A função desenvolvida para que isso aconteça é apresentada em jQuery via JavaScript conforme mostrado abaixo.

```
$('#button#executar').click(function(){
    enviarComandos();
});

enviarComandos = function(){
    comandos = [];
    $('#span.comando').each(function(index){
        comandos.push($(this).data('comando'));
    });
    var array_comandos = comandos.toString();
    $.post( "action.php", { comandos: array_comandos })
    .done(function( data ) {
        console.log(data);
    });
};
```

Este capítulo mostrou como foram desenvolvidas as funcionalidades que envolvem o projeto proposto. O próximo capítulo irá demonstrar estas em conjunto com o robô funcional e também a interface gráfica.

4 RESULTADOS E DISCUSSÃO

Este capítulo descreve os resultados da implementação do sistema proposto. A primeira seção descreve como um usuário final aciona o robô e como faz as conexões necessárias para sua utilização. A segunda seção faz a demonstração dos resultados obtidos através da implementação do sistema. E por fim, descreve a discussão do trabalho.

4.1 UTILIZAÇÃO DO SISTEMA

O projeto descrito constitui-se de uma aplicação *web* hospedada em um minicomputador *Raspberry Pi*, que pode ser acessado por um navegador através de uma rede *wifi*, disponibilizado pelo próprio minicomputador, capaz de controlar os movimentos de um modelo robótico através de instruções programadas em sua interface. A união dos *hardware* descrito resultou no protótipo robótico exibido na figura 12.

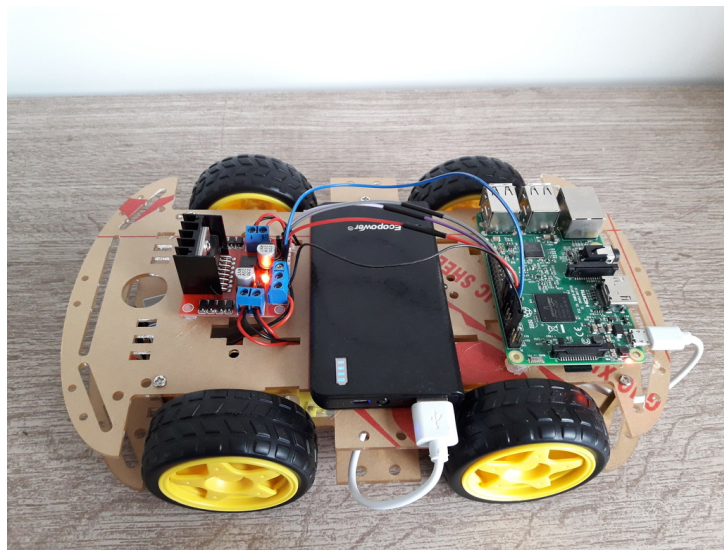


Figura 12 – Protótipo robótico final

Fonte: Imagem reproduzida pelo autor

Ao ligar o *Raspberry Pi* ele ativa e libera o sinal de *wifi*. Ao conectar um dispositivo a essa rede inserindo a senha de segurança, e é possível abrir a aplicação através de um navegador de internet (ex. *Chrome*), digitando o IP setado no *Raspberry pi* barra o nome da aplicação (10.199.199.254/rasprobo). Essa ação dá acesso à interface da aplicação que foi hospedada no *Raspberry Pi*. O usuário insere os comandos desejados na interface e ao clicar no botão "Executar" o modelo robótico reproduz esses movimentos na sequência lógica que foram programados.

4.2 DEMONSTRAÇÃO DO SISTEMA

A figura 13 apresenta a interface desenvolvida em sua versão desktop, com alguns comandos de movimento inseridos para exemplificar uma possível programação visual.

