

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ  
CURSO DE TECNOLOGIA EM SISTEMAS PARA INTERNET  
CÂMPUS GUARAPUAVA

Eduarth Heinen

**RASPIBLOCOS: Ambiente de Programação Didático Baseado em  
Raspberry Pi e Blockly**

PROPOSTA DE TRABALHO DE CONCLUSÃO DE CURSO SUPERIOR EM  
TECNOLOGIA EM SISTEMAS PARA INTERNET

GUARAPUAVA  
1º Semestre de 2015

## 1.1. Título

RASPIBLOCOS – Ambiente de Programação Didático Baseado em Raspberry Pi e Blockly

## 1.2. Modalidade do Trabalho

- (X) Pesquisa
- (X) Desenvolvimento de sistemas

## 1.4. Área do Trabalho

Informática na Educação e Educação em Ciência da Computação.

## 1.5. Resumo

A utilização da robótica na educação é tema de projetos de pesquisa desde a década de 80. Entre os resultados relatados por projetos nesta área, destacam-se o aumento da participação dos alunos, o desenvolvimento de habilidades de raciocínio lógico e resolução de problemas, bem como o estímulo do interesse por campos da Ciência, Tecnologia, Engenharia e Matemática. Recentemente, este ramo de pesquisa recebeu o incentivo da popularização de processadores ARM – *Advanced RISC Machine* – e de computadores de placa única como Arduíno e Raspberry Pi. Tais recursos simplificam a construção de modelos robóticos e experiências eletrônicas. Observando estas oportunidades, este projeto propõe o desenvolvimento de um ambiente de programação que possibilite programar o comportamento de um conjunto de componentes eletrônicos por meio da interface de uma aplicação web. A interface será construída utilizando a biblioteca Blockly, de forma que pequenos programas possam ser construídos encaixando blocos que representam instruções predefinidas. Um minicomputador Raspberry Pi será responsável por controlar componentes de *hardware* como *leds*, sensores, botões e atuadores. Após o desenvolvimento do ambiente, se houverem alunos interessados e recursos disponíveis, a viabilidade de realizar oficinas de programação deve ser considerada. Nestas oficinas alunos de graduação em cursos que incluam programação em seus currículos, serão guiados através da construção e programação de um modelo robótico utilizando o experimento. O projeto pretende construir uma ferramenta que possibilite o desenvolvimento de lições de programação estimulantes e dirigidas à experimentação.

## 2. PROPOSTA DE TRABALHO

### 2.1. Introdução

Existem indícios de que a aplicação de metodologias de ensino que envolvam a construção autônoma de teorias pelos alunos alcance melhores resultados que abordagens baseadas em aulas expositivas (PAPERT, 1980). Apoiados por estes indícios, pesquisadores têm proposto a utilização da robótica na construção de ambientes que sirvam de contexto para a formulação e validação de hipóteses. Projetos nesse sentido buscam utilizar a robótica e a tecnologia na educação, com o objetivo de estimular o interesse e o aprendizado dos alunos em conteúdos que envolvam conceitos complexos.

A popularização de brinquedos programáveis como a linha Lego Mindstorms<sup>1</sup> e computadores de placa única como Arduíno<sup>2</sup> e Raspberry Pi<sup>3</sup> simplificou a construção de modelos robóticos, apoiando o surgimento de projetos de pesquisa na área de Informática na Educação. Integrando robôs e noções de programação ao conteúdo apresentado aos alunos, estes projetos buscam avaliar o impacto da interação com robótica e tecnologia no aprendizado. Entre os

---

<sup>1</sup> Lego Mindstorms – <http://www.lego.com/en-us/mindstorms/>

<sup>2</sup> Arduíno – <http://www.arduino.cc/>

<sup>3</sup> Raspberry Pi – <https://www.raspberrypi.org/>

resultados observados, destacam-se o aumento da participação e motivação dos alunos, além do desenvolvimento de habilidades de raciocínio lógico e resolução de problemas (SAYGIN et al., 2012; SALEIRO et al., 2013) . Iniciativas dessa natureza, junto de organizações como Code.org<sup>4</sup>, Computing at School<sup>5</sup> e The Raspberry Foundation<sup>6</sup>, têm o propósito de democratizar o acesso à programação e robótica, e trabalham para integrar esses conteúdos ao currículo escolar, possibilitando que mais pessoas aprendam a programar e a construir modelos eletrônicos.

