

## **ROBOLT-0: UMA PLATAFORMA ROBÓTICA MULTI-APLICAÇÃO, EXTENSÍVEL, OPEN-SOURCE E DE BAIXO CUSTO**

**A. F. Sousa<sup>1</sup>, R. R. M. de Santana<sup>2</sup>, J. P. C. M. Oliveira<sup>3</sup>, R. C. Sá<sup>4</sup>**

<sup>1</sup>Discente de graduação em Engenharia de Computação - IFCE. Pesquisadora LIT/IFCE. e-mail: alyne@lit.ifce.edu.br; <sup>2</sup>Discente de graduação em Engenharia de Computação - IFCE. Pesquisadora LIT/IFCE. e-mail: ryllari.marques@lit.ifce.edu.br; <sup>3</sup>Discente de graduação em Engenharia de Computação - IFCE. Pesquisador LIT/IFCE. e-mail: paolo@lit.ifce.edu.br; <sup>4</sup>Professora de graduação em Engenharia Mecatrônica - IFCE. e-mail: rejane@lit.ifce.edu.br

1 **RESUMO:** **Robolt-0** é uma plataforma robótica projetada para ser usada em diferentes  
2 aplicações, a principal delas: o ensino de robótica e programação através da montagem do  
3 hardware e criação de aplicativos de controle e monitoramento do robô. Pode ser empregado,  
4 ainda, em testes de inteligência artificial, facilitando a coleta e análise de dados ao prover uma  
5 arquitetura genérica e bem definida. Ele é extensível, permitindo que novos componentes  
6 possam ser adicionados facilmente, e open-source, possibilitando sua evolução e facilitando a  
7 padronização, além do intercâmbio de aplicações. Neste artigo, descrevemos seu desenho,  
8 concepção e testes, além da abordagem da escolha de materiais de baixo custo, propiciando  
9 alta reprodutibilidade.

10 **Palavras-chave:** Robótica, Programação, Educação.

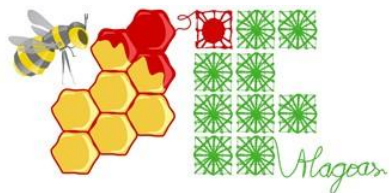
### **ROBOLT-0: A MULTI-APPLICATION, EXTENSIBLE, OPEN-SOURCE AND LOW COST ROBOTIC PLATFORM**

15 **ABSTRACT:** **Robolt-0** is a robotic platform designed to be used in different applications, the  
16 main one: education for robotics and programming by assembling hardware and building  
17 applications of control and monitoring in the robot. It can be used, also, in artificial  
18 intelligence tests, facilitating the collection and analysis of data by providing a generic and  
19 well-defined architecture. It is extensible, allowing new components to be added easily, and  
20 open-source, allowing its evolution and facilitating standardization, besides the exchange of  
21 applications. On this work, we describe the design, the concept and the tests performed other  
22 than the approach of choice of low cost materials, allowing high reproducibility.

23 **KEYWORDS:** Robotics, Programming, Education.

## **INTRODUÇÃO**

26 Nas últimas décadas, em decorrência do avanço tecnológico, pesquisas e estudos  
27 relacionados a robótica têm sido o alvo de muitos pesquisadores, dentro e fora do ambiente  
28 acadêmico. Conhecer a fundo as possibilidades e funcionalidades que uma plataforma  
29 robótica pode gerar, tem sido um meio de abrir os horizontes para a aplicação da tecnologia  
30 em áreas diversas e amplas no cotidiano humano.



