

# EDUBOT: PROPOSTA DE UM ROBÔ MÓVEL NÃO-HOLONÔMICO COM ACIONAMENTO DIFERENCIAL INTEGRADO COM FRAMEWORK SOA

CARLOS SOLON SOARES GUIMARÃES JUNIOR\*, EDUARDO HENRIQUE MACIEL\*, RENATO VENTURA BAYAN HENRIQUES\*, CARLOS EDUARDO PEREIRA\*

\*Av. Osvaldo Aranha, 103 - Bairro Bom Fim  
Universidade Federal do Rio Grande do Sul  
Porto Alegre, Rio Grande do Sul, Brasil

Emails: carlos.solon@ufrgs.br, eduardo.maciell@ufrgs.br, rventura@ece.ufrgs.br, cpereira@ece.ufrgs.br

**Abstract**— In the current educational context, the use of robotics will greatly enhance creativity and personal development, promoting integration between different areas of knowledge, for this area of engineering is seen as multidisciplinary. The study of robots can have an experience in practice, being in a developing group or alone. Typically studies in this area using known scientific methods, mechanical simulations, implementations that involve the use of software and hardware together and prototyping. This article proposes the development of a mobile robotic platform with integrated non-holonomic differential drive, called EduBot (Educational Robotics). This platform seeks to integrate the various components of software and hardware that make up the development of an educational mobile robotic platform as well as the integration of a framework based on Service Oriented Architecture (SOA), which will provide the libraries and components for the various existing platforms on the market today. The study EduBot platform, has been widely applied and validated with the students of the classes of Electrical Engineering, Computer Engineering, Control Engineering and Automation of the Federal University of Rio Grande do Sul.

**Keywords**— Mobile robot, EduBot Platform, Framework, SOA.

**Resumo**— No contexto educacional, a utilização da robótica pode ampliar significativamente a criatividade e o desenvolvimento intelectual, promovendo a integração entre diferentes áreas de conhecimento. A robótica tem, em tal contexto, um grande potencial como ferramenta interdisciplinar, religando as várias disciplinas. Esta proposta de trabalho propõe o desenvolvimento de uma plataforma robótica móvel não-holonômica com acionamento diferencial, denominada EduBot (*Educational Robotics*). Esta plataforma busca integrar os diversos componentes de software e hardware, que compõem o desenvolvimento de uma plataforma robótica móvel educacional, bem como a integrações *frameworks*, que fornecem pacotes de bibliotecas e componentes. A plataforma proposta EduBot, vem sendo desenvolvido, aplicado e validado com os discentes das turmas de Engenharia da Computação da Universidade Federal do Rio Grande do Sul.

**Palavras-chave**— Robô Móvel, Plataforma EduBot, Framework, SOA.

## 1 Introdução

Os conceitos de robótica, podem se relacionar com questões energéticas, capacidade de perceber, modelar, planejar e atuar para alcançar determinados objetivos, sem a intervenção (ou com uma intervenção muito pequena) do operador humano, já que o robô pode se locomover em ambientes estruturados ou não estruturados, total ou parcialmente conhecidos. O papel deste deve ser desempenhado pelo próprio sistema de controle do veículo, que o deve suprir com a inteligência necessária para fazê-lo mover corretamente.

O robô móvel escolhido para o projeto é o robô com acionamento diferencial ou uniciclo, como também é conhecido no estudo da arte de robôs modelados com acionamento diferencial, em geral, o eleito por pesquisadores para experimentar novas estratégias de controle por possuir uma cinemática simples. É uma estrutura formada por duas rodas fixas convencionais, sobre um mesmo eixo, controladas de maneira independente, e por uma roda passiva que lhe confere estabilidade. O sistema de tração-direção associado ao robô lhe permite controlar de forma independente suas ve-

locidades linear e angular. As vantagens que derivam de sua estrutura mecânica e da eletrônica de controle fazem desta configuração a preferida para robôs de laboratório (Siegwart, 2004).

