

UNIVERSIDADE DE PASSO FUNDO
INSTITUTO DE CIÊNCIAS EXATAS E GEOCIÊNCIAS
PROGRAMA DE PÓS-GRADUAÇÃO EM
COMPUTAÇÃO APLICADA

**Plataforma interativa de aprendizagem de
programação voltada a disseminação do
pensamento computacional utilizando
robótica remota**

Velcir Barcaroli

Passo Fundo

2017

UNIVERSIDADE DE PASSO FUNDO
INSTITUTO DE CIÊNCIAS EXATAS E GEOCIÊNCIAS
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO APLICADA

**PLATAFORMA INTERATIVA DE
APRENDIZAGEM DE
PROGRAMAÇÃO VOLTADA A
DISSEMINAÇÃO DO PENSAMENTO
COMPUTACIONAL UTILIZANDO
ROBÓTICA REMOTA**

Velcir Barcaroli

Dissertação apresentada como requisito parcial
à obtenção do grau de Mestre em Computação
Aplicada na Universidade de Passo Fundo.

Orientador: Prof. Dr. Marco Antonio Sandini Trentin

Passo Fundo

2017

CIP – Catalogação na Publicação

B242p Barcaroli, Velcir
Plataforma interativa de aprendizagem de programação
voltada a disseminação do pensamento computacional
utilizando robótica remota / Velcir Barcaroli. – 2017.
71 f. : il. color. ; 30 cm.

Orientador: Prof. Dr. Marco Antonio Sandini Trentin.
Dissertação (Mestrado em Computação Aplicada) –
Universidade de Passo Fundo, 2017.

1. Robótica. 2. Robôs - Programação. 3. Computação -
Educação. I. Trentin, Marco Antonio Sandini, orientador.
II. Título.

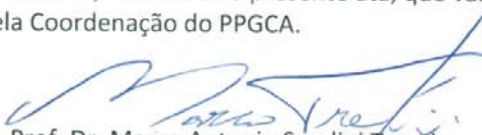
CDU: 004.42

Catálogo: Bibliotecário Luís Diego Dias de S. da Silva – CRB 10/2241


**ATA DE DEFESA DO
TRABALHO DE CONCLUSÃO DE CURSO DO ACADÊMICO**


VELCIR BARCAROLI

Aos dezoito dias do mês de junho do ano de dois mil e dezessete, às 14 horas, realizou-se, no Instituto de Ciências Exatas e Geociências, prédio B5, da Universidade de Passo Fundo, a sessão pública de defesa do Trabalho de Conclusão de Curso "**Plataforma interativa de aprendizagem de programação voltada a disseminação do pensamento computacional utilizando robótica remota**", de autoria de Velcir Barcaroli, acadêmico do Curso de Mestrado em Computação Aplicada do Programa de Pós-Graduação em Computação Aplicada – PPGCA/UPF. Segundo as informações prestadas pelo Conselho de Pós-Graduação e constantes nos arquivos da Secretaria do PPGCA, o aluno preencheu os requisitos necessários para submeter seu trabalho à avaliação. A banca examinadora foi composta pelos doutores Marco Antonio Sandini Trentin, Luiz Eduardo Schardong Spalding e Roberta Pasqualli. Concluídos os trabalhos de apresentação e arguição, a banca examinadora considerou o candidato APROVADO. Foi concedido o prazo de até quarenta e cinco (45) dias, conforme Regimento do PPGCA, para o acadêmico apresentar ao Conselho de Pós-Graduação o trabalho em sua redação definitiva, a fim de que sejam feitos os encaminhamentos necessários à emissão do Diploma de Mestre em Computação Aplicada. Para constar, foi lavrada a presente ata, que vai assinada pelos membros da banca examinadora e pela Coordenação do PPGCA.


Prof. Dr. Marco Antonio Sandini Trentin
Presidente da Banca Examinadora
(Orientador)


Prof. Dr. Luiz Eduardo Schardong Spalding
(Avaliador Interno)


Profa. Dra. Roberta Pasqualli - IFSC
(Avaliadora Externa)


Prof. Dr. Rafael Rieder
Coordenador do PPGCA

Dedico este trabalho a minha família, em especial
a minha esposa e meu filho.

AGRADECIMENTOS

Agradeço acima de tudo a Deus pelo dom da vida e pela oportunidade de vencer mais essa etapa. Agradeço ao Espírito Santo por me iluminar todos os dias e permitir melhorar sempre, como pessoa, cristão e profissional.

Agradeço a minha esposa Janinha e meu filho Eduardo pelas orações, pela paciência, auxílio e compreensão durante o desenvolvimento desse trabalho. Sem o apoio de vocês nada faria sentido.

Agradeço a minha mãe Izélia, meus irmãos Volmir e Vilamir, meus cunhados e cunhadas, meus sobrinhos e afiliados e a toda a família pelo incentivo, palavras de apoio e compreensão em todos os momentos. Meu pai Olindo (*in memoriam*), que em sua humildade foi sempre um incentivador para a busca da formação acadêmica. A família é a sustentação do ser humano.

Agradeço o meu orientador Marco Antônio Sandini Trentin por acreditar no meu trabalho e com seu apoio e experiência fazer que essa dissertação chegasse ao final. Também agradeço ao coordenador do PPGCA professor Carlos Holbig e demais professores pelo aprendizado que me proporcionaram. Aproveito e estendo meus agradecimentos a todos os funcionários do programa.

Agradeço o professor Elton Zeni, seus conselhos foram importantes em minha decisão para o mestrado. Quero também agradecer a todos os colegas de trabalho pelo apoio.

Por fim, agradeço aos colegas de mestrado, a boa convivência que tivemos, as trocas de experiências, a colaboração em atividades, tudo isso contribui para meu crescimento pessoal e profissional.

