

# Simulador de Robô para Auxílio ao Ensino de Programação

Marco T. Chella, Claudio M. Oliveira, José Caique O. da Silva

Departamento de Computação

Universidade Federal de Sergipe (UFS) – São Cristóvão, SE – Brazil

Chella@ufs.br, cmoliveira1000@hotmail.com, caiqueprog@yahoo.com.br

**Abstract.** *This project's objective is to develop a mobile robot simulator that applies gamification concepts for improve the programing concepts learning. The simulator can be used on programing disciplines of computer science and engineering courses. The robot control is made by an application programming interface (API) built at this moment for Python and Java programing languages. It is expected from this application to let youth more interested in technology career.*

**Resumo.** *Este projeto tem como objetivo o desenvolvimento de um simulador de robô móvel aplicando o conceito de gamificação para a aprendizagem de conceitos de programação. O simulador pode ser utilizado em cursos de computação e engenharia nas disciplinas de programação. O simulador é constituído por um ambiente gráfico 3D contendo o robô móvel com sensores e atuadores. A interação e controle dos diversos componentes do robô se dão por meio de uma interface de programação de aplicações (API) no momento desenvolvida para as linguagens mais usadas nos cursos de computação incluindo Java, Python. É esperado que a utilização do simulador possa favorecer o desenvolvimento do pensamento computacional nos alunos e estimular jovens em carreiras tecnológicas.*

## 1.Introdução

É conhecido que no Brasil existe um grande índice de reprovação nas disciplinas de programação dos cursos de nível superior, a evasão estudantil no ensino superior é um problema que afeta diretamente o resultado dos sistemas educacionais por não produzir nenhum retorno.

As perdas de estudantes que iniciam e não terminam seus cursos são desperdícios sociais, acadêmicos e econômicos. No setor público, são recursos públicos investidos sem o devido retorno. Já no setor privado, é uma importante perda de receitas. Em ambos os casos, a evasão pode ser tomada como fonte de ociosidade de professores, funcionários, equipamentos e espaço físico [Giraffa e Mora 2014].

Pesquisas feitas mostram que a evasão dos estudantes dos cursos de tecnologia não é algo novo [Pazeto e Prietch 2010]. Um dos motivos para esta desistência está relacionado com as dificuldades encontradas nas disciplinas de programação que possuem como objetivo o desenvolvimento do raciocínio lógico voltado para a resolução de problemas em áreas diversas.

Estes índices de evasão são preocupantes, pois a busca pelos cursos na área das ciências exatas é pequena, o que resulta em muitas vagas em aberto no mercado de trabalho que terminam não sendo preenchidas.

Segundo [Giraffa e Mora 2014] esses índices de desistência estão associados às deficiências relacionadas à expressão em língua materna (escrita), hábitos de estudo e pesquisa, interpretação de texto e devido a uma formação básica deficitária no que

tange a conteúdos de matemática. Outro motivo que pode levar a reprovação nas disciplinas de programação é a falta de motivação e associação da teoria com a prática. Segundo [Neto e Schuvartz 2007] o uso de ferramentas computacionais para o ensino aumenta a produtividade e assimilação do conteúdo estudado.

## 2. Trabalhos Relacionados

Os softwares simuladores, que buscam reproduzir o mundo real, têm sido amplamente utilizados no desenvolvimento e aperfeiçoamento de produtos e estudo de fenômenos físicos. Na área educacional os simuladores podem oferecer uma forma segura e de baixo custo para exploração de conteúdos de estudo.

O SARGE [Craighead et al. 2008] é um simulador de drones de resgate desenvolvido por alunos da University of South Florida que pode ser utilizado para a preparação de profissionais que realizam pesquisas com esses equipamentos, fazendo com que o custo para a preparação dos mesmos seja diminuído significativamente.

O Gazebo, Gazebosim [2014], é um simulador de robôs desenvolvido pela Open Source Robotics Foundation que é usado pela DARPA Robotics Challenge para programação e teste de uma versão virtual de robôs humanóides das forças armadas, com o intuito de desenvolver uma API robusta para o mesmo e diminuir custos.

ProGame, [Dantas e Sales 2010], é um jogo para o ensino de algoritmos e programação com o propósito de aumentar o estímulo no aprendizado de algoritmos por meio da gamificação do estudo de programação.

## 3. Motivação

Segundo o Sebrae [2014] o mercado de jogos digitais no Brasil em 2012 cresceu 32% em relação a 2011 possuindo um total de 35 milhões de usuários. Com essa porcentagem de crescimento considerável em apenas um ano é notório que o público brasileiro possui um interesse crescente por jogos eletrônicos. Dessa forma o uso do simulador que constitui num ambiente gráfico onde existe a interação direta com o usuário, onde são apresentados alguns desafios, pode despertar o interesse pelo estudo de programação.