A parte do software do projeto se preocupa com o desenvolvimento do *firmware* e integração dos módulos do projeto com um *framework* SOA, para disponibilizar serviços para simular, controlar e interagir com o robô, sempre levando em consideração o baixo acoplamento dos componentes de software e hardware e a simplicidade de estender as funcionalidades existentes, para construir uma ferramenta educacional para uso na robótica (da Silva, 2009).

Na seção 2 são apresentados alguns dos principais conceitos do projeto sobre o desenvolvimento mecânico, eletrônico e computação da plataforma de *hardware* e *software*. A seção 3 apresenta a implementação e resultados do projeto e por fim, a seção 4 são feitas as considerações finais sobre o trabalho desenvolvido e sugestões para trabalhos futuros.

## 2 Desenvolvimento do Projeto

### 2.1 Projeto Mecânico

O projeto mecânico da plataforma de hardware Edubot atualmente está na versão 03 (Guimarães, 2014). Na Figura 1 o modelo 3D da plataforma oferece suporte a diversos *hardwares*. A montagem do modelo e seus componentes, são projetados e desenhados em *software* CAD de maneira modular, podendo-se tornar um projeto simples ou complexo dependendo dos requisitos do projeto.

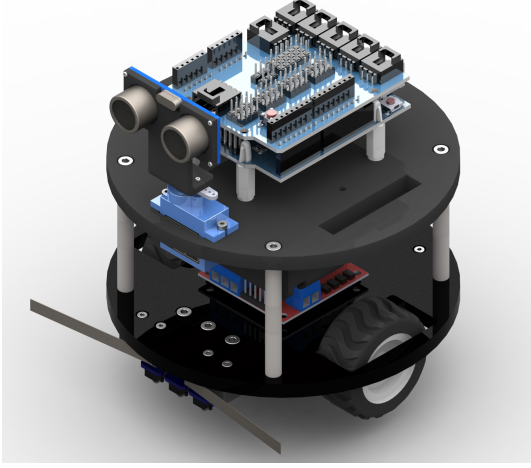


Figura 1: EduBot versão 03, desenhado por Eduardo Henrique Maciel.

O modelo cinemático do robô móvel do tipo unicyclo está baseado nas seguintes considerações: o robô é constituído por um chassi rígido e rodas não-deformáveis, que se movem em um plano horizontal. Assume-se que o plano das rodas permanece na vertical durante a movimentação, com a rotação se dando em torno de um eixo horizontal cuja orientação é fixa ou variável com relação a um sistema de coordenadas associado ao corpo do robô,  $\{X_c, Y_c, \theta_c\}$ , descrito em relação ao sistema inercial  $\{X_0, Y_0, \theta_0\}$ , com  $\theta$  sendo o ângulo que define a orientação do robô. A partir da fig:robot01 pode-se definir o vetor  ${}^0\xi_c$  que contém as coordenadas da pose do robô e a matriz  ${}^0R_c$  que expressa a orientação do robô em relação ao sistema inercial  $\{X_0, Y_0, \theta_0\}$ .

$${}^0\xi_c = \begin{bmatrix} X_c \\ Y_c \\ \theta_c \end{bmatrix} \quad (1)$$

$${}^0R_c \xi_c = \begin{bmatrix} \cos(\theta_c) & -\sin(\theta_c) & 0 \\ \sin(\theta_c) & \cos(\theta_c) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2)$$

Ainda, a velocidade do robô descrita em relação ao sistema inercial pode ser dada por:

$${}^0\dot{\xi}_c = \begin{bmatrix} \dot{X}_c \\ \dot{Y}_c \\ \dot{\theta}_c \end{bmatrix} \quad (3)$$

e descrita no sistema  $\{X_0, Y_0, \theta_0\}$  por  ${}^c\dot{\xi}_c = {}^cR_0 {}^0\dot{\xi}_c$ , com  ${}^cR_0$  dado por:

$${}^cR_0 = {}^0R_c^{-1} = {}^0R_c^T = \begin{bmatrix} \cos(\theta_c) & \sin(\theta_c) & 0 \\ -\sin(\theta_c) & \cos(\theta_c) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4)$$

Considera-se que o contato entre as superfícies das rodas e do piso satisfaz as condições de rotação pura, isto é, uma velocidade constante de rotação das rodas equivale a uma velocidade constante de translação das mesmas. A Figura 2 mostra uma representação esquemática detalhada do robô em estudo.