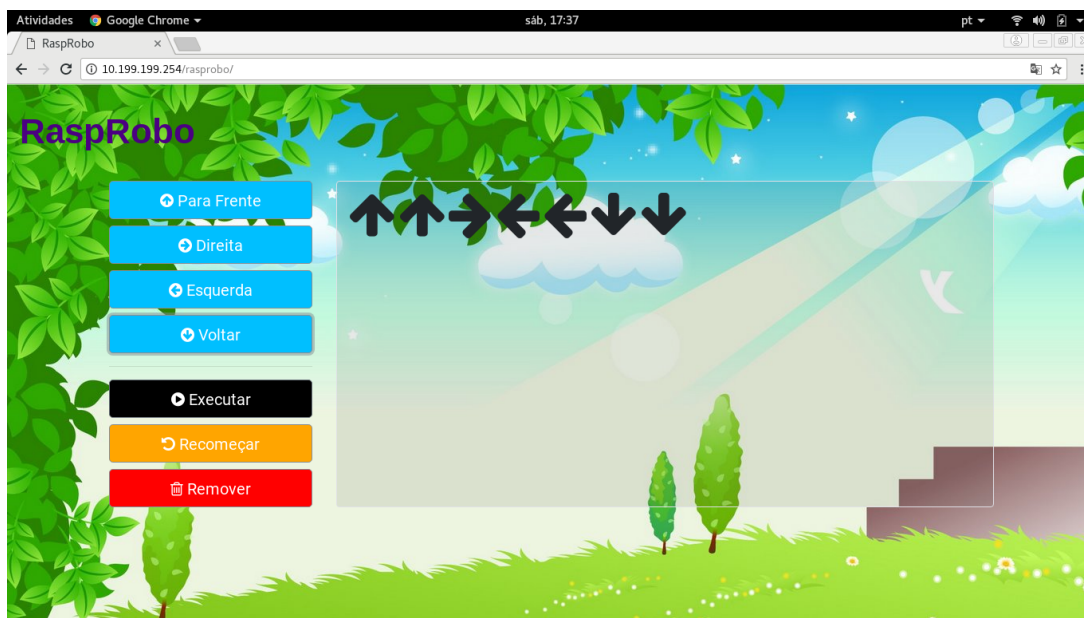


Figura 13 – Interface versão desktop

Fonte: Imagem reproduzida pelo autor

A figura 14 apresenta a interface do projeto na versão mobile, acessada por um navegador de celular.

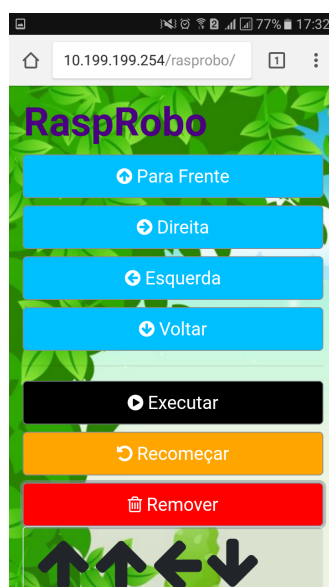


Figura 14 – Interface versão mobile

Fonte: Imagem reproduzida pelo autor

A figura 15 mostra de maneira prática como funciona a adição da funcionalidade para fazer o robô ir para frente durante 1 segundo. Basta o usuário clicar no botão azul chamado "Para Frente". Após isso, uma seta preta irá aparecer do lado esquerdo da tela.