31 Muitas plataformas robóticas genéricas existem (PIO *et al.*, 2006), porém são, em  
32 geral, caras, o que dificulta seu uso em muitos cenários, principalmente no ensino e pesquisa  
33 (SCHILLING *et al.*, 2002), entretanto, a adaptabilidade e reconfiguração em tempo de  
34 execução são uma ferramenta importante para exploração de nuances mais profundas da  
35 robótica, de modo a existir uma necessidade de uma plataforma aberta, extensível e barata.  
36 Aberta para propiciar melhorias, extensível para poder agir em qualquer cenário, desde  
37 acompanhamento de idosos (MICHAUD *et al.*, 2007) ou pacientes em geral ao observar suas  
38 leituras biológicas e reações, até monitoramento de ambientes perigosos e educação.

39 Dentro desta realidade, o presente trabalho tem como objetivo descrever e detalhar o  
40 desenvolvimento de uma plataforma robótica multi-aplicação, o **robolt-0**, que foi desenhado  
41 levando em consideração o baixo custo na montagem, alta performance e diversidade quanto  
42 a sua aplicação, bem como a possibilidade de ser facilmente estendido e replicável.

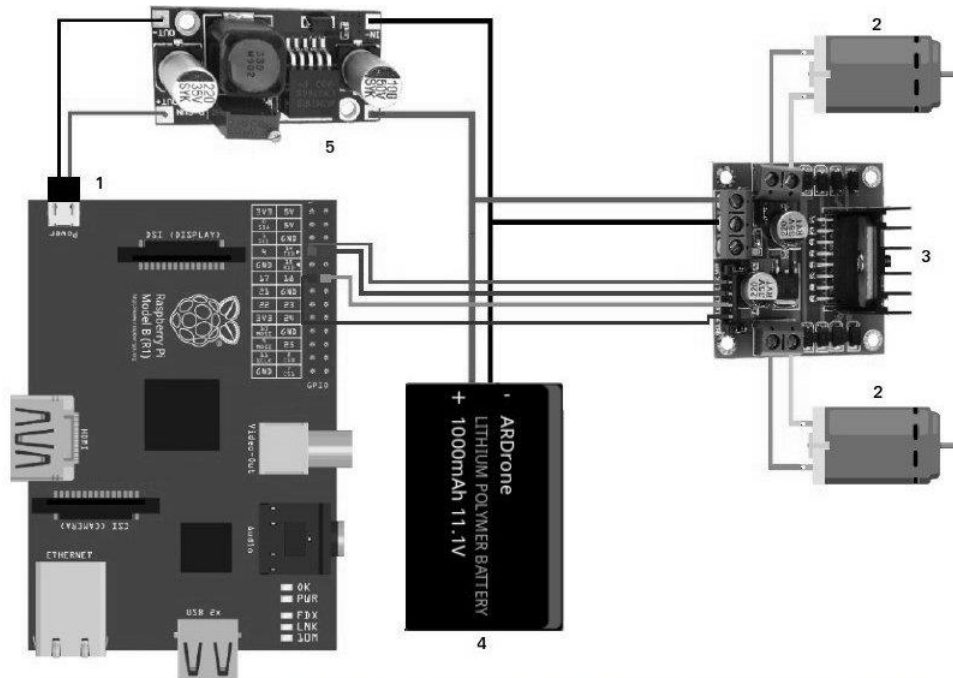
43

## 44 MATERIAL E MÉTODOS

### 45 Especificação do hardware

46 De forma geral, a montagem foi realizada de acordo com o esquemático presente na  
47 figura 1.

48



1 - Raspberry Pi Model B, 2 - Motores, 3 - L298N, 4 - Bateria de polimero de litio, 5 - LM2596

49

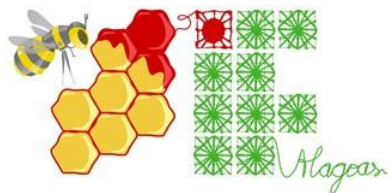
50 **Figura 1.** Componentes estruturais do protótipo. IFCE, 2016

51

52 Um Raspberry Pi Model B, baseado no SoC(System-on-Chip) Broadcom BCM2835,  
53 que contém um ARM1176JZFS (ARM11 usando um núcleo com arquitetura ARMv6) com  
54 ponto flutuante, rodando a 700Mhz e uma GPU Videocore 4, é o núcleo estrutural do robô. É  
55 ele quem recebe e processa os sinais de sensores, bem como, envia os sinais de controle para  
56 os motores. O sistema operacional utilizado é o GNU/Linux Raspian 7(wheezy) kernel  
57 4.1.13+.