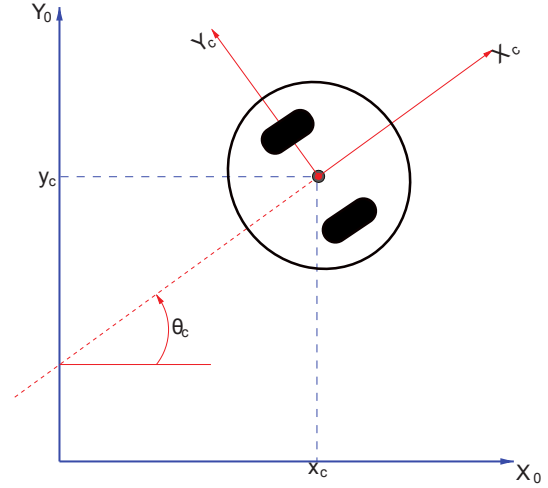


Figura 2: Esquema para obtenção do modelo cinemático do robô móvel unicyclo. Fonte: (Siegwart, 2004)

Além disso, em cada instante de tempo, o movimento do robô pode ser visto como uma rotação instantânea ao redor do seu Centro Instantâneo de Rotação (CIR), que no caso do robô móvel unicyclo encontra-se no eixo central que conecta as suas rodas, conforme mostra a Figura 3. Na figura,  $v_e$  e  $v_d$  representam, respectivamente, as velocidades das rodas esquerda e direita, ambas em relação ao solo, e  $R$  é o raio de curvatura instantâneo da trajetória do robô, isto é, a distância do CIR ao ponto central do eixo virtual que conecta as suas rodas (Lages, 1998); (Barros, 2014).

Desse modo, se  $v_e = v_d$ , então o raio  $R$  é infinito e o robô move-se em linha reta. Por outro lado, se  $v_e = -v_d$ , então o raio  $R$  é zero e o robô gira ao redor do ponto central do eixo virtual que conecta as suas rodas. Para qualquer outro valor de  $v_e$  e  $v_d$  o robô move-se em linha curva, ao redor do seu Centro Instantâneo de Rotação, o qual

está situado a uma distância  $R$  do ponto central do eixo virtual que conecta as suas rodas, alterando tanto a sua posição quanto a sua orientação.

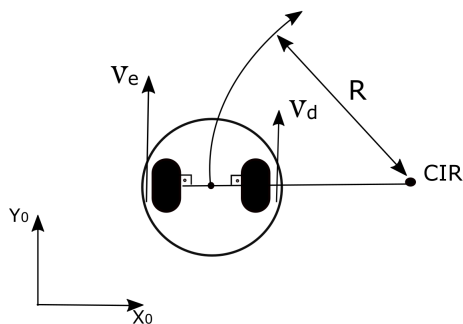


Figura 3: Representação do CIR do robô móvel uniciclo. Fonte: (Siegwart, 2004)

O modelo cinemático de postura do robô uniciclo, em notação matricial, é dado por

$$\begin{bmatrix} \dot{X}_c \\ \dot{Y}_c \\ \dot{\theta}_c \end{bmatrix} = \begin{bmatrix} \cos(\theta_c) & -\sin(\theta_c) \\ \sin(\theta_c) & \cos(\theta_c) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ \omega \end{bmatrix}, \quad (5)$$

onde a velocidade linear  $u$  e a velocidade angular  $\omega$  são as entradas de controle. Cabe mencionar, no entanto, que em robôs reais, sinais de entrada desse tipo não podem ser aplicados aos robôs diretamente, uma vez que o acionamento dos mesmos é feito por motores elétricos. Em virtude disso, as velocidades são aplicadas indiretamente, através de um controle de velocidade de baixo nível. Além disso, é possível considerar que as velocidades  $u$  e  $\omega$  são desenvolvidas instantaneamente, desde que as mesmas sejam baixas e a inércia do robô seja pequena (Martins, 2009).