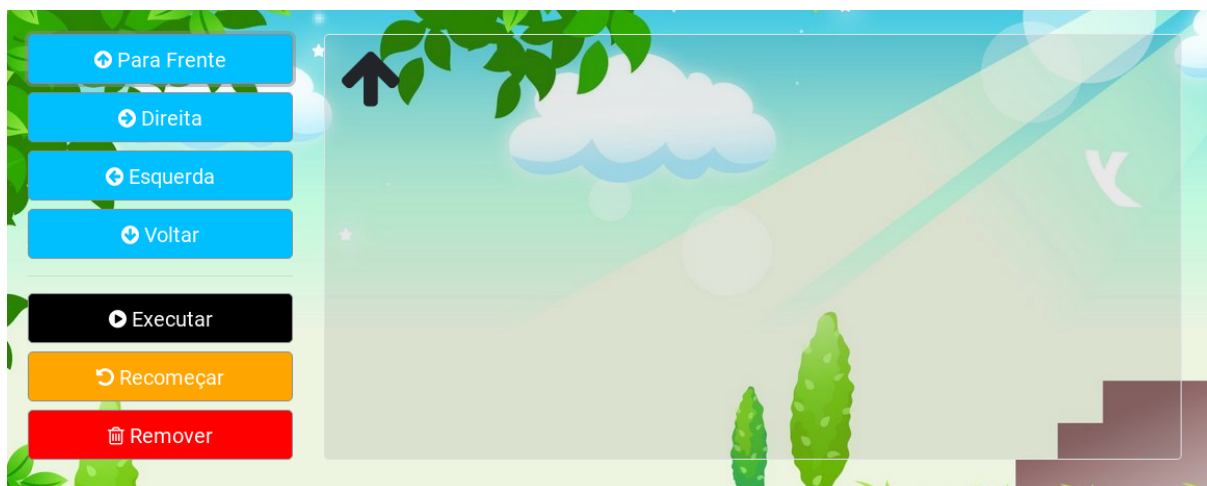


Figura 15 – Botão Frente

Fonte: Imagem reproduzida pelo autor

No exemplo da figura 16, está mostrando, o usuário final está programando o robô ir para frente, depois virar para a direita, virar para a esquerda e por fim, voltar de ré.

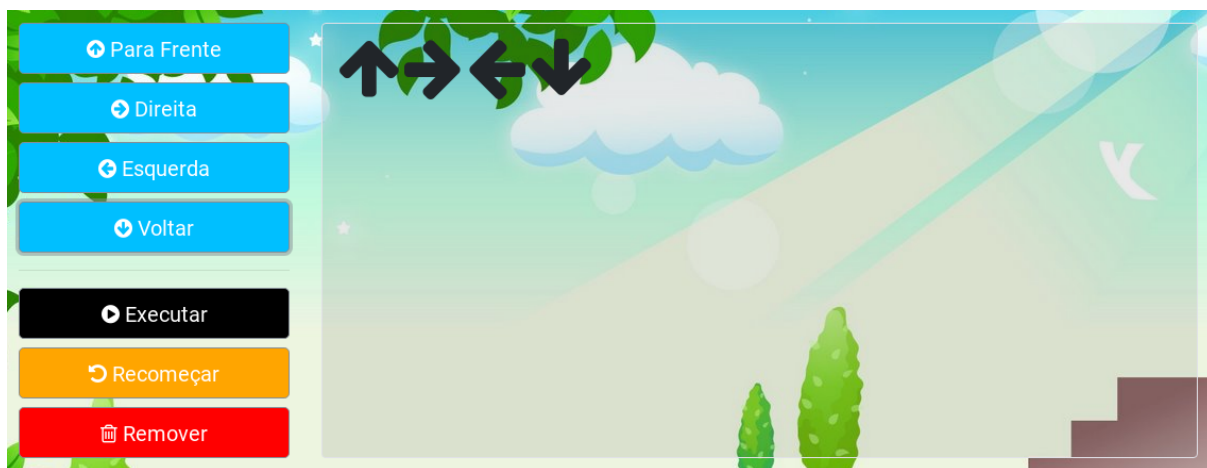


Figura 16 – Botões Para Frente, Direita, Esquerda e Voltar

Fonte: Imagem reproduzida pelo autor

Se um comando for inserido errado, pode-se removê-lo clicando sobre a seta, A mesma ficará vermelha (figura 17 (A)) e ao clicarmos no botão "Remover", ela é removida da área delimitada (figura 17 (B)).