A carência de profissionais em áreas relacionadas aos campos da Ciência, Tecnologia, Engenharia e Matemática<sup>7</sup> (ITURRATE et al., 2013; VANDEVELDE et al., 2013) é outro fator apontado pelos autores como motivação para a pesquisa. O contato com experiências que integram tecnologia e robótica em lições escolares é percebido como um incentivo ao interesse por essas áreas.

Brinquedos programáveis, como a linha LEGO Mindstorms, embora amplamente utilizados em exposições, lições e competições de robótica (GROUT; HOULDEN, 2014) , são vendidos a preços considerados restritivos. Soluções de baixo custo como a placa eletrônica programável Picoboard<sup>8</sup>, por outro lado, são acessíveis, mas oferecem recursos limitados. Projetos de pesquisa em sua maioria descrevem soluções que dependem de um requisito específico, como uma placa impressa por encomenda, ou representam soluções proprietárias, utilizadas por uma universidade ou projeto de pesquisa em andamento.

Buscando contribuir para o campo de pesquisa, o corrente projeto pretende desenvolver um ambiente de programação que permita construir pequenos programas, encaixando blocos coloridos em sua interface. Esses programas devem, então, serem traduzidos em instruções enviadas à componentes eletrônicos conectados ao minicomputador Raspberry Pi. Desta forma, pretende-se oferecer um ambiente capaz de traduzir programas, compostos por blocos, em reações no modelo físico (Raspberry Pi). Como consequência, pretende-se estimular o interesse e a participação dos alunos em lições de programação, permitindo que conceitos complexos sejam demonstrados.

Depois desta introdução, na Seção 2.2, são apresentados os objetivos da proposta. Cabe a Seção 2.3 expor o estado da arte relacionado. Na Seção 2.4, são expostos os procedimentos metodológicos envolvidos na construção do projeto. O diferencial tecnológico é exposto na Seção 2.5. A Seção 2.6 apresenta o calendário de desenvolvimento do projeto. Os recursos necessários são apresentados na Seção 2.7, e a disponibilidade destes recursos é exposta na Seção 2.8. A Seção 2.9 apresenta o planejamento de horários destinados ao desenvolvimento do projeto. A Seção 2.10, ao final do documento, expõe as Referências Bibliográficas citadas na proposta.

## 2.2. Objetivos

### 2.2.1. Objetivo Geral

Desenvolver um ambiente de programação visual utilizando a biblioteca Blockly e o minicomputador Raspberry Pi, de maneira que reações físicas no modelo eletrônico decorram de instruções representadas pelos blocos encaixáveis na interface.

### 2.2.2. Objetivos Específicos

- Construir uma aplicação web com interface baseada na biblioteca Blockly;
- Integrar a aplicação e o minicomputador Raspberry Pi;
- Programar blocos da interface para controlar *leds*, motores de corrente contínua, sensores de proximidade e botões;
  - Elaborar um conjunto de exemplos que apresente o potencial da ferramenta e utilizá-lo em experimentos com alunos de graduação;
  - Disseminar os resultados em publicações e eventos da área.

<sup>4</sup> Code.org – [code.org/](http://code.org/)

<sup>5</sup> Computing at School – [www.computingatschool.org.uk/](http://www.computingatschool.org.uk/)

<sup>6</sup> Raspberry Foundation – [www.raspberrypi.org/](http://www.raspberrypi.org/)

<sup>7</sup> Do inglês, *Science, Technology, Engineering and Mathematics* (STEM).

<sup>8</sup> Picoboard – <http://www.picocrocket.com/picoboard.html>

### 2.3. Estado da Arte

O desenvolvimento de um conjunto de robótica voltado à educação, com um controlador Raspberry Pi e interface baseada em Blockly, foi proposto anteriormente por Saleiro et al. (2013). A equipe de pesquisadores desenvolveu um sistema que utiliza o minicomputador como controlador de múltiplos robôs. Cada pequeno robô, chamado “robô infante”, é capaz de se mover através de uma grade de linhas pretas com fundo branco. A interface da aplicação, baseada em Blockly, oferece blocos com instruções como “seguir em frente” e “girar 90° à direita”, permitindo que o usuário construa um pequeno roteiro para levar o robô até os objetivos marcados no mapa.

Os autores apontam que esse conjunto pode ser utilizado por professores em lições de diversas disciplinas, simplesmente alterando os objetivos e o contexto do exercício para corresponder ao conteúdo da lição. Além de motivar os alunos a aprenderem o conteúdo abordado, então percebido como um requisito necessário para que o robô possa atingir os objetivos do exercício, o sistema possibilita desenvolver habilidades de raciocínio matemático e dedução lógica.