58 Um case genérico para Raspberry foi utilizada para facilitar a montagem, visto que  
59 possui as dimensões ideais, tanto para acomodação do raspberry Pi, sua função original,  
60 quanto para o encaixe das caixas de redução dos motores. Além disso, reduz os custos  
61 estruturais, pois não há necessidade de criação de um suporte.

62 Os motores com caixa de redução foram alocados paralelamente sob a case do  
63 Raspberry Pi e fixados com fita dupla-face. Os motores estão levemente fora da case para  
64 evitar contato, evitando assim aquecimento e desgastes dos mesmos, além de aumentar a  
65 estabilidade do robô.



66 Um driver de motor DC com ponte H, L298N, é responsável pela mudança de sentido  
67 da corrente que atua nos motores, possibilitando diferentes movimentações. O controle é feito  
68 através dos pinos de propósito geral (GPIO) do Raspberry Pi.

69 A roda omnidirecional foi posicionada à frente do suporte, entre os dois pneus, com a  
70 finalidade de equilibrar o robô e permitir uma melhor movimentação.

71 A bateria escolhida, de polímero de lítio 11.1V, é o padrão usado habitualmente em  
72 VANTs. Foi escolhida tanto por suportar os dois motores, a ponte H e o Raspberry Pi, quanto  
73 por ser facilmente encontrada em lojas de aeromodelismo e relativamente barata. Além disso,  
74 é bem compacta, cabendo e podendo ser presa ao case usado como base.

75 Um regulador de tensão, LM2596, é necessário para reduzir a tensão de 11.1v da bateria  
76 para 5v, propiciando a alimentação do Raspberry Pi com a mesma fonte dos motores.

77

## 78 Modelagem

79 O protótipo foi modelado usando a ferramenta open-source Blender, destinada a  
80 originalmente à criação de jogos, de forma que modelar e simular se tornasse uma tarefa mais  
81 simples.

82 No blender, os componentes do modelo foram descritos e renderizados através de uma  
83 interface de script, anexa originalmente na aplicação, utilizando a linguagem de programação  
84 python e uma biblioteca, bpy, que faz o binding das funções internas.

85 Na Figura 2 é possível notar o uso da biblioteca bpy em parte do script.

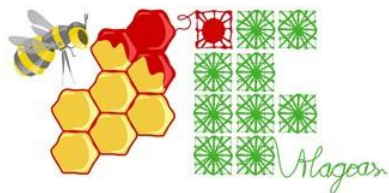
86

```
62 def girar():  
63     bpy.ops.object.select_all(action='SELECT')  
64     bpy.data.objects['Camera'].select_set(True)  
65     bpy.data.objects['Lamp'].select_set(True)  
66     menos_pi = math.pi/12  
67     bpy.ops.transform.rotate(value=(menos_pi, menos_pi, menos_pi))  
68  
69 if __name__ == "__main__":  
70     rodinhas()  
71     motor()  
72     base()
```

87

88 **Figura 2.** Trecho do código em python. IFCE, 2016.