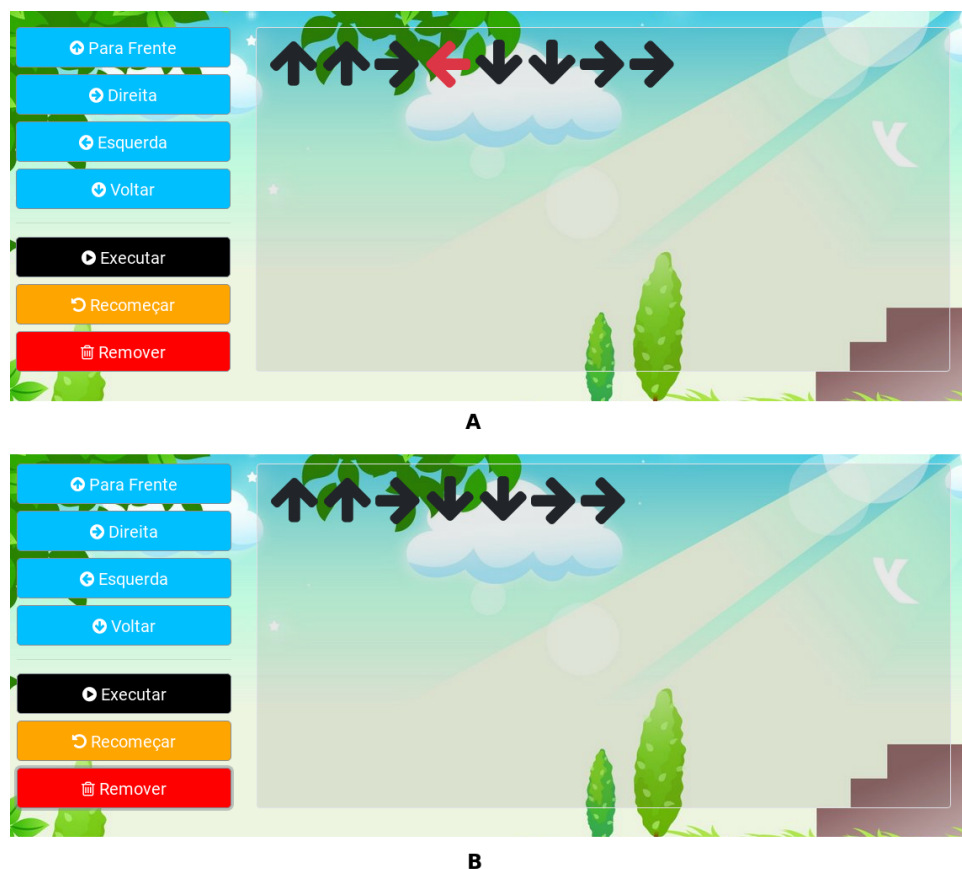


Figura 17 – (A) Seta Seleccionada; (B) Seta removida

Fonte: Imagem reproduzida pelo autor

A funcionalidade de limpar os dados foi implementada através do botão "Recomeçar" que ao ser clicado remove todas as setas inseridas (Figura 18), limpando a área delimitada, permitindo assim iniciar outra sequência de comandos.