Por fim, tem-se que o robô móvel uniciclo é controlável. Isto significa que ele sempre pode ser conduzido de uma postura inicial  $\xi_0$  a uma postura final  $\xi_f$  em um tempo finito, por meio da manipulação dos seus sinais de entrada (Siegwart, 2004).

## 2.2 Projeto Eletrônico

Para uma melhor descrição do sistema foi desenvolvido um diagrama de blocos, Figura 4, que é um diagrama cujo objetivo é a representação gráfica do processo e modelo do sistema. Através de figuras geométricas e ligações, descrevem-se as relações entre cada subsistema e o fluxo de informação. O protótipo do *hardware* do sistema pode ser composto de atuadores, módulos e sensores, que estão sendo configurados conforme as relações geométricas do robô para determinar a melhor adaptação dos componentes. O tratamento das informações captadas e transmitidas pelos módulos e sensores é processado através de um microcontrolador, dotado de um algoritmo para ativar os atuadores. Os módulos e sensores estimulam

os atuadores conforme a lógica de programação e controle do projeto. O protótipo do *hardware* é baseado em plataformas flexíveis *open-hardware* para o desenvolvimento, a reutilização dos circuitos proporcionam a criação de novas plataformas para sistemas embarcados.

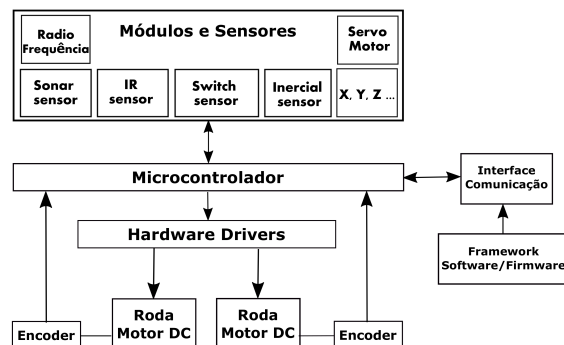


Figura 4: Diagrama de Blocos para o *hardware* do Robô Uniciclo. Fonte: Autor.

Neste trabalho, para testes de validação dos componentes de *hardware*, esta sendo utilizado o microcontrolador Atmega328P. O microcontrolador é conectado com um *driver* ponte-h para o acionamento dos motores. A ponte-h utiliza duas pontes completas independentes (*full-bridge drive*), cujo objetivo é ativar os motores de Corrente Contínua (CC), uma vez que no projeto são usados motores de 9V e a saída do microcontrolador fornece um valor máximo de 5V *Transistor-Transistor Logic* (TTL). O objetivo é criar uma plataforma de *hardware* como uma opção de baixo custo, com o objetivo de controlar os sentidos de rotação dos motores e, por consequência, o sentido da corrente que circula entre os polos dos motores, visto que os motores CC alteram seus sentidos de rotação quando inverte-se sua polaridade. Na Figura 1 pode-se observar a imagem em CAD 3D da ponte-h e a plataforma Arduino inicialmente utilizadas no protótipo. O *driver* é controlado por três entradas para cada motor, onde a primeira entrada é de habilitação (EnA) que ativa ou desativa a ponte-h e as outras duas (In1 e In2) determinam a circulação interna da corrente. A escolha dos componentes se deu por uma combinação de fatores: baixo preço, tempo de desenvolvimento, facilidade de encontrar no mercado e programação do microcontrolador em três níveis (AVR *Assembler*, AVR GCC e *Wiring C/C++*). A comunicação entre a interface de controle do *framework* SOA e o dispositivo se dá através de uma conexão serial padrão. O ATmega328P permite comunicação serial no padrão universal *Asynchronous Receiver/Transmitter* (UART) TTL (5 V), que esta disponível nos pinos digitais 0 (RX) e 1 (TX). Um conversor Universal Serial Bus (USB)-TTL é responsável pela comunicação serial com o *chip* Atmega8U2 que encaminha esta comunicação serial