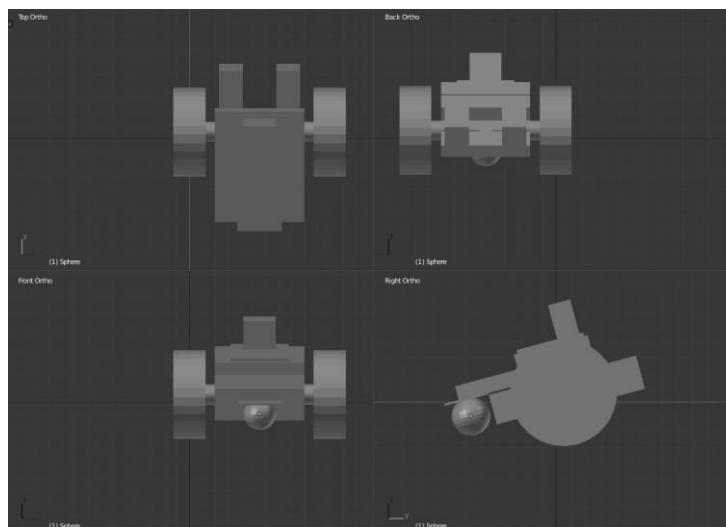
89



90           Devido ao fato de ser uma plataforma robótica multi-aplicação, a estrutura do protótipo  
91 foi prototipada sobre uma matriz de inércia desejada, porém proporcionando um design  
92 elegante e atrativo, além de um melhor desempenho (velocidade).

93           A modelagem gráfica da estrutura resultou em um protótipo de 00cm de altura, 00cm de  
94 comprimento, 00cm de largura e massa de 00g. A Figura 3 mostra o resultado no Blender do  
95 código em python que gerou o modelo do protótipo.

96



97

98           **Figura 3.** Modelo final gerado no Blender. IFCE, 2016.

99

### 100 **Especificação e estrutura do software**

101           Para gerenciar e facilitar o desenvolvimento de novas aplicações para plataforma, foi  
102 desenvolvido o  $\Sigma$ -r (sigma-r - System InteGrator and Message Architect for robotics), um  
103 middleware integrável a sistemas POSIX, onde, através da abstração das camadas de mais  
104 baixo nível, do uso de uma arquitetura de enfileiramento de mensagem baseada no padrão  
105 publicadores/inscritos e exposição de funcionalidades através de API, provê tanto a facilitação  
106 de adição de novos hardwares, como a leitura de dados dos sensores. Além disto, templates  
107 de novas aplicações, sensores e hardwares quaisquer, bem como as ferramentas para gerar e  
108 testá-los, são parte integral do middleware.

109           O núcleo do middleware é uma camada de abstração para as chamadas de comunicação  
110 inter-objetos, para o gerenciamento de novos hardwares acoplados e fornece uma API para  
111 aplicações externas.

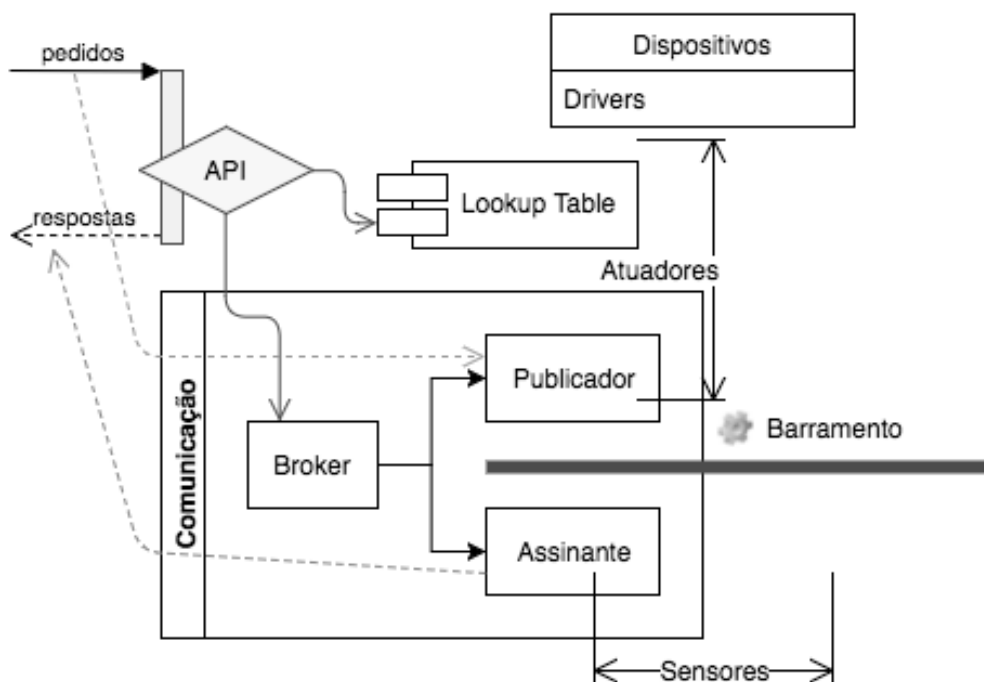
112 Em paralelo, há um broker, recebendo e armazenando todas as mensagens advindas de  
113 sensores e transmitindo todas as mensagens de controles para os atuadores, além de possuir  
114 uma Lookup Table relacionando todos os elementos do sistema e indexando-os com UUID.