Muito obrigado a todos que de alguma forma contribuíram para essa formação.

“Eu não sou quem eu gostaria de ser; eu não sou quem eu poderia ser, ainda, eu não sou quem eu deveria ser. Mas graças a Deus eu não sou mais quem eu era! ”

(Martin Luther King Jr.)

PLATAFORMA INTERATIVA DE APRENDIZAGEM DE PROGRAMAÇÃO VOLTADA A DISSEMINAÇÃO DO PENSAMENTO COMPUTACIONAL UTILIZANDO ROBÓTICA REMOTA

RESUMO

O pensamento computacional no mundo contemporâneo é uma habilidade fundamental para qualquer um, não apenas para cientistas da computação. Juntamente com a leitura, a escrita e a aritmética, sugere-se adicionar o pensamento computacional na habilidade analítica de cada criança. A Robótica Pedagógica, que é caracterizada por ambientes de aprendizagem onde o aluno pode montar e programar um robô, poderá ser um caminho para disseminação do Pensamento Computacional. Acredita-se que este trabalho oportunizará a jovens estudantes o contato com tecnologias digitais e atuais, favorecendo o desenvolvimento do pensamento computacional entre eles. O objetivo desse trabalho é construir uma plataforma que permita a programação remota de um braço robótico através da internet afim de que possa ser utilizada por alunos do ensino fundamental e médio, como meio para disseminação do pensamento computacional. Nessa pesquisa são apresentadas tecnologias e ferramentas que viabilizam o desenvolvimento de uma plataforma de programação remota de um braço robótico. A plataforma está dividida em quatro módulos: a parte física compreendendo arena e braço robótico, módulo de imagem responsável pela geração dos vídeos, módulo de interface com o usuário que permite a programação do robô em qualquer computador conectado a internet e o módulo de integração responsável por gerenciar toda a plataforma. Na realização de testes com voluntários, a plataforma se mostra como desafiadora, provocando nos mesmos um sentimento de competição, diversão e motivação para a realização dos desafios propostos. Esse ambiente permite que alunos aprendam conceitos de robótica e interajam de forma interativa com um robô, programando-o e compreendendo-o, sem ter a necessidade de possuir um dispositivo robótico

Palavras-Chave: Braço Robótico Remoto, Laboratórios Remotos, Pensamento Computacional, Plataforma Interativa, Robótica Educacional.

INTERACTIVE LEARNING PLATFORM FOR PROGRAMMING TO DISSEMINATE COMPUTATIONAL THINKING USING REMOTE ROBOTICS

ABSTRACT

Computational thinking in the contemporary world is a fundamental skill for anyone, not just computer scientists. Along with reading, writing, and arithmetic, it is suggested to adding computational thinking to each child's analytical ability. Pedagogical Robotics, which is characterized by learning environments where the student can assemble and program a robot, could be a way to disseminate Computational Thinking. It is believed that this work will make possible young students the contact with current and digital technologies, favoring the development of computational thinking among them. The objective of this study is to build a platform that allows students of primary and secondary education, as a means to spread the computational thinking, can use remote programming of a robot arm via the internet so that. In this research are presented technologies and tools that enable the development of a remote programming platform of a robotic arm. The platform divided into four modules: the physical part comprising robotic arena and arm, image module responsible for generating the videos, user interface module that allows the robot to be programmed on any computer connected to the internet and the integration module responsible For managing the entire platform. In conducting tests with volunteers, the platform proves to be challenging, provoking in them a sense of competition, fun and motivation to achieve the proposed challenges. This environment allows students to learn robotics concepts and interact interactively with a robot, programming it and understanding it, without having to have a robotic device.

Keywords: Remote Robotic Arm, Computational Thinking, Educational Robotics, Interactive Platform, Remote Laboratories.

LISTA DE FIGURAS

Figura 1.	Parte física do LabVad.	27
Figura 2.	Área de codificação.	28
Figura 3.	A Sala de Desafios.	29
Figura 4.	Código NXT-python.	30
Figura 5.	Arena com os robôs do projeto SyRoTek.	31
Figura 6.	Ambiente de acesso e desenvolvimento do SyRoTek.	32
Figura 7.	Interface do Amazon S3.	38
Figura 8.	Esquemático.	42
Figura 9.	Tela de login.	43
Figura 10.	Tela de inicial do sistema.	43
Figura 11.	Tela de inicial do sistema.	44
Figura 12.	Área de desenvolvimento dos programas.	45
Figura 13.	Tela de históricos dos programas submetidos.	47
Figura 14.	Plataforma física do sistema.	47
Figura 15.	Arena.	48
Figura 16.	Peças.	49
Figura 17.	Micricontrolador.	50
Figura 18.	Braço.	50
Figura 19.	Equação base para os movimentos suaves dos motores.	51
Figura 20.	Algoritmo para o movimento suave dos motores.	51
Figura 21.	Câmera responsável pela filmagem das atividades.	52
Figura 22.	Repositório do GitHub.	54
Figura 23.	Recurso <i>Authentication</i> do Firebase.	55
Figura 24.	Estrutura de dados usadas no Firebase.	55
Figura 25.	Código responsável pela carga dos módulos da interface.	56
Figura 26.	Programação do Arduino.	57
Figura 27.	Código para gravação do vídeo.	58
Figura 28.	Código para envio do vídeo.	59
Figura 29.	Código para envio do vídeo.	59
Figura 30.	Escolaridade dos participantes dos testes.	61
Figura 31.	Conhecimento de programação dos usuários.	61
Figura 32.	Satisfação do usuário com a plataforma.	62