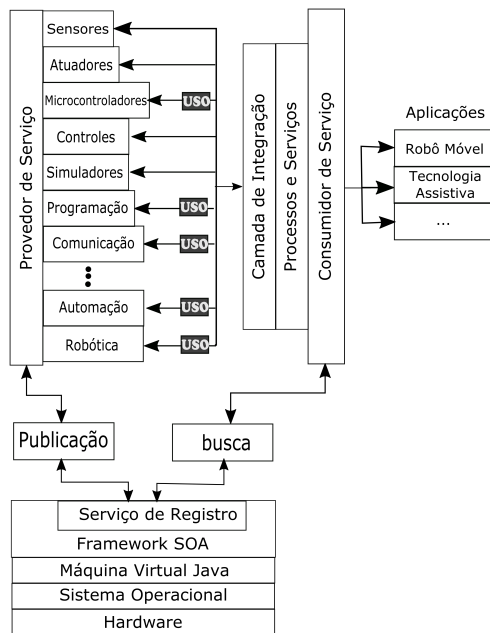


Figura 5: Arquitetura para o *Framework SOA* Proposto. Fonte: Autor.

através da USB fornecendo uma porta virtual para o *software* no computador (Arduino, 2005).

### 2.3 Projeto *Framework SOA*

Uma arquitetura orientada a serviços é um modelo para construção de soluções de *software* que utiliza como seu principal elemento unidades de desenvolvimento denominadas serviços, que são elementos auto-descritos, agnósticos de plataforma, que executam funções e que podem variar desde simples requisições até processos de negócio complexos. É uma solução utilizada na integração de sistemas e na comunicação entre aplicações diferentes. Com esta tecnologia é possível que novas aplicações possam interagir com aquelas que já existem e que sistemas desenvolvidos em plataformas diferentes sejam compatíveis. Os serviços são componentes que permitem às aplicações enviar e receber dados. A arquitetura SOA é constituída por três componentes básicos: o servidor de registro, o provedor de serviços e o consumidor de serviços. As interações entre esses componentes são de busca, publicação e interação de operações. A Figura 5 mostra a arquitetura de camadas proposta para o desenvolvimento do *framework SOA*.

O provedor de serviços representa a camada que hospeda o serviço permitindo que os clientes acessem o serviço. O provedor de serviços fornece o serviço e é responsável por publicar a descrição do serviço que provê. O solicitante ou consumidor de serviços é a aplicação que está procurando, invocando uma interação com o serviço, ou seja, requisita a execução de um serviço. O consumidor de serviço pode ser uma pessoa ou organização. O

servidor de registro é um repositório central que contém a descrição (informação) de um serviço, e é por meio do servidor de registro que essas descrições são publicadas e disponibilizadas para localização. Os consumidores buscam por serviços no servidor de registro e recuperam informações referentes à interface de comunicação para os serviços durante a fase de desenvolvimento ou durante a execução do cliente, denominadas interação estática (*static bind*) e interação dinâmica (*dynamic bind*), respectivamente. Na interação estática, o cliente recupera a assinatura do serviço, necessária à codificação. Na interação dinâmica, o cliente recupera os valores de parâmetros e a localização do serviço (Fugita, 2009).

## 3 Implementações e Resultados

O diagrama de implantação modela o interrelacionamento entre recursos de infra-estrutura, de rede ou artefatos de sistemas. Normalmente apresentamos servidores neste diagrama. Estes recursos são chamados de *nodes* ou nós. Cada nó é um máquina física que contém um ou vários componentes. Outros dispositivos podem ser representados com os estereótipos *processor* ou *device*. Foi desenvolvido o diagrama de implantação de integração do *framework SOA* com a plataforma do robô móvel EduBot, Figura 6.