A inserção de desafios, objetivos e recompensas é conhecida como gamificação [Fardo 2013], que consiste em transformar atividades monótonas em aventuras prazerosas com maior produtividade e aprendizado. Este projeto tem por objetivo unir o ensino utilizando um simulador que aplica o conceito de gamificação, onde o desafio é desenvolver algoritmos para resolução de labirintos e a recompensa seria o aprendizado no adquirido no processo.

Esta abordagem permite ao usuário uma forma livre para testar seus algoritmos interagindo e visualizando de maneira simples e didática como as modificações nos seus próprios algoritmos influenciam no desempenho de determinado agente no decorrer da simulação. Dessa forma, o programador que antes vivia preso a atividades insípidas e a uma tela preta, é inserido em um ambiente onde o mesmo decide o que fazer e como fazer, obtendo resultados mais significativos.

## 4. Ambiente de aplicação

O projeto consiste em um simulador, que assim como O SARGE [Craighead et al. 2008], foi desenvolvido por meio do motor de jogos Unity3D, este foi escolhido por possuir ferramentas que diminuem a complexidade de desenvolvimento (interface gráfica e física), e por possuir uma documentação bastante completa e acessível. O simulador implementa um robô móvel e seu conjunto de sensores e atuadores para interação com o ambiente ao redor do mesmo.

A arquitetura do projeto funciona com a idéia de cliente-servidor, a qual pelo menos dois aplicativos se comunicam via sockets que utilizam o protocolo de controle de transmissão (TCP), de maneira a possibilitar a simulação. O aplicativo servidor é o simulador em si o qual mostra ao usuário, por meio da interface gráfica, o comportamento do(s) robô(s) no ambiente proposto. Já os aplicativos clientes são escritos e compilados pelo próprio usuário, o qual escreverá algoritmos para o(s) robô(s) em uma linguagem de programação de sua escolha no ambiente de desenvolvimento que mais se sentir confortável, fazendo a comunicação com o servidor controlando determinado(s) robô(s), utilizando uma API que é disponibilizada para cada linguagem de programação (atualmente Java e Python). A Figura 2 ilustra o funcionamento do sistema.

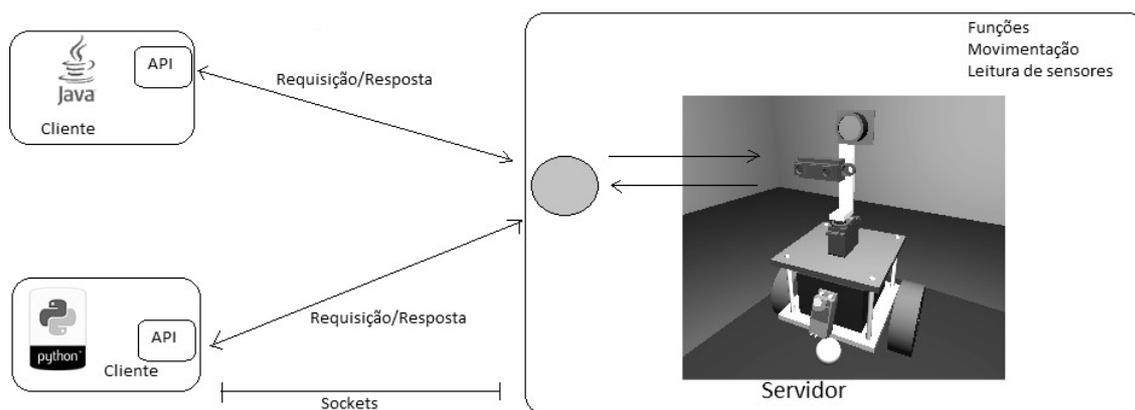


Figura 2: Diagrama de blocos

A API possui uma documentação que especifica as funções que mapeiam os sensores e atuadores do robô mostrados na Figura 4, dessa forma as principais funções para interação do robô com o ambiente são as leituras dos sensores (sonar, infravermelho, etc) e movimentação (frente, ré, direita, esquerda, direcionar sensores). A Figura 3 demonstra um código simples em Java o qual quando em execução faz o robô andar para frente, até detectar um obstáculo, e então para.

```
public static void metodoJava() throws Exception
{
    RoboMovel robo;
    robo = new RoboMovel(); // instancia o robô
    robo.connectIDE(0);     // conecta ao simulador
    robo.servo(90);        // posiciona servomotor
    robo.frente();         // liga motores
    while(sensor()>10.0){} // checa distância para parede
    robo.parar();         // para quando estiver próximo
}
```

Figura 3: Uso de funções do simulador no ambiente Java.