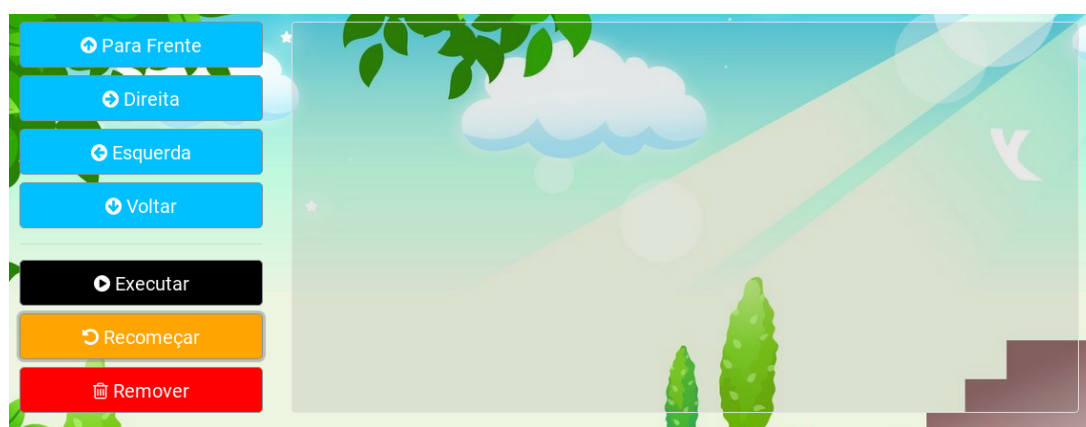


Figura 18 – Botão Recomeçar

Fonte: Imagem reproduzida pelo autor

4.3 DISCUSSÃO

Este projeto foi construído com o intuito de instigar crianças a aprenderem sucintamente programação visual e estimular o raciocínio lógico, a robótica é utilizada nesse trabalho tem o intuito de despertar o interesse no fazer e ver acontecer que é o que o projeto propõe. A figura 19 exhibe a versão final hardware e software do projeto proposto totalmente implementado e funcional.



Figura 19 – Projeto final

Fonte: Imagem reproduzida pelo autor

O código deste trabalho está disponível na página de compartilhamento de código colaborativo *github* (www.github.com/rasprobo)¹ para receber melhorias por quem se interessar em trabalhar em seu código de maneira colaborativa ou por pessoas que queirão implementar e colocar para funcionar em suas escolas, sendo uma contribuição com a sociedade, fazendo com que o acesso às informações se propaguem e gere conhecimento à quem estiver proposto a buscá-lo.

¹ <<https://www.github.com/rasprobo/>>

5 CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS

A popularização de minicomputadores e microcontroladoras tornou acessível a realização de experiências em robótica e programação com eletrônica básica. Comunidades ao redor do mundo surgiram em volta destas tecnologias, compartilhando informações e apoiando a realização de novas experiências. Beneficiando a área robótica na educação pelo decorrente acúmulo e riqueza de informações.(HEINEN et al., 2015)

Papert (1985) em sua pesquisa utilizava-se de tecnologias para criar experimentos e modelos interativos. Hoje em dia, seus estudos são percebidos como responsáveis por incentivar o aperfeiçoamento de habilidades de raciocínio e resolução de problemas, e por alcançar melhores resultados, quando comparada com aulas expositivas tradicionais.

As crianças podem aprender mais quando visualizam o que construíram, além de intuitivo, permite expandir a curiosidade sem restringirem-se a conteúdos específicos. Diversos fatores apontam a relevância do ensino de lógica de programação para crianças. O contato com a computação possibilita desenvolver habilidades nas mais variadas áreas do conhecimento, tais como aplicações e técnicas para identificar e resolver problemas.

O uso das tecnologias na educação tem que ser incentivado, tanto o uso tradicional de computadores quanto métodos e opções alternativas que possam incluir esses recursos a didática aplicada pelo professor, podendo assim não só possibilitar a expansão do crescimento e o desenvolvimento do aluno, mas também prepara o professor para trabalhar com seus alunos que já nasceram nessa era digital.

O presente projeto teve como objetivo apresentar e incentivar o ensino de lógica de programação e robótica de uma forma diferente e indutiva. O desenvolvimento do projeto baseia-se em iniciativas open-source e continuará nessa mesma linha, para que pesquisadores e colaboradores possam agregar conhecimentos e possam expandi-lo.