Para desenvolvimento do *software* foi definida a utilização da linguagem de programação Java para o desenvolvimento do *framework SOA*. Java é o termo geral usado para denotar o *software* e seus componentes, que incluem Java *Runtime Environment* (JRE), Java *Virtual Machine* (JVM) e também *Plug-in*. Para fazer a integração de um ambiente para a manipulação e controle de diferentes serviços de *software* e *hardware* foi escolhido o *framework SOA* Myrobotlab (MRL, 2011), que disponibiliza um conjunto de classes, bibliotecas de código e componentes, que colaboram para prestar diferentes tipos de serviços. Do ponto de vista prático do projeto pretendido, o *framework SOA* é utilizado como uma semi-aplicação flexível e extensível para permitir a elaboração de partes complementares específicas das aplicações do projeto EduBot. A Figura 7 mostra o sistema em execução apresentando a interface gráfica do usuário.

A classe *Runtime* do *framework SOA* Myrobotlab é uma das principais classes do sistema, fornece o acesso ao ambiente de tempo de execução em que o aplicativo está sendo executado e é responsável pela criação e remoção de todos os serviços existentes e registros estáticos associados. *Runtime* é um dos principais serviços existentes do pacote e pode ser acessado através do componente *org.myrobotlab.service.Runtime*, descarregado do repositório (MRL, 2011). Depois de implementado o projeto, ao ser depurado temos a

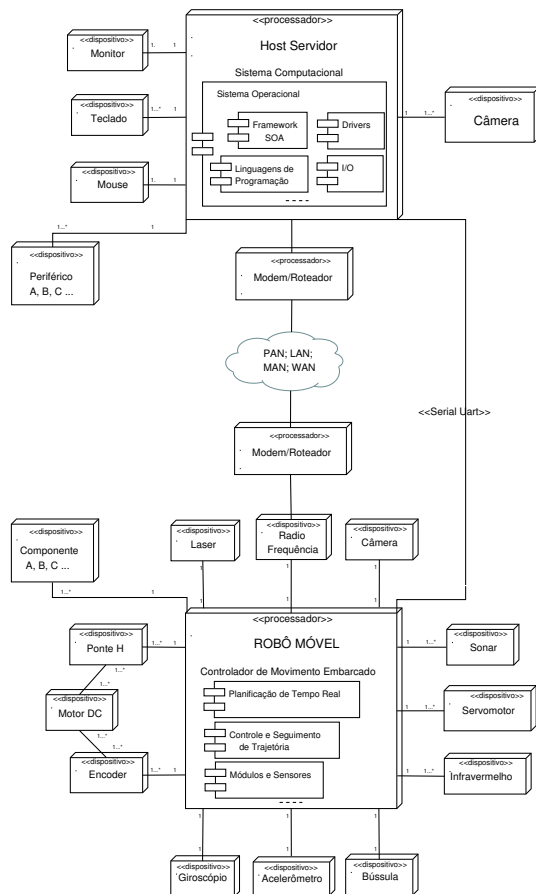


Figura 6: Diagrama de implantação do *Framework* SOA com o robô móvel EduBot. Modelo parcial para o diagrama de implantação, nós com esteriótipos dispositivos (ou processadores) podem ser adicionados ou removidos conforme as necessidades do problema a ser resolvido. Fonte: Autor.