115 Toda comunicação e troca de mensagens interna é feita através de um modelo  
116 publicador/inscrito disponibilizado pelo ZeroMQ, um pseudo enfileirador de mensagem, leve,  
117 multi plataformas e disponível em múltiplas linguagens de programação.

118 A camada de abstração relacionada à gestão de todo hardware tenta usar, inicialmente, o  
119 conceito de dispositivos tratados como arquivos, dos sistemas POSIX de forma a inicializar o  
120 IO do dispositivo através de syscalls ao kernel do sistema. Em último caso, usa-se a Interface  
121 de Função Estrangeira (FFI) do python para se conectar a um código nativo, geralmente em C,  
122 compilado como uma biblioteca compartilhada. Todo este processo é transparente ao  
123 programador, sendo relevante apenas ao desenvolvedor do dispositivo e seu driver, pois uma  
124 vez conectado ao sistema, seu uso se baseia em se inscrever no seu canal e ler ou escrever  
125 comandos.

126 Um diagrama estrutural do sistema pode ser visto na imagem 4.

127



128

129 **Figura 4.** Diagrama estrutural do Σ-r. IFCE, 2016.

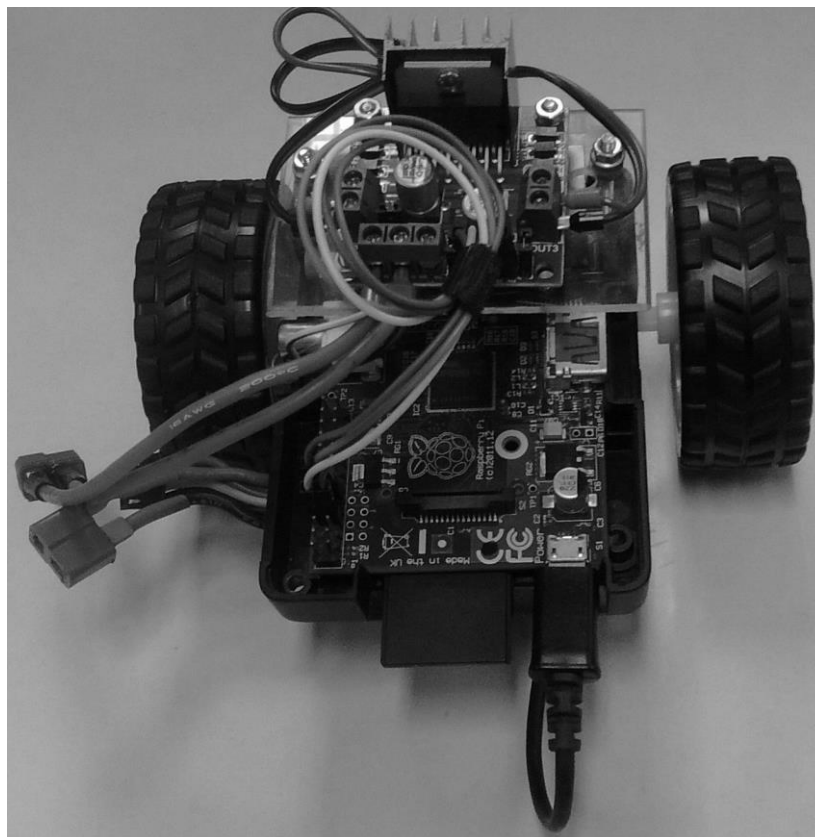
130

131

## 132 RESULTADOS E DISCUSSÃO

133 Estruturalmente, os requisitos operacionais definidos na modelagem foram obtidos  
134 depois da montagem, o que nos proporcionou uma garantia de estabilidade e integridade do  
135 robô. Nas figuras 5, 6 e 7 vemos a montagem final, respectivamente na visão de frente, lateral  
136 e costas.