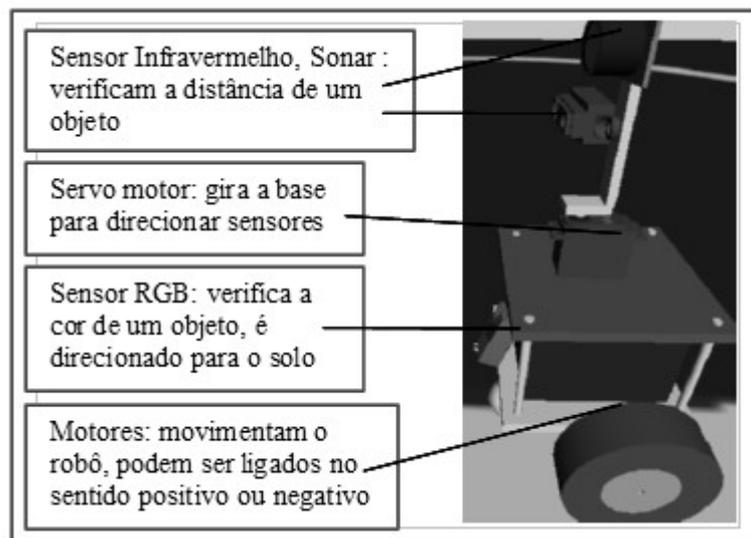


Figura 4: O robô e seus sensores/atuadores.

O simulador possibilita a escolha do ambiente no qual o robô será inserido, podendo variar desde um cenário de sumô de robôs até um labirinto modificável pelo usuário, como mostra a Figura 5. O objetivo da flexibilização do cenário é que assim o usuário poderá fazer um ambiente particular e desenvolver um algoritmo para a resolução do mesmo testando via simulação. Para os problemas que envolvem a resolução de labirintos, o usuário poderá implementar soluções a partir de algoritmos já existentes ou variações deles, como Trémaux's [Nien-Zheng et al. 2011] e o algoritmo de inundação [Willardson 2001], Figura 6 ilustra esse tipo de cenário. A outra forma de simulação que pode ser explorada está relacionada às competições de sumô de robô [Chew 2009], ilustrado na Figura 7.

Essas competições apesar de serem muito importantes por difundirem o uso de robôs [Passold 2006], e possibilitarem o aprendizado em eletrônica, mecânica e programação dos envolvidos nas mesmas, possuem o limitante que é a construção do robô que nem sempre é algo simples e barato, o que se torna uma barreira para o crescimento da área, além de exigir um espaço para a realização de testes. O uso de simuladores é uma boa opção para contornar esse problema, já que o competidor pode desenvolver os algoritmos e testá-los obtendo um resultado realista dispensando a utilização de um robô real.

O simulador possibilita a atuação de mais de um robô no mesmo ambiente, como o sistema é baseado na comunicação em rede o usuário poderá formar equipes em rede local (LAN) para a resolução de problemas, possibilitando o trabalho em equipe, ou poderá fazer competições, ou ainda utilizar mais que uma aplicação no mesmo computador cada uma delas executando um algoritmo diferente para verificar qual seria mais eficiente no ambiente proposto.

Pode acontecer que os problemas a princípio não sejam resolvidos, mas deve ser considerado que com o estudo, e pela visualização do funcionamento de seus algoritmos, o usuário terá um aprendizado significativo, e dessa forma melhorará seu raciocínio para a resolução de problemas computacionais.