Ao término deste projeto a resalta-se a afirmação de Cruz (2008) que o conhecimento não possa ser transmitido nem de máquina para pessoa, nem de pessoa para pessoa, pois não é passível de ser recebido pronto. Ele é construído a partir de experiências anteriores e do contato com o mundo, necessitando, portanto, ser (re)feito por cada indivíduo ao decorrer de cada ação executada.

Sendo assim incentivar que o aluno experimente, manipulando características e conceitos de um ambiente repleto de informações é contribuir para o desenvolvimento de seu próprio conhecimento.

5.1 TRABALHOS FUTUROS

O projeto encontra-se em uma forma funcional, porém há sugestões de melhorias e complementos. Considera-se a possibilidade de fazer um manual passo a passo para a montagem do protótipo robótico, instalação e configuração dos recursos utilizados. Disponibilizar esse manual e a aplicação para que outras pessoas possam colaborar com o projeto além de poderem adquirir o material utilizado e montar o seu próprio "Rasprobo"(código disponível em www.github.com/rasprobo¹).

Na interface, sugere-se a implementação de outros movimentos e também de laços de repetição para incentivarem mais a lógica de programação, ampliando a aplicação e, conseqüentemente, a utilização do projeto em atividades voltadas para o ensino-aprendizagem.

Sugere-se também o levantamento de um estudo a durabilidade dos recursos utilizados tais como as baterias. E como a diminuição de sua carga pode influenciar no desempenho da realização dos movimentos.

Sugere-se a realização de um estudo aprofundado sobre psicometria na área de cores, podendo compreender e desenvolver melhorias nas cores utilizadas na interface, além da possibilidade de poder utilizar somente símbolo ao invés de palavras que descrevem a função dos botões e entender qual o impacto e em qual nível de idade e conhecimento isso pode surtir mais efeito na compreensão da atividade proposta.

E por fim, sugere-se trabalhar com oficinas, sala multidisciplinar ou adaptar a utilização do robô de acordo com o conteúdo trabalhado em sala de aula, talvez engajar em algum projeto social e aplicar atividades para buscar na prática o resultado que almejado.

¹ <<https://www.github.com/rasprobo/>>

REFERÊNCIAS

- ALECRIM, E. Conhecendo o servidor apache (http server project). *São Paulo*, 2006. Citado na página 34.
- ALTOÉ, A.; FUGIMOTO, S. M. A. Computador na educação e os desafios educacionais. In: *Congresso Nacional de Educação-EDUCERE, 9ª. Encontro Sul Brasileiro de Psicopedagogia*. [S.l.: s.n.], 2009. Citado na página 12.
- ARGENTO, H. *Teoria construtivista*. 2009. Citado na página 16.
- BOGDANCHIKOV, A.; ZHAPAROV, M.; SULIYEV, R. Python to learn programming. In: IOP PUBLISHING. *Journal of Physics: Conference Series*. [S.l.], 2013. v. 423, n. 1, p. 012027. Citado na página 23.
- BOSSUET, G.; FISCHER, L. M. V. *O computador na escola: o sistema logo*. [S.l.: s.n.], 1985. Citado na página 26.
- CANO, C. A. et al. Os recursos da informática e os contextos de ensino e aprendizagem. *Para uma tecnologia educacional*. Porto Alegre: ArtMed, p. 156–182, 1998. Citado na página 26.
- CARDOSO, A.; COSTA, E.; SOUSA, G. *O Potencial Educativo do Raspberry Pi*. [S.l.]: 16f, 2013. Citado 2 vezes nas páginas 20 e 21.
- CRUZ, J. M. d. O. The teaching-learning process in the information society. *Educação & Sociedade*, SciELO Brasil, v. 29, n. 105, p. 1023–1042, 2008. Citado na página 44.
- FURIA, F. *Crianças que programam: uma habilidade para toda a vida*. 2014. Disponível em: <<http://www.playground-inovacao.com.br/criancas-que-programam-uma-habilidade-para-toda-a-vida/>>. Acesso em: 26 agosto 2017. Citado na página 13.
- GOMES, R. C. S.; GHEDIN, E. O desenvolvimento cognitivo na visão de jean piaget e suas implicações a educação científica. *Encontro Nacional de Pesquisa Em Educação Em Ciências*, v. 8, p. 29–39, 2011. Citado na página 26.
- HEINEN, E. et al. Raspiblocos: Ambiente de programação didático baseado em raspberry pi e blockly. In: *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*. [S.l.: s.n.], 2015. v. 26, n. 1, p. 567. Citado 2 vezes nas páginas 27 e 44.
- JÚNIOR, J. et al. Ensino de algoritmos e programação: uma experiência no nível médio. In: *XIII Workshop de Educação em Computação (WEI'2005)*. São Leopoldo, RS, Brasil. [S.l.: s.n.], 2005. Citado na página 14.
- JUNIOR, M. J. B. Ensinando programação de computadores nas escolas: a proposta do code. or. *Revista Educacional Interdisciplinar*, v. 2, n. 1, 2015. Citado na página 13.
- KELLEHER, C.; PAUSCH, R. Lowering the barriers to programming: A taxonomy of programming environments and languages for novice programmers. *ACM Computing Surveys (CSUR)*, ACM, v. 37, n. 2, p. 83–137, 2005. Citado na página 17.