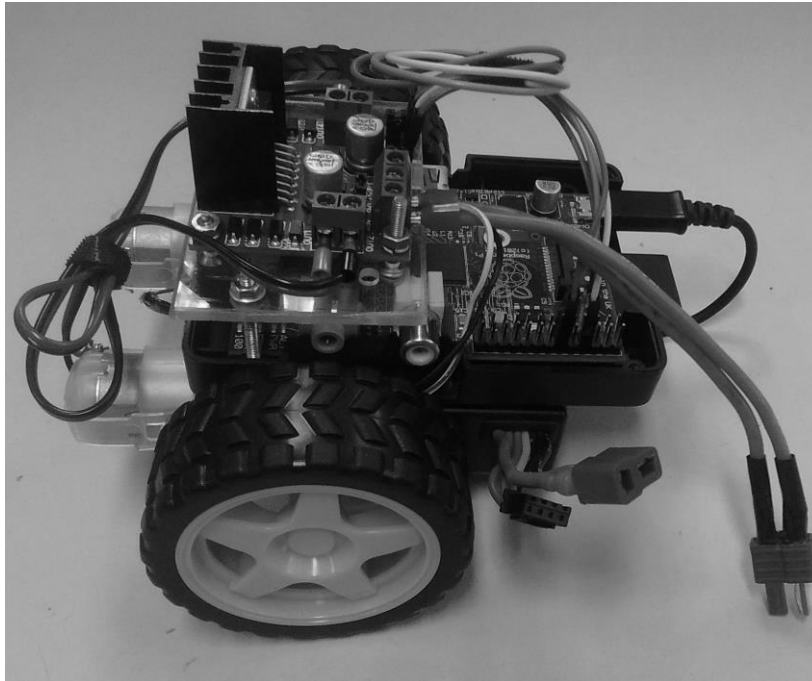
137



138

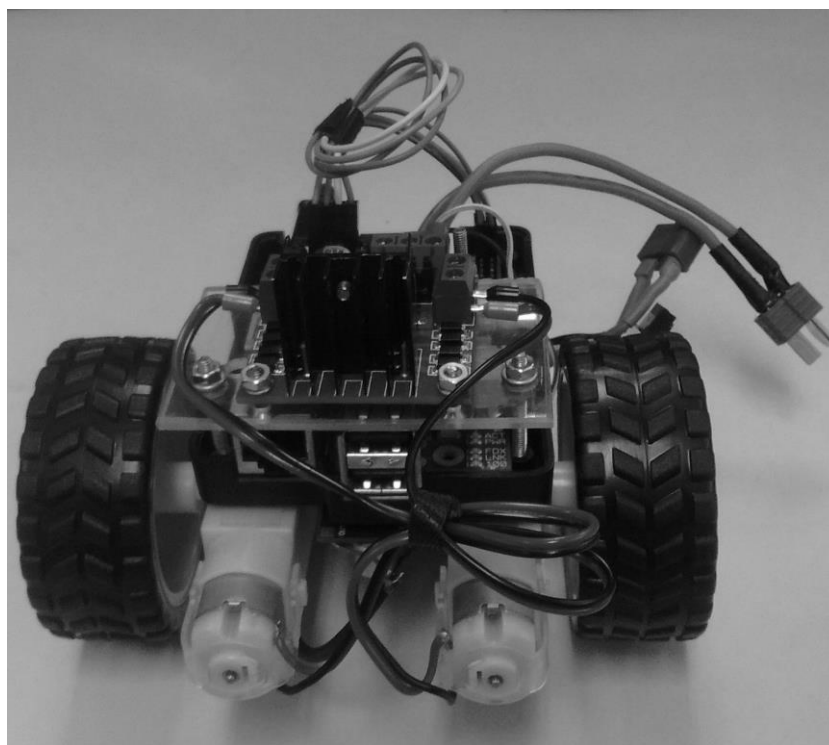
139 **Figura 5.** Visão frontal. IFCE, 2016.

140



141  
142  
143

**Figura 6.** Visão lateral. IFCE, 2016.

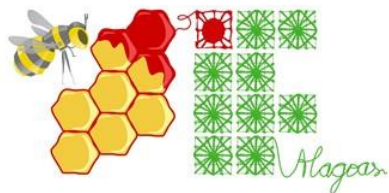


144  
145  
146

**Figura 7.** Visão de trás. IFCE, 2016.

147 Em paralelo, a plataforma está em testes em laboratórios e escolas de programação para  
148 crianças e, até o presente momento, tem se mostrado uma excelente ferramenta tanto para o





149 ensino de programação básica, quanto para robótica e eletrônica. Além disto, a plataforma  
150 está sendo utilizada também, com sucesso, na validação de testes de aplicações de  
151 reconhecimento e planejamento de caminhos em um ambiente de laboratório.

152

## 153 **CONCLUSÕES**

154 A plataforma foi montada, está funcional, e todos os seus detalhes serão  
155 disponibilizados para ser replicada em qualquer escola, faculdade ou laboratório em breve.  
156 Embora mais testes sejam necessários, pode-se observar o auxílio da plataforma no ensino de  
157 programação e robótica, principalmente em crianças, onde uma abordagem lúdica aliada ao  