A metodologia aplicada pelos pesquisadores incluiu lições de Geografia e de Reciclagem de Lixo Eletrônico, com turmas da 3ª e 4ª séries do ensino fundamental em escolas de Faro em Portugal. A lição de Geografia, por exemplo, definiu como objetivos no mapa a posição das principais montanhas de Portugal em relação a Faro, permitindo abordar o conteúdo da disciplina e desenvolver habilidades de raciocínio matemático, dedutivo, abduutivo e indutivo. Os pesquisadores relataram que a utilização do sistema de robótica nas lições levou os alunos a adotarem a postura de formuladores de hipóteses, buscando soluções em conjunto e discutindo-as abertamente (SALEIRO et al., 2013).

Entretanto, a aplicação utilizada no experimento de Saleiro et al. (2013) não está disponível ao público, e os robôs, mesmo sendo feitos com materiais acessíveis, dependem de uma placa impressa por encomenda. Além disso, o projeto de Saleiro é focado em alunos do ensino básico e, portanto, as instruções disponíveis na interface são bastante simples e limitadas às capacidades dos robôs do experimento.

Outra experiência com objetivos semelhantes foi desenvolvida por Iturrate et al. (2013). Neste projeto, um robô, construído sobre um minicomputador Arduíno Uno, navega através de um labirinto de acordo com as instruções recebidas do computador que controla o experimento. As imagens do laboratório onde o labirinto e o robô se encontram são transmitidas pela Internet até o usuário da aplicação. O conceito de laboratórios remotos, aplicado na pesquisa, busca oferecer uma alternativa para a democratização do acesso ao conhecimento desenvolvido nas universidades, conectando os experimentos à Internet. Desta maneira, a pesquisa pretendeu possibilitar que estudantes de qualquer lugar do mundo participassem das experiências desenvolvidas (ITURRATE et al., 2013).

O experimento de Iturrate et al. (2013) baseia-se no modelo proposto anteriormente por Dziabenko et al. (2012). A interface do jogo apresenta os jogadores ao cenário de um pouso forçado em um planeta alienígena. O jogador, como integrante da tripulação da nave, deve programar o robô para recuperar as peças espalhadas pelo cenário. Ao alcançar cada um dos objetivos, representados por etiquetas de rádio frequência (RFID), uma questão relacionada ao tema abordado pela lição é apresentada ao jogador. Dessa forma, assim como no experimento de Saleiro, os jogadores percebem o conteúdo como requisito para alcançar os objetivos, sendo que diversos conteúdos diferentes podem ser apresentados pela simples alteração do cenário ou o tema das perguntas.

Como os robôs são construídos sobre o controlador Arduíno Uno, a aplicação que controla o experimento de Iturrate envia somente instruções previamente programadas aos robôs. Assim como o projeto de Saleiro, esses robôs dependem de uma placa impressa sob encomenda, mas os autores assumem a possibilidade de expandir as instruções disponíveis na aplicação. No momento da publicação do artigo, ainda não haviam sido realizados testes da aplicação e, como os autores apontam em seus resultados, a interface ainda não oferecia todos os recursos de contextualização necessários para criar o ambiente proposto.

Ambos os projetos mencionados constituem ferramentas com potencial para a utilização multidisciplinar da robótica na educação. Entre as vantagens observadas, estão a facilidade de