Figura 33.	Usuários que voltariam a utilizar a plataforma.	63
Figura 34.	Usuários que indicariam a plataforma.	63

LISTA DE TABELAS

Tabela 1.	Funções e Significados do NXT-Python.....	30
-----------	---	----

SUMÁRIO

1	INTRODUÇÃO	14
2	REFERENCIAL TEÓRICO	17
2.1	PENSAMENTO COMPUTACIONAL	17
2.2	INTRODUÇÃO A ROBÓTICA	19
2.2.1	Inteligência Artificial	20
2.2.2	Robótica em Nuvem	21
2.2.3	Robótica Educacional	22
2.2.4	Laboratórios Remotos na Educação	23
2.3	PENSAMENTO COMPUTACIONAL E ROBÓTICA EDUCACIONAL.	24
3	TRABALHOS RELACIONADOS	26
3.1	LABVAD: DESENHO E IMPLEMENTAÇÃO DO LABORATÓRIO VIRTUAL DE ATIVIDADES DIDÁTICAS COM ROBÓTICA	26
3.2	LERP: LABORATÓRIO DE ENSINO DE ROBÓTICA E PROGRAMAÇÃO	28
3.3	PROJETO SYROTEK	30
4	MATERIAIS E MÉTODOS	34
4.1	TECNOLOGIAS UTILIZADAS	34
4.1.1	Blockly	34
4.1.2	Framework JavaScript AngularJS	35
4.1.3	Firebase	36
4.1.4	NodeJS	36
4.1.5	Amazon S3	37
4.1.6	Ferramenta FFmpeg	39
4.1.7	Arduino	40
4.2	REQUISITOS DO SISTEMA	41
4.3	MÓDULOS DA PLATAFORMA	42
4.3.1	Módulo de interface web	42
4.3.1.1	Sistema de Login	43
4.3.1.2	Ajuda do sistema e desafios	44
4.3.1.3	Área de desenvolvimento e submissão dos programas	45
4.3.1.4	Visualização dos resultados	46

4.3.2	Módulo Físico	46
4.3.2.1	Arena	48
4.3.2.2	Braço Robótico	49
4.3.2.3	Programação Embarcada	49
4.3.3	Módulo de vídeo	52
4.3.4	Módulo de integração	53
4.4	IMPLEMENTAÇÃO DOS MÓDULOS	54
4.4.1	Módulo de Interface WEB	54
4.4.2	Módulo Físico	57
4.4.3	Módulo de Vídeo	58
4.4.4	Módulo de Integração	59
4.5	RESULTADOS E DISCUSSÃO	60
5	CONCLUSÃO	64
	REFERÊNCIAS	66

1. INTRODUÇÃO

Em 2006, Jeannette Wing [1] despertou a atenção da comunidade científica quando propôs a equiparação do pensamento computacional (do inglês, *computational thinking*) às habilidades básicas de leitura, escrita e aritmética [2]. Basicamente, Wing [1] descreveu o pensamento computacional (PC) como sendo a habilidade de empregar técnicas e conceitos da ciência da computação para resolver problemas do cotidiano, tais como colocar na mala os itens necessários para uma viagem (busca e cache), decidir entre comprar um carro e andar de táxi (algoritmo online) e refazer os passos para localizar um objeto perdido (*backtracking*).

A Computação está envolvida em quase todas as atividades humanas, das artes às tecnologias, e hoje, um cidadão ignorante do ponto de vista da Ciência da Computação poderá ter muitas dificuldades [3]. O desenvolvimento de habilidades computacionais na educação básica é necessário por promover a capacidade de resolução de problemas, apoiar e relacionar-se com outras ciências, promover múltiplos caminhos profissionais futuros, dentre outros aspectos [4]. Assim, diversas iniciativas têm sido realizadas em escolas brasileiras e do exterior visando a disseminação do pensamento computacional, habilidade que baseia-se em fundamentos da Ciência da Computação [5].

Nos dias atuais, inúmeros estudos têm se desenvolvido a fim de demonstrar as possibilidades e a necessidade de inserção de conceitos de Computação na educação básica. Estas iniciativas relatam experiências bem sucedidas do ensino de lógica de programação, estruturação de algoritmos e abstração de conceitos computacionais a alunos dos Ensinos Fundamental e Médio [6]. Por meio da exploração de ambientes de linguagem de programação e da exploração de jogos digitais, os alunos são estimulados à exploração do raciocínio lógico necessário para a construção de algoritmos [7].

Muito tem se discutido sobre a inserção de tecnologias e disseminação do pensamento computacional na educação brasileira. Para Almeida [8], o maior desafio ainda é universalizar o acesso às TIC para atingir todo o contingente de alunos brasileiros, docentes e estabelecimentos escolares e ampliar a compreensão de que o alicerce conceitual para o uso de tecnologias na educação é a sua integração ao currículo, ao ensino e à aprendizagem ativa, numa ótica de transformação da escola e da sala de aula em um espaço de experiência, de formação de cidadãos e de vivência democrática, ampliado pelo seu uso.

Pesquisas apontam, de forma geral, o uso de jogos digitais como um poderoso instrumento no processo de ensino-aprendizagem [9]. Mais especificamente, jogos que estimulam o pensamento computacional envolvem o desenvolvimento do raciocínio através de resolução de tarefas com o auxílio de simulações passo-a-passo e são geralmente baseados em lógica de programação, sem necessariamente envolver uma linguagem formal [10]. Nesse processo, o jogador-aprendiz torna-se um agente protagonista na produção de conhecimento através de investigação e exploração [9]. Além disso, os jogos podem se tornar elementos motivadores por oferecerem momentos lúdicos e interativos no processo de aprendizagem, sendo atrativos ao aluno e aumentando assim a disposição para aprender [10]. Entretanto, jogos digitais ainda são muito pouco utilizados em contextos educacionais [10].