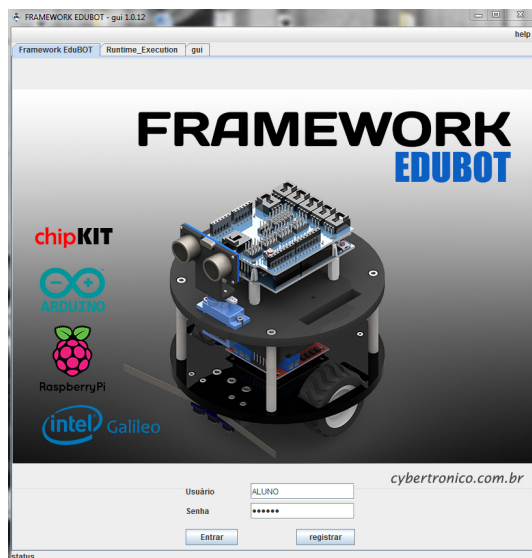


Figura 7: Desenvolvimento da Graphical User Interface (GUI) do usuário do *Framework* SOA. Fonte: Autor

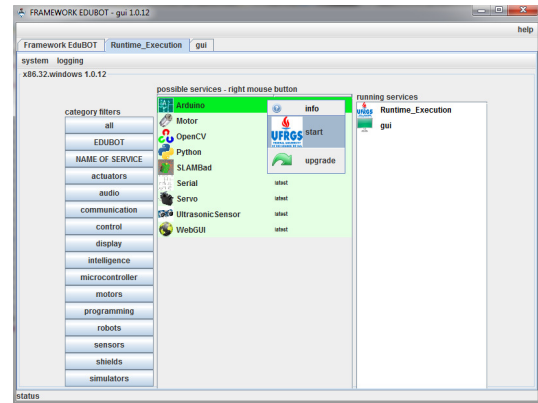


Figura 8: Serviços do *framework*. Fonte: Autor.

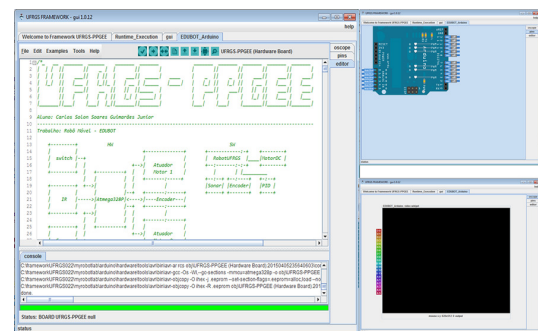


Figura 9: Serviço para edição e programação do *hardware*. Fonte: Autor

execução do *framework* SOA. A Figura 8 mostra os diferentes serviços que estão sendo desenvolvidos para trabalhar com robótica.

O serviço do Arduino é usado atualmente para se comunicar e controlar a versão do robô EduBot que contém a plataforma Arduino, as bibliotecas do *firmware* dos componentes estão desenvolvidos em C++ e encontram-se dentro do *framework* SOA. O *framework* SOA é executado em um *host* que pode ser conectado com a plataforma do *hardware* do robô móvel através de uma porta serial padrão, o qual utiliza-se de uma biblioteca *firmata* que implementa o protocolo para se comunicar com software do computador *host*, desse modo o *framework* SOA detecta a plataforma do robô móvel como uma plataforma *slave* do *framework* SOA. O serviço pode ser utilizado para comunicação, aquisições de dados, programação e envio do *firmware* para o *hardware* do robô móvel. A Figura 9 mostra um dos serviços do *software* do serviço Arduino sendo executado pelo usuário para interação com o *hardware* do robô móvel.

Ao desenvolver uma aplicação dotada de uma GUI, é necessário definir quais componentes (objetos) serão utilizados e a disposição que eles terão na janela. O componente *Swing* possui inúmeras classes utilizadas na construção da GUI do *framework* SOA. Neste ponto, a linguagem começa a ficar interessante, pois as aplicações são criadas a



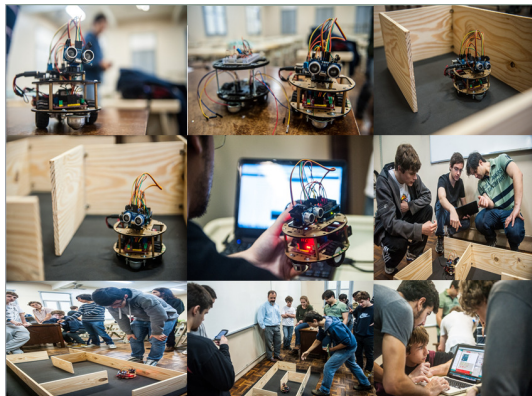


Figura 10: Validação do projeto EduBot integrado com Framework Soa aplicado com estudantes da UFRGS. Fonte: Autor.