- MALTEMPI, M. V. Construcionismo: pano de fundo para pesquisas em informática aplicada à educação matemática. *Educação matemática: pesquisa em movimento*. São Paulo: Cortez, p. 264–282, 2004. Citado na página 16.
- MILETTO, E. M.; BERTAGNOLLI, S. de C. *Desenvolvimento de Software II: Introdução ao Desenvolvimento Web com HTML, CSS, JavaScript e PHP-Eixo: Informação e Comunicação-Série Tekne*. [S.l.]: Bookman Editora, 2014. Citado 2 vezes nas páginas 23 e 24.
- MULLER, N. Framework, o que é e para que serve. *Documento online*: < http://www.oficinadanet.com.br/artigo/1294/framework_o_que_e_e_para_que_serv_e > Acessado em, v. 10, p. 18, 2014. Citado na página 25.
- PAPERT, S. *LOGO: Computadore e Educação*. [S.l.]: Brasiliense s.a., 1985. Citado 3 vezes nas páginas 14, 16 e 44.
- PAPERT, S. *A máquina das crianças: repensando a escola na era da informática*. [S.l.: s.n.], 1994. Citado na página 19.
- RICHARDSON, M.; WALLACE, S. Primeiros passos com o raspberry pi. *Primeira Edição*. Novatec Editora Ltda, 2013. Citado 2 vezes nas páginas 20 e 21.
- RODRIGUES, W. d. S. Atividades com robótica educacional para as aulas de matemática do 6. ao 9. ano do ensino fundamental: utilização da metodologia lego® zoom education. Universidade Estadual Paulista (UNESP), 2014. Citado na página 19.
- SALEIRO, M. et al. A low-cost classroom-oriented educational robotics system. In: SPRINGER. *International Conference on Social Robotics*. [S.l.], 2013. p. 74–83. Citado na página 27.
- SILVA, A. F.; GUERREIRO, A. M. G.; GONÇALVES, L. M. G. *RoboEduc: Uma Metodologia de Aprendizado com Robótica Educacional*. Tese (Doutorado), 2009. Citado 2 vezes nas páginas 18 e 19.
- SUDOL, L. A. *Visual Programming Pedagogies and Integrating Current Visual Programming Language Features*. Tese (Doutorado) — Robotics Institute, 2009. Citado na página 17.
- ZILLI, S. d. R. et al. A robótica educacional no ensino fundamental: perspectivas e prática. Florianópolis, SC, 2004. Citado na página 19.
- ZILLI, S. R. Apostila de robótica educacional. *Expoente Informática-Gráfica Expoente*, 2002. Citado na página 19.