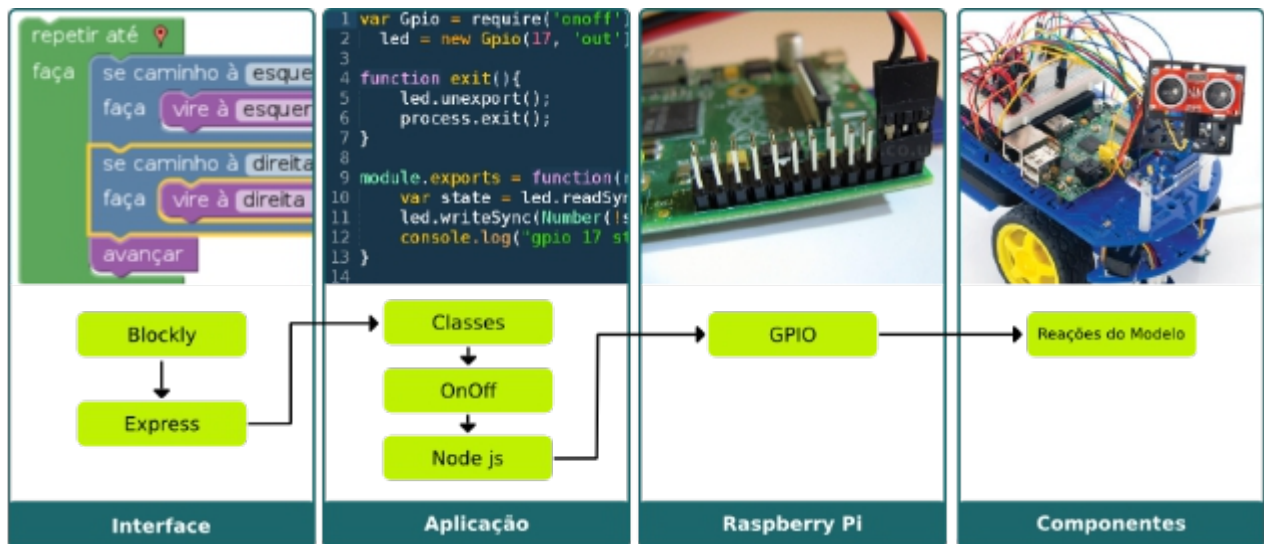
configuração e utilização, apontada como objetivo de ambos os projetos de pesquisa, bem como a preocupação dos pesquisadores em criar o contexto para apoiar os objetivos dos robôs.

Considerando que ambos os projetos possuem limitações, argumenta-se que uma iniciativa aberta construída com componentes acessíveis atingiria, potencialmente, maior abrangência. A fluência em JavaScript do Raspberry Pi permite que o próprio minicomputador atue como controlador tanto da experiência quanto dos componentes eletrônicos. Do ponto de vista didático, destaca-se também, que as lições poderiam ser mais complexas, permitindo tratar conceitos de linguagens de programação e eletrônica, direcionando o projeto para alunos de graduação.

## 2.4. Procedimentos Metodológicos

O desenvolvimento do projeto será dividido nas seguintes etapas:

1. **Construir a aplicação web inicial:** A aplicação que atuará como interface de programação do experimento será desenvolvida utilizando o *ramework* Express e a plataforma Node.js. A biblioteca Blockly também será utilizada, com a finalidade de prover um ambiente visual de programação por meio de blocos encaixáveis. A Figura 1 ilustra a arquitetura e a interação entre os componentes da aplicação, a biblioteca Blockly e o *ramework* Express constituem a interface, que é a primeira camada da arquitetura. Deve-se ter, ao fim desta etapa, um protótipo funcional do ambiente;



**Figura 1:** Representação da arquitetura da aplicação, ilustrando a interação entre os componentes: a biblioteca Blockly e o framework Express constituem a Interface. A camada de Aplicação, formada pelas classes específicas, a biblioteca OnOff e a plataforma Node.js, recebe as instruções enviadas pela Interface e aciona os métodos que controlam as portas GPIO do minicomputador. As portas GPIO, por sua vez, controlam os componentes eletrônicos conectados, resultando em movimentos do modelo eletrônico.

2. **Integrar a aplicação ao minicomputador Raspberry Pi:** A biblioteca *OnOff*<sup>9</sup> oferece um conjunto de métodos que permitem controlar o estado das portas GPIO<sup>10</sup> do Raspberry Pi. A classe responsável por receber as instruções da interface e controlar as portas físicas do minicomputador será construída utilizando estes métodos. Funções como “aguardar x segundos” e “executar rotina” devem ser adicionadas à mesma classe. Também serão instalados os programas necessários para hospedar a aplicação no minicomputador. Desta etapa deve resultar um protótipo capaz de acionar *leds* e motores de corrente direta

<sup>9</sup> Biblioteca OnOff – <https://github.com/fivdi/onoff>

<sup>10</sup> GPIO, do inglês General Purpose Input Output, portas programáveis de entrada e saída de dados.

utilizando estruturas de controle (“se-então”, “enquanto-faça”) e blocos para as portas GPIO do Raspberry Pi;