Dois fatores importantes constituem-se como obstáculos para a sua adoção: dificuldade dos professores em encontrarem jogos que se integrem ao seu planejamento didático; e desconhecimento sobre como avaliar a qualidade dos jogos [11]. De fato, apesar de uma quantidade razoável de propostas para avaliar software educacional, ainda não existem métodos consolidados específicos para tal, e menos ainda para jogos digitais educacionais [12]. [13].

Neste cenário desafiador de disseminação do pensamento computacional é que a robótica educacional, também conhecida como robótica pedagógica, vai encontrando seu espaço, mostrando-se capaz de integrar estas tecnologias ao conteúdo pedagógico de forma lúdica e interdisciplinar [14].

Para Silva [15], atualmente existem diversos kits educacionais que podem auxiliar neste processo de introdução da robótica na educação. No entanto, apesar das vantagens e facilidades que estes kits apresentam, nem todas as escolas possuem recursos financeiros para adquiri-los. Mesmo quando estas conseguem adquirir, tanto o professor quanto o aluno podem ter interesse em possuir seus próprios kits, para realizar seus experimentos e/ou praticar em casa o que foi aprendido no ambiente escolar. Infelizmente, nem todos têm poder aquisitivo suficiente para isto [14].

Por outro lado existem tecnologias que já permitem o desenvolvimento de um ambiente interativo utilizando robótica remota que pode ser utilizado através da internet. Nesse sentido esse ambiente poderá ser utilizado por professores e alunos a partir de qualquer computador, seja na escola ou nas suas casas, basta o mesmo estar conectado a internet.

Com o advento da Internet das Coisas¹ o futuro próximo será altamente tecnológico e informatizado, a automação se fará cada vez mais presente e esses alunos dos dias de hoje poderão ser profissionais dessas áreas no futuro. O incentivo da prática de programação e o contato com a robótica pode ser um fator determinante para que esses alunos optem por ingressar em cursos voltados a computação ou áreas correlatas.

Dessa forma, o problema da presente pesquisa é responder a seguinte pergunta: É possível desenvolver um ambiente robótico interativo de aprendizagem utilizando robótica remota oferecendo aos estudantes e professores a oportunidade de trabalhar em um ambiente de robótica utilizando a Internet como um meio para se conectar a programas e robôs físicos e desenvolver atividades e/ou participar de competições facilitando o desenvolvimento do pensamento computacional?

O objetivo geral desse trabalho é construir uma plataforma que permita a programação remota de um braço robótico através da internet afim de que possa ser utilizada por alunos do ensino fundamental e médio, como meio para disseminação do pensamento computacional.

Afim de atingir o objetivo principal deste trabalho elaborou-se os seguintes objetivos específicos: Criar uma arena onde os robôs irão fazer as atividades; Projetar e criar o robô que será programado e utilizado na arena; Implementar a comunicação entre o robô e o sistema; Desenvolver um sistema web onde os usuários poderão testar seus códigos, remotamente, no robô; Apresentar um plano de atividades para testar a plataforma.

¹(Internet of Things) revolução tecnológica a fim de conectar dispositivos eletrônicos utilizados no dia-a-dia (como aparelhos eletrodomésticos, eletroportáteis, máquinas industriais, meios de transporte entre outros).

Assim espera-se que este ambiente permita que alunos aprendam conceitos de robótica e interajam de forma interativa com um robô, programando-o e compreendendo-o, sem ter a necessidade de possuir um dispositivo robótico. Dessa maneira, acredita-se que será possível ampliar a robótica educativa, especialmente a programação de robôs em situações onde não se teriam condições suficientes de se desenvolver tais atividades. Outro ponto a se destacar é que as dificuldades referentes a montagem do robô nessa solução podem ser abstraídas, não necessitando que o aluno domine, inicialmente, alguns conhecimentos preliminares em áreas como eletrônica e mecânica, por exemplo. Isso de certa forma é interessante, pois facilita a prática da programação de robôs, especialmente em escolas públicas e faz com que todos possam adquirir de forma indireta, porém interativa, novos conhecimentos, bem como a obtenção de benefícios cognitivos que a programação pode proporcionar.

A estrutura da dissertação está organizada em 5 capítulos. No capítulo 1 é feita a contextualização, motivação, bem como é feita a apresentação dos objetivos desse trabalho. No capítulo 2 apresenta-se uma revisão literária abordando conceitos de pensamento computacional, robótica e inteligência artificial. Também realiza-se uma abordagem da utilização desses conceitos aplicados a educação como robótica educacional e laboratórios remotos na educação. No capítulo 3 são apresentados trabalhos relacionados a robótica e laboratórios remotos voltados a educação que tem relação ao tema da dissertação. No capítulo 4 apresenta-se os materiais e métodos utilizados nesse trabalho. São descritas as tecnologias utilizadas para o desenvolvimento da plataforma e os requisitos do sistema. São também apresentados os módulos do sistema e como eles foram desenvolvidos. Finalmente são apresentados os resultados e discussões. No capítulo 5 apresenta as considerações finais e os indicativos de trabalhos futuros. Por fim, são apresentadas as referências bibliográficas que serviram de embasamento para desenvolver esse trabalho.

2. REFERENCIAL TEÓRICO

Este capítulo apresenta uma visão geral sobre o estado da arte dessa pesquisa especificando conceitos do pensamento computacional e da robótica bem com sua relação voltada a educação.

2.1 PENSAMENTO COMPUTACIONAL

Para Silva [16], o Pensamento Computacional é considerado uma competência universal, que deve ser adicionada à capacidade analítica de cada criança como um ingrediente vital de sua aprendizagem escolar. A combinação do pensamento crítico com os fundamentos da computação define uma metodologia para resolver problemas, denominada pensamento computacional [1].