158 resultado de ações serem visíveis instantaneamente propiciarem um constante interesse,  
159 mesmo em idades menores, à ciência e tecnologia, além de instigar a curiosidade.

160

## 161 **REFERÊNCIAS**

162 BLENDER 2.78.0 9d70344: API documentation. Blender Foundation. Disponível em:  
163 <[http://www.blender.org/api/blender\\_python\\_api\\_2\\_78\\_release/](http://www.blender.org/api/blender_python_api_2_78_release/)>. Acesso em: 1 set. 2016.

164

165 JOSEPH, L. **Learning Robotics Using Python**: Design, simulate, program, and prototype an  
166 interactive autonomous mobile robot from scratch with the help of Python, ROS, and Open-  
167 CV!. 1. ed. Birmingham: Packt Publishing, 2015. 303p.

168

169 PIO, J; CASTRO, T; JÚNIOR, A. A Robótica Móvel como Instrumento de Apoio à  
170 Aprendizagem de Computação. **SBIE**, Brasília, DF, 2008.

171

172 SCHILLING, K; ROTH, H; RÖSCH, O. Mobile Mini-Robots for Engineering Education.  
173 **Global J. of Engng. Educ.**, v.6, n. 1, Australia, 2002.

174

175 MICHAUD, F et al. Telepresence Robot for Home Care Assistance. **AAAI Spring**  
176 **Symposium: Multidisciplinary Collaboration for Socially Assistive Robotics**. 2007. p. 50-  
177 55.