**3. Programar blocos para representar componentes e instruções específicos da aplicação:** Por meio da especialização do código responsável pelo controle das portas GPIO, devem ser desenvolvidas classes e blocos que representem o funcionamento de *leds*, motores de corrente direta, botões e sensores de proximidade. Como exposto na Figura 1, as instruções enviadas pelos blocos da interface serão interpretadas por estas classes acionando métodos que, através da biblioteca OnOff, controlarão as portas físicas do minicomputador. Prevê-se a aplicação de técnicas de Desenvolvimento Dirigido por Testes ou TDD<sup>11</sup>. Esta metodologia, além de fornecer uma ferramenta para testar a aplicação de maneira instantânea e abrangente, ajuda a constituir a documentação inicial do projeto, formada pelas especificações que cada teste descreve. A construção do modelo robótico deve acompanhar o desenvolvimento. Com isso a definição de cada classe deve ser seguida da adição do respectivo componente ao modelo. O resultado desta etapa será um protótipo que ofereça blocos com instruções específicas para cada componente eletrônico suportado pelo experimento. Esta camada de abstração permitirá ao usuário fornecer instruções de mais alto nível aos componentes do minicomputador. As instruções previstas para cada componente são:

- Motores: “mover para frente” e “mover para trás”;
- Botões: “quando pressionado faça...”;
- Leds: “acender” e “apagar”;
- Sensores de distância: “calcular distância” e “quando distância for menor que x cm faça...”;

Trata-se do primeiro protótipo testável, em que, utilizando as instruções disponíveis, será possível construir e programar um robô capaz de se mover por uma sala evitando obstáculos;

**4. Elaborar um conjunto de exemplos de utilização do ambiente e propor a realização de experimentos com acadêmicos:** Um conjunto de exemplos, com a finalidade de demonstrar o potencial da ferramenta, deve ser distribuído junto da aplicação. Estes exemplos devem orientar a programação dos componentes do modelo eletrônico. Será verificada também a possibilidade de se realizar um experimento com alunos de graduação em cursos que incluam disciplinas de programação em seus currículos, buscando avaliar o potencial didático e refinar o funcionamento da aplicação. Entretanto, a realização do experimento é sujeita a fatores como disponibilidade de alunos interessados, recursos para aquisição dos equipamentos, além do trâmite de importação destes. O experimento consistirá de uma oficina dividida em três etapas: (1) o modelo eletrônico e a interface serão apresentados aos alunos que serão guiados pela montagem e programação dos componentes; (2) os alunos serão desafiados a construir um robô capaz de se mover pela sala evitando obstáculos; (3) ao final da oficina deve ser realizada uma pesquisa de opinião sobre o experimento e a oficina. Presume-se que poderá haver dificuldades de interpretação das instruções representadas pelos blocos. Assim, os testes também devem fornecer indícios desses problemas e de como resolvê-los. Os dados obtidos serão analisados em busca de indícios que apoiem o atributo didático dado ao experimento;

**5. Disseminar os resultados em publicações e eventos da área:** Os resultados alcançados na pesquisa serão analisados e submetidos a eventos da área. A aplicação será distribuída como código aberto.

## 2.5. Diferencial Tecnológico

<sup>11</sup> TDD, do inglês *Test Driven Development*, desenvolvimento guiado por testes.

As principais características em que este projeto difere dos projetos descritos no Estado da Arte são: a distribuição do projeto como código aberto; o minicomputador Raspberry Pi atuando simultaneamente como hospedeiro da aplicação e controlador dos componentes; e o objetivo de desenvolver lições de programação para alunos de graduação.

Considerando a popularidade do minicomputador Raspberry Pi<sup>12</sup>, a distribuição livre da aplicação do experimento deve permitir que novos usuários entrem em contato com o projeto. Um maior número de usuários possibilita que surjam mais críticas e sugestões, além de atrair a contribuição de outros programadores na expansão e desenvolvimento da aplicação.

Argumenta-se que a utilização do minicomputador como hospedeiro da aplicação e controlador dos componentes constitui uma vantagem e um desafio. Dessa maneira é possível programar toda a aplicação e o controle dos componentes utilizando uma única linguagem de programação, JavaScript. Porém, a alimentação do conjunto se torna uma questão a ser considerada, uma vez que o consumo de energia do minicomputador, somado ao consumo de cada componente envolvido no modelo eletrônico, pode exigir baterias potentes.

A pesquisa dirigida por Saleiro (2013) foi direcionada para alunos do ensino fundamental e, sendo assim, a interface da aplicação foi construída para que os eles pudessem criar roteiros de ações que seriam realizadas pelo minirobô. Direcionando o experimento a acadêmicos de graduação, pretende-se criar modelos e blocos gradativamente mais complexos, expandindo as capacidades do experimento.