Em março de 2006, Jeanette Wing publicou um importante artigo, Computational Thinking, no Journal of the Association for Computing Machinery. Wing afirma que "o pensamento computacional envolve a resolução de problemas, a concepção de sistemas e a compreensão do comportamento humano, baseando-se nos conceitos fundamentais da ciência da computação. Ele representa uma habilidade de aplicação universal para todos, e não apenas para os cientistas da computação". De modo geral, o artigo argumenta que esta nova competência deve ser adicionada à capacidade analítica de cada criança como um ingrediente vital da sua aprendizagem escolar [16].

Vários organismos profissionais e grupos de pesquisa, por exemplo, nos Estados Unidos, Reino Unido e Holanda, pediram mais atenção ao pensamento computacional no currículo. No EDUsumMIT 2013², destinada a avançar nesta discussão, discutindo sobre os principais componentes do pensamento computacional, sua importância e de outras competências do século 21, e seu lugar no currículo [16].

Empresas multinacionais apoiam a proposta do Pensamento Computacional e se empenham na disseminação de sua metodologia. Em 2007, a Microsoft e a Universidade de Carnegie Mellon criaram o Centro de Pensamento Computacional [17]. A Google, por sua vez, tem se empenhado em promover esta metodologia em todo o currículo do ensino primário e secundário nos Estados Unidos [18]. O Iraniano Hadi Partovi criou o code.org³ que tem o objetivo de estimular jovens a aprender os primeiros passos em programação. Ele afirma que todos os alunos, em todas as escolas, devem ter a oportunidade de aprender ciência da computação [19]. O programa conta com o apoio de grandes ícones da computação como Bill Gates da Microsoft e Mark Zuckerberg do Facebook, de políticos como Barack Obama bem como atletas e celebridades. O intuito não é criar pequenos gênios, mas sim usar a programação como pano de fundo para desenvolver habilidades como criatividade e raciocínio lógico em criança e adolescentes.

²EDUsumMIT é uma comunidade global de pesquisadores, formuladores de políticas e educadores empenhados em apoiar a integração efetiva da Tecnologia da Informação (TI) na educação, promovendo a disseminação e o uso da pesquisa. <http://www.curtin.edu.au/edusummit/>

³<https://code.org/>

Com base no princípio de que todos os alunos devem demonstrar competência nas habilidades básicas do Pensamento Computacional na conclusão do ensino médio, em 2010, as organizações CSTA (Computer Science Teachers Association), ISTE (International Society for Technology in Education) e a NSF (National Science Foundation) propuseram um conjunto de ferramentas, denominadas Computational Thinking toolkit [4], com o objetivo de desenvolver as habilidades do Pensamento Computacional na educação primária e secundária dos EUA. O material proposto pelas organizações citadas conta com um quadro de progressão, no qual nove conceitos inerentes à computação são eleitos como fundamentais para o desenvolvimento do Pensamento Computacional nas escolas [18]:

- Coleta de dados: é o processo de reunir dados de forma apropriada.
- Análise de dados: é o passo que objetiva tornar os dados coletados coerentes, encontrando padrões e tirando conclusões a partir destes dados.
- Representação de dados: é o processo de organizar apropriadamente as informações por meio de tabelas, gráficos, palavras, imagens ou qualquer outro recurso disponível.
- Decomposição de problemas: é a capacidade de divisão das tarefas em partes menores e manuseáveis.
- Abstração: é a redução da complexidade de um problema para focar na questão principal.
- Algoritmos e procedimentos: são definidos como uma possível série organizada de passos para resolver um problema ou atingir algum objetivo.
- Automação: é a utilização de computadores ou máquinas para fazer tarefas repetitivas ou tediosas.
- Simulação: é a representação ou a modelagem de um processo e a sua execução.
- Paralelismo: é a forma de organizar recursos para simultaneamente desenvolver tarefas que atinjam um objetivo em comum.

Para Andrade [18] o Pensamento Computacional é considerado como um processo de solução de problemas que inclui (mas não se limita a) características tais como: i) formular problemas de modo que seja possível usar um computador para ajudar a resolvê-los; ii) organizar os dados logicamente de modo que, futuramente, seja possível analisá-los; iii) representar dados através de abstrações, tais como modelos e simulações; iv) criar formas de automatizar as soluções através do pensamento algorítmico; v) permitir identificar, analisar e implementar soluções possíveis, com o objetivo de conseguir a combinação mais eficiente e eficaz de etapas e recursos; e, vi) generalizar e transferir este processo de solução de problemas a uma variedade de outros problemas.

O pensamento computacional, chamado por Wing [20] como uma atitude universal e um conjunto de habilidades que devem fazer parte do repertório de cada criança vem do campo da ciência da computação, mas é uma competência importante e influencia quase todas as disciplinas. Na

mesma linha, Bundy [21] relatou que a capacidade de pensar computacionalmente é essencial para a compreensão conceitual em todos os campos, através dos processos de resolução de problemas e pensamento algorítmico. O Pensamento computacional é visto como uma competência importante porque os alunos de hoje não só vão começar a trabalhar em campos relacionados a computação, mas também precisa lidar com a computação em sua vida cotidiana e na economia global de hoje [22]; [23]. Ao mesmo tempo, no entanto, os cientistas da computação discutem a necessidade de aumentar o interesse dos alunos pela ciência da computação, prestando atenção ao Pensamento Computacional (e não programação) no currículo obrigatório para oferecer aos alunos a opção de continuar seus estudos ou sua carreira em campos relacionados à Ciência da Computação [1]; [24].