partir de janelas gráficas. Ao projetar uma aplicação gráfica, é necessário definir todos os componentes que serão utilizados, seus objetivos e sua posição na janela de maneira modular e flexível, de maneira a facilitar o uso *software* pelos estudantes, como mostra a Figura 10.

Os resultados da conexão da plataforma do robô móvel e *framework* SOA foram aplicados e validados nos discentes das turmas de Graduação de Engenharia da Computação, Engenharia Elétrica e Engenharia de Controle e Automação da Universidade Federal do Rio Grande do Sul (UFRGS), o projeto está em constante evolução para se chegar em uma versão que seja usual para os discentes.

#### 4 Conclusões

Construir um robô móvel e um *framework* SOA, assim como todo sistema grande e complexo, exige um planejamento e uma modelagem prévia. Esse planejamento leva à construção de uma arquitetura, definindo módulos, componentes e suas interações. Com este trabalho, espera-se facilitar o desenvolvimento de projetos para a comunidade de robótica e criar um robô do tipo uniciclo integrado com *framework* SOA possibilitando ser uma ferramenta de ensino e aprendizagem, que pode ser adaptado para estudantes de diferentes níveis e habilidades. Espera-se com esta aplicação, diminuir pelo menos um pouco o retrabalho de construir um projeto de robótica. O projeto vem sendo documentado para o fornecimento de um manual para utilização do *hardware* e *software* do projeto. Trabalhos futuros incluem criar sistemas de controle para o robô EduBot. Programação em blocos, processamento de imagens, áudio e identificação por rádio frequência também estão sendo investigados. O *framework* SOA vem sendo desenvolvido para suportar diferentes plataformas de *hardwares* embarcados.

#### Referências

- Arduino (2005). Plataforma de hardware arduino. Disponível em: <<http://arduino.cc/>>. Acesso em: 03 jan. 2015.
- Barros, T. T. T. (2014). *Modelagem e implementação no ros de um controlador para manipuladores móveis*, Master's thesis, Universidade Federal do Rio Grande do Sul, PPGE.
- da Silva, A. F. (2009). *RoboEduc: Uma Metodologia de Aprendizagem com Robótica Educacional*, PhD thesis, Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal do Rio Grande do Norte, Centro de Tecnologia, RN, Brasil.
- Fugita, H. S. (2009). *Mapos: Método de análise e projeto orientado a serviços*, Master's thesis, Escola Politécnica da Universidade de São Paulo, São Paulo.
- Guimarães, C.S.S; Tamayo, J. H. R. (2014). Robótica para los procesos de enseñanza de la mecatronica: Desarrollo del prototipo edubotv2 para la mejora de procesos de ingeniería de control, *Ciudadania Digital Y Nuevas alfabetizaciones* 5(4): 32–52.
- Lages, W. F. (1998). *Controle e Estimção de Posição e Orientação de Robôs Móveis*, PhD thesis, Instituto Tecnológico de Aeronáutica - ITA, São José dos Campos, SP, Brasil.
- Martins, F. N. (2009). *Modelagem e Compensação da Dinâmica de Robôs Móveis e sua Aplicação em Controle de Formação*, PhD thesis, Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal do Espírito Santo, Av. Fernando Ferrari, ES, Brasil.
- MRL (2011). Framework orientado a serviços. Disponível em: <<http://www.myrobotlab.org/>>. Acesso em: 06 fev. 2015.
- Siegwart, R.; Nourbakhsh, I. (2004). *Autonomous Mobile Robots*, Massachusetts Institute of Technology - MIT.