## 2.6. Planejamento do Trabalho

A Tabela 1 apresenta o calendário das etapas de desenvolvimento do projeto. Conforme indicado no calendário, a aplicação inicial já foi desenvolvida. A integração com o minicomputador, bem como a definição dos blocos e classes específicos são previstos para o final do semestre. O próximo semestre compreenderá a análise da possibilidade de testes, escrita do documento monográfico final e tentativa de publicação dos resultados.

---

<sup>12</sup> Raspberry Pi no Palácio de Buckingham, 3 milhões de unidades vendidas (tradução livre) - <http://www.raspberrypi.org/Raspberry-Pi-at-buckingham-palace-3-million-sold/>

**Tabela 1:** Distribuição das etapas de desenvolvimento do projeto através dos dois semestres e das disciplinas de TCC.

Atividades	TCC I					TCC II				
	Mar	Abr	Mai	Jun	Jul	Ago	Set	Out	Nov	Dez
1. Desenvolvimento da aplicação inicial.										
- Entrega da Proposta de TCC - 06/04										
- Defesa da Proposta de TCC - 13 a 17/04										
2. Integração com o minicomputador.										
- Entrega do Projeto de TCC - 15/06										
- Defesa do Projeto de TCC - 22 a 26/06										
3. Blocos e instruções específicos.										
4. Testes em oficinas de programação.										
5. Publicação dos resultados.										
- Entrega do TCC - data a ser definida										
- Defesa do TCC - data a ser definida										

## 2.7. Recursos Necessários

Os recursos necessários para o desenvolvimento do projeto são: um minicomputador Raspberry Pi, um adaptador Wi-Fi, uma placa de ensaio, dois motores de corrente direta, um sensor de proximidade, leds, resistores, baterias, um regulador de tensão, um chassi plástico e rodas.

## 2.8. Disponibilidade dos Recursos Acima Relacionados

Dentre os recursos listados, estão disponíveis: o minicomputador Raspberry Pi, a placa de ensaio, leds e resistores. Foram encomendados dois motores de corrente direta com rodas, o sensor de proximidade e o chassi plástico. Os modelos e configurações do adaptador *Wi-Fi* e do regulador de tensão já foram definidos, porém ainda não encomendados. As baterias podem ser pilhas genéricas, não impedindo que melhores alternativas sejam utilizadas.

## 2.9. Horário de Trabalho

Os horários de trabalho destinados ao desenvolvimento do projeto e às reuniões de orientação constam na Tabela 2.



**Tabela 2:** Agenda semanal de desenvolvimento do projeto indicando, também, o dia e horário da reunião semanal de orientação.

	seg	ter	qua	qui	sex	sab
13h - 13h50m						
13h50m - 14h40m		TCC		TCC		TCC
14h40m - 15h30m		TCC		TCC		TCC
15h40m - 16h30m		TCC		TCC		TCC
16h30m - 17h20m						
17h20m - 18h10m				ORT		
18h50m - 19h40m						
19h40m - 20h30m						
20h30m - 21h20m						
21h30m – 22h15m		TCC			TCC	
22h15m - 22h50m		TCC			TCC	

## 2.10. Referências Bibliográficas

DZIABENKO, O; GARCÍA-ZUBIA, J.; ANGULO, I. Time to play with a microcontroller managed mobile bot. In: IEEE. **Global Engineering Education Conference (EDUCON), 2012 IEEE**. [S.l.], 2012. p. 1–5.

GROUT, V.; HOULDEN, N. Taking computer science and programming into schools: The glyndŵr/bcs turing project. **Procedia-Social and Behavioral Sciences**, Elsevier, v. 141, p. 680–685, 2014.

ITURRATE, I. et al. A mobile robot platform for open learning based on serious games and remote laboratories. In: IEEE. **1st International Conference of the Portuguese Society for Engineering Education (CISPEE), 2013**. [S.l.], 2013. p. 1–7.

PAPERT, S. **Mindstorms: Children, Computers, and Powerful Ideas**. New York, NY, USA: Basic Books, Inc., 1980. ISBN 0-465-04627-4.

SALEIRO, M. et al. A low-cost classroom-oriented educational robotics system. In: **Social Robotics**. [S.l.]: Springer, 2013. p. 74–83.

SAYGIN, C. et al. Design, development, and implementation of educational robotics activities for k-12 students. In: **Proceedings of 2012 American Society for Engineering Education Annual Conference & Exposition**. [S.l.: s.n.], 2012.

VANDEVELDE, C. et al. Overview of technologies for building robots in the classroom. In: **International Conference on Robotics in Education**. [S.l.: s.n.], 2013. p. 122–130.