2.2 INTRODUÇÃO A ROBÓTICA

A robótica é a ciência responsável por planejar e construir robôs, englobando várias áreas, como as engenharias mecânica, elétrica e eletrônica, incluindo também diversos ramos da física e da computação. Entretanto, a popularização da ficção científica influenciou o público leigo de tal forma que ele continua a imaginar o robô como um humanoide que pode falar, andar, ver e escutar e que se assemelha, se não a uma pessoa comum, a algo como robôs de filmes. É possível encontrar robôs de diversos tipos, cuja semelhança em nada tem a ver com a do homem, atuando em diferentes situações, desde pesquisas no fundo do mar a expedições espaciais a outros planetas do sistema solar [25].

Para Medeiros [26], existem muitas definições para um robô, entre as quais duas são ressaltadas:

1. Dispositivo ou máquina automática que realiza funções normalmente associadas a seres humanos. Essa definição é encontrada em dicionários de uso geral. Ela faz aparecer duas características que o senso comum atribui aos robôs: Autonomia - capacidade de executar suas tarefas sem interferência humana; Antropomorfismo - as capacidades e tarefas de um robô devem assemelhar-se a capacidades e tarefas que normalmente são atribuídas a seres humanos e não das máquinas.

Tal definição, embora muito próxima do conceito usual de robô, não é muito precisa, pois uma máquina de lavar roupas, por exemplo, poderia se enquadrar na definição.

2. Máquinas que, além de serem capazes de reproduzir tarefas e movimentos implícitos em sua construção, complementam a parte mecânica com dispositivos eletrônicos inteligentes de suporte, os quais dão certo grau de autonomia a esses engenhos. Nesta definição aparecem dois novos aspectos usualmente associados a robôs: Mobilidade - um robô deve ter a capacidade de executar algum movimento; Inteligência - o conceito de inteligência é tão ou mais difícil de definir que o termo robô. Usualmente, são adotadas definições subjetivas, que indicam como inteligente todo o comportamento que se assemelha a um comportamento eminentemente humano e não puramente animal.

Essa definição também é bastante abrangente, pois inclui certos engenhos que, embora apresentem muitos problemas similares aos robôs, usualmente não são considerados como tais, como por exemplo, um avião ou veículo com piloto automático.

Do ponto de vista da engenharia, a robótica é muito atraente, já que incorpora uma série de ciências básicas. Ela trabalha com movimentos, o que é de interesse da mecânica; com a eletrônica; e com a computação, que atua como o cérebro de todo esse sistema [25].

O que diferencia um robô de uma máquina automatizada, é que a máquina automatizada executa determinada tarefa de forma exclusiva, sendo muito eficiente e rápida neste sentido. Já um robô pode assumir várias funções em uma fábrica. Na linha de montagem, por exemplo, ele pode ser empregado em um processo de soldagem, ou, então, na parte de pintura. Ou seja, o robô pode ter várias tarefas, sendo programado para isso, enquanto que a máquina automatizada não, pois só executa um tipo de função. Há muitas pesquisas na área de robótica. Hoje em dia, fora do universo industrial, estão sendo desenvolvidos vários tipos de robôs, desde os humanoides aos que se assemelham aos animais, por exemplo. Existem também os robôs que em nada tem a ver com a figura humana, ou de animais, que são empregados para as mais diversas atividades, incluindo tarefas delicadas, como desativar bombas [25].

2.2.1 Inteligência Artificial

Para Luger [27], a Inteligência Artificial (IA) pode ser definida como o ramo da Ciência da Computação que se ocupa da automação do comportamento inteligente. Essa definição enfatiza que a Inteligência Artificial faz parte da Ciência da computação e que, desse modo, deve ser baseada em princípios teóricos e aplicados nesse campo. Esses princípios incluem as estruturas de dados usadas na representação do conhecimento, os algoritmos necessários para aplicar esse conhecimento e as linguagens e técnicas de programação usadas em sua implementação.

O campo da Inteligência Artificial (IA) está inserido no domínio mais amplo da Ciência da Computação e pode ser entendido como a ciência e técnica que visa a reproduzir artificialmente comportamentos considerados inteligentes. Dentre esses comportamentos incluem-se: raciocínio lógico-matemático, aprendizagem, linguagem, reconhecimento visual de formas e solução de problemas em geral. Dessa forma, há diversas subáreas que se encarregam de problemas específicos: representação do conhecimento, aprendizagem de máquina, processamento de linguagem natural, reconhecimento de padrões, resolução de problemas etc. Outra subárea importante refere-se à construção de sistemas especialistas, que simulam o desempenho de um especialista humano. Para concretizar essas soluções, a IA emprega diversas linguagens e ferramentas computacionais, algumas delas comuns a outras áreas da Ciência da Computação [28].

Entretanto, essa definição sofre com o fato de que a própria inteligência não é muito bem definida ou compreendida. Embora a maioria das pessoas esteja certa de que reconhece o comportamento inteligente quando o vê, não é certo que alguém possa chegar perto de definir a inteligência de

um modo que seria específico o suficiente para ajudar na avaliação de uma programa de computador supostamente inteligente, enquanto ainda captura a vitalidade e a complexidade da mente humana.

Para Luger [27], a definição inicial de inteligência artificial pode gerar várias interpretações. No entanto, essa dificuldade em chegar a uma definição exata de Inteligência Artificial é totalmente compreensível. A inteligência artificial ainda é uma disciplina jovem, e sua estrutura, suas considerações e seus métodos não estão definidos tão claramente quanto aqueles de uma ciência mais madura, como a física. A inteligência artificial sempre esteve mais preocupada com a expansão das capacidades da ciência da computação do que com a definição dos seus limites. Manter essa exploração baseada em princípios teóricos sólidos é um dos desafios que os pesquisadores de Inteligência Artificial, em geral enfrentam.