Figura 5: Tela de entrada

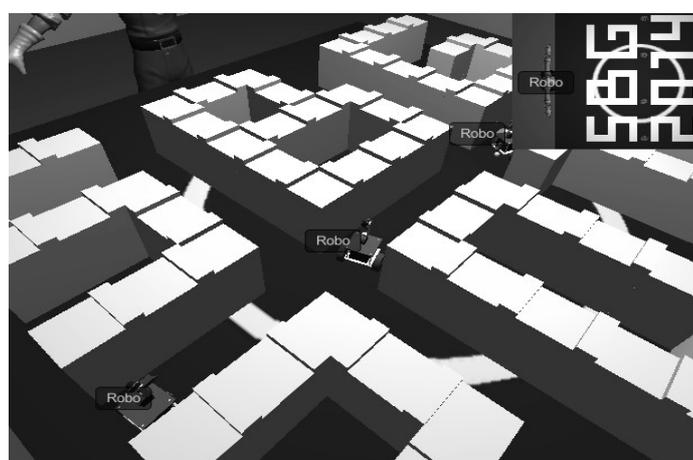


Figura 6: Cenário labirinto

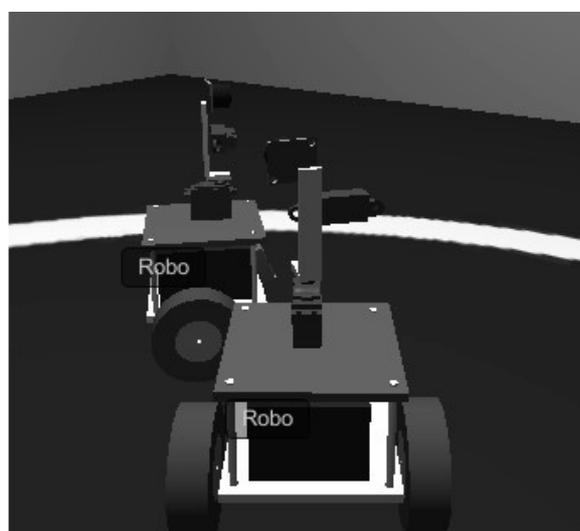


Figura 7: Cenário sumô de robô

## 5. Considerações Finais

Os simuladores apresentam uma dinâmica interessante por toda sua praticidade e experiência que pode ser adquirida durante o seu uso, essa afirmação pode ser feita dado os exemplos apresentados neste artigo onde grandes instituições como a DARPA fazem uso destes.

Aplicar os simuladores para que estes possam auxiliar no aprendizado é uma alternativa interessante podendo trazer uma melhoria do ensino no ambiente de computação. Outro ponto positivo sobre o uso de simuladores é o seu baixo custo que o torna mais acessível que os modelos reais (se tratando de robôs), além de evitar riscos e prejuízos quando está sendo utilizado.

Diante do trabalho feito os próximos passos serão a aplicação do projeto em turmas do departamento de computação da Universidade Federal de Sergipe e a disponibilização dos resultados obtidos.

## Referências

- Chew, M.T. et. al (2009). "Robotics competitions in engineering education". [http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=4804032&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxppls%2Fabs\\_all.jsp%3Farnumber%3D4804032](http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=4804032&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxppls%2Fabs_all.jsp%3Farnumber%3D4804032) [acesso em 12 de julho de 2014].
- Dantas, V. e Sales, C. (2010). "ProGame: um jogo para o ensino de algoritmos e programação". <http://www.br-ie.org/pub/index.php/sbie/article/viewFile/1558/1323> [acesso em 29 de maio de 2014].
- Fardo, M. (2013). "A gamificação aplicada em ambientes de aprendizagem". <http://seer.ufrgs.br/renote/article/viewFile/41629/26409> [acesso em 03 de julho de 2014].
- GazeboSim. <http://gazebosim.org/wiki/Overview/> [acesso em 12 de junho de 2014].
- Giraffa, L. e Mora, M. (2014) "Evasão na Disciplina de Algoritmo e Programação: Um Estudo a partir dos Fatores Intervenientes na Perspectiva do Aluno". [http://www.alfaguia.org/www-alfa/images/ponencias/clabesIII/LT\\_1/ponencia\\_completa\\_136.pdf](http://www.alfaguia.org/www-alfa/images/ponencias/clabesIII/LT_1/ponencia_completa_136.pdf) [acesso em 03 de maio de 2014].
- Neto, W. e Schuvartz, A. (2007). "Ferramenta Computacional de Apoio ao Processo de Ensino/Aprendizagem dos Fundamentos de Programação de Computadores". <http://www.lbd.dcc.ufmg.br/colecoes/sbie/2007/0022.pdf> [acesso em 30 de junho 2014].
- Nien-Zheng, Y. et. al (2011). "Recursive Path-finding in a Dynamic Maze with Modified Tremaux's Algorithm". <http://waset.org/publications/11300/recursive-path-finding-in-a-dynamicmaze-with-modified-tremaux-s-algorithm> [acesso em 14 de julho de 2014].
- Passold, F. (2006). "Despertando para a importância das competições de robô". [http://usuarios.upf.br/~fpassold/sumo\\_cobenge\\_2006.pdf](http://usuarios.upf.br/~fpassold/sumo_cobenge_2006.pdf) [acesso em 12 de julho de 2014].

Pazeto, T. e Prietch, S. (2010). "Estudo sobre a Evasão em um Curso de Licenciatura em Informática e Considerações para Melhorias". <https://www.cesmac.com.br/erbase2010/papers/weibase/65258.pdf>. [acesso em 27 de maio de 2014].

Sebrae. Brasil tem o maior mercado de games do mundo em 2012. <http://www.sebrae2014.com.br/Sebrae2014/Alertas/Brasil-tem-o-maior-mercado-de-games-no-mundo-em-2012#.VEP1cVXF8y4> [acesso em 17 de junho de 2014].

Willardson, D. (2001). "Analysis of Micromouse Maze Solving Algorithms". <http://web.cecs.pdx.edu/~edam/Reports/2001/DWillardson.pdf> [acesso em 14 de julho de 2014].