Imaginar uma máquina que possa decidir em vários aspectos a vida do homem, desde questões mais simples do cotidiano a problemas mais complexos, mesmo que ainda esteja distante da realidade, não é de hoje que a ciência da computação vem desenvolvendo maneiras de tornar um robô autônomo. Esses estudos compõem uma área específica da computação dedicada à Inteligência Artificial (IA), na qual o desafio está em criar na máquina dispositivos e sistemas que simulem a capacidade do homem de raciocinar, incluindo elementos de percepção que a levem a tomadas de decisões ou à resolução de problemas mais complexos.

O grande foco da área de Inteligência Artificial é o de, justamente, copiar algumas características humanas importantes, como a do aprendizado. Existem ferramentas computacionais que tentam copiar a capacidade de aprendizado do homem. Através de sensores, um robô pode começar a aprimorar uma determinada atividade, sendo programado para observar o ambiente e melhorar os seus procedimentos, explica [25].

Caurin [25] lembra que há métodos para fazer com que um robô pareça inteligente, e isso se dá por meio de tentativas de recriar o sistema nervoso do ser humano: as chamadas Redes Neurais Artificiais. Compostas por técnicas computacionais que apresentam um modelo matemático inspirado na estrutura neural de organismos inteligentes, uma grande Rede Neural Artificial, por exemplo, pode ter centenas, ou até milhares, de unidades de processamento. A partir daí, programam-se lógicas para que a máquina tome decisões baseadas nas informações que ela coleta do meio ambiente através dos seus sensores.

2.2.2 Robótica em Nuvem

A robótica em nuvem é um campo emergente da robótica baseada em computação na nuvem, armazenamento na nuvem e outras tecnologias de Internet centradas em volta dos benefícios de infraestrutura convergida em serviços compartilhados. Permite a robôs beneficiar-se da capacidade computacional, armazenamento e recursos de comunicações de centros de processamento de dados modernos. Além disso, retira custos gerais de manutenção e atualizações, e reduz a dependência de customização do middleware [29].

A robótica em nuvem permite a robôs aproveitar-se do aumento rápido das taxas de transferência de dados para desfazer-se de tarefas de grande processamento e sem exigências de processamento em tempo real. Isto tem um valor especial em robôs móveis, onde a computação on-board implica em exigências de poderes computacionais adicionais que podem reduzir a duração operacional e reprimir a mobilidade de robô bem como aumentar preços [30].

Outras definições mais abrangentes da Robótica em Nuvem podem incluir aspectos relacionados à Internet para robótica ou para robôs, como tendências em direção ao compartilhamento online de hardware e de software de fonte aberto, crowdsourcing em robótica, telepresença, e computação baseada no ser humano. Outras definições realçam as conexões entre robótica e campos emergentes relacionados como a Internet de Coisas, a Web de Coisas, redes de sensores, bigdata entre outros [30].

O carro autônomo do Google é um exemplo a ser considerado. Ele usa a rede para indexar mapas, imagens e dados sobre trajetórias prévias de condução, clima e tráfego para determinar a localização espacial e tomar decisões. Os dados de cada carro é compartilhada através da rede para otimização estatística e aprendizagem de máquina realizada por computação em paralela na nuvem [31].

James Kuffner do Google cunhou o termo "Cloud Robotics" em 2010. Robótica em nuvem e sistemas de automação pode ser amplamente definida como qualquer sistema de robô ou de automação que se baseia em dados ou código de uma rede para dar suporte ao seu funcionamento, ou seja, onde nem todos os sensores, computação, e memória são integrados num único sistema independente [31].

Há pelo menos quatro vantagens potenciais para usar a robótica na nuvem: 1) Big Data: o acesso a bibliotecas atualizadas de imagens, mapas e dados de objeto/produto, 2) Cloud Computing: o acesso à computação paralela sob demanda para análise estatística, aprendizagem, e planejamento de movimento, 3) Aprendizagem Coletiva: robôs e sistemas partilham trajetórias, políticas de controle, e os resultados, e 4) computação humana: uso de crowdsourcing para aproveitar as habilidades humanas para a análise de imagens e vídeo, classificação, aprendizagem e recuperação de erros [31].

A nuvem também pode fornecer acesso: a) conjuntos de dados, publicações, modelos, critérios de referência e ferramentas de simulação, b) competições livres de projetos e sistemas, e c) software de código aberto. É importante reconhecer que a robótica em nuvem e Automação levanta novas questões críticas relacionadas com a latência da rede, qualidade de serviço, privacidade e segurança [31].

2.2.3 Robótica Educacional

Para Silva [15], a robótica que visa o desenvolvimento de robôs para, de algum modo, auxiliar o homem em tarefas complexas e repetitivas. Sendo, portanto, uma área que agrega várias áreas do conhecimento, traz em si a interdisciplinaridade. Isso se mostra também na diversidade de setores em que robôs podem ser utilizados. Recentemente a robótica chegou a escola. Nesse âmbito, a robótica

toma uma nova forma, deixa de ser eminentemente para a produção de robôs para se constituir um novo mediador no processo ensino-aprendizagem [32].

Segundo Castilho [33], a robótica educacional, também conhecida como Robótica Pedagógica, é caracterizada por ambientes de aprendizagem onde o aluno pode montar e programar um robô ou sistema robotizado. Vai desde a simulação na tela do computador, como por exemplo, a implementação de um relógio digital ou contador que aparece na tela do computador e possui apenas sensores externos até meios físicos externos ao computador. Um robô inteligente com capacidade de decisão numa competição pode ser um projeto bastante estimulante ao aprendiz e é viável numa escola.

A Robotics Industries Association⁴ (RIA) define um robô como sendo um objeto reprogramável e multifuncional, pensado e projetado, visando auxiliar o ser humano para realizar algumas tarefas. Para DAbreu, Mirisola e Ramos [34] o desenvolvimento de atividades desta natureza envolve pelo menos a concepção, implementação, construção, automação e controle do dispositivo desenvolvido. A inclusão de atividades pedagógicas com robótica, em algumas escolas, vem se constituindo uma prática constante, na forma interdisciplinar de se promover o aprendizado de conceitos curriculares utilizando sistemas robóticos que integram o fazer como meio de aprender.

A fim de facilitar o processo de construção de robôs, principalmente para principiantes, existem diversos kits de robótica que são usados no âmbito educacional. A inserção de recursos tecnológicos no processo de ensino/aprendizagem é um grande desafio para o Brasil, pois uma parcela considerável dos estudantes não tem acesso a computadores ou jogos educativos, sejam esses na escola ou em sua residência [35].

Apesar de muitas pesquisas indicarem a Robótica Educacional como sendo uma ferramenta que envolve questões multidisciplinares, portanto rica pedagogicamente, ela, infelizmente não faz parte do cotidiano das escolas brasileiras. A explicação para tal fato, passa pela dificuldade na aquisição do equipamento. Essa dificuldade reside, principalmente, no momento de sua compra, pois seu custo, ainda ultrapassa as condições de aquisição [36].

De acordo com Trentin, Pérez e Teixeira [37] neste contexto de emergência da tecnologia na escola, a robótica apresenta-se como dispositivo potencializador da aprendizagem sendo considerada uma das dez áreas mais importantes de pesquisa no mundo, ainda pouco difundida no Brasil, principalmente em termos educacionais.

2.2.4 Laboratórios Remotos na Educação

O ensino de ciências e Robótica pode se apoiar em laboratórios virtuais ou remotos. Chao *et al* [38] explicam que, nos laboratórios virtuais, os alunos utilizam modelos baseados em dados matemáticos e estatísticos. Esses modelos aceitam a entrada e o ajuste de variáveis, sem o contato ou manipulação de dispositivos reais. Deste modo, os alunos visualizam e manipulam gráficos simulando diversas situações com o propósito de imitar um processo do mundo real.

⁴<http://www.robotics.org/>

Nos laboratórios remotos, os alunos controlam dispositivos reais operados por instruções fornecidas por software. Além disso, câmeras são utilizadas para monitorar o funcionamento de diversos dispositivos em tempo real. Portanto, em função da gama de recursos oferecidos pelos laboratórios remotos, podemos afirmar que estes estão mais próximos dos laboratórios físicos do que os laboratórios virtuais [39].

Silva *et al* [40] apontam que a experimentação remota, mesmo a distância, permite ao aluno interação prática com os mecanismos físicos. Os laboratórios remotos permitem uma imersão real, o que os diferenciam qualitativamente de simples simuladores ou laboratórios virtuais, que disponibilizam apenas experiências gravadas e resultados simulados.

Para Ertugrul [41], a principal vantagem no uso de laboratórios remotos na educação é o fato que, com esses laboratórios, os discentes passam a ter acesso a equipamentos que não teriam devido ao custo, tempo, distância e escassez.

Os laboratórios remotos representam a melhor alternativa para trabalhar com alunos, caso não haja disponibilidade de uso ou a existência de laboratórios físicos no ambiente escolar [42]. Além da experimentação remota, estes laboratórios são projetados para oferecer criação de grupos de discussão e fóruns, através de vídeo conferência; estímulo da aprendizagem colaborativa; aprendizado por intermédio de tentativa e erro; possibilidade de o aluno manipular e analisar dados reais dos experimentos realizados e flexibilidade na escolha da hora e local para a realização de experimentos [42].

2.3 PENSAMENTO COMPUTACIONAL E ROBÓTICA EDUCACIONAL.

Catlin [43] declarou que muitas escolas britânicas a muito tempo ensinam programação Logo permitindo que as crianças controlem uma tartaruga robô para efetuar as manobras mais complexas. No entanto, o ensino de Logo e controle de tartarugas nunca parou. Há mais de trinta anos a utilização de robôs educacionais, disfarçados como brinquedos ou tecnologia de controle programáveis, tem sido prática corrente nas escolas primárias do Reino Unido. Este trabalho não tem tido a oportunidade de publicizar, mas na prática tem acontecido nas salas de aula.

Desenvolver nos estudantes competências relacionadas ao pensamento computacional são benefícios que a robótica educacional pode proporcionar. Conforme afirma Almeida [44], a robótica pedagógica é um meio de instruir os estudantes sobre os conhecimentos da tecnologia atual e melhorar habilidades e competências tais como o trabalho de pesquisa, a capacidade crítica, o saber contornar as dificuldades na resolução de problemas e o desenvolvimento do raciocínio lógico.

Essas competências são importantes na formação do estudante. Complementando essa visão, Oliveira [45] *apud* Zilli [46] afirma que as competências as quais podem ser desenvolvidas por meio da robótica são: raciocínio lógico; habilidades manuais e estéticas; relações interpessoais e intrapessoais; utilização de conceitos 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.

A motivação em aprender é outro benefício apontado na robótica educacional, pois os alunos se tornam parte ativa da aula. Como muitos alunos desconhecem a robótica, o interesse em aprender sobre aquele objeto torna-se uma chance de dinamizar a aula e adquirir atenção dos aprendizes. Mesmo com a visão lúdica que a robótica pode ser vista, ela exige um grande empenho cognitivo em todas as fases do projeto, desde a construção do protótipo, programação das tarefas, testes, até a versão final do robô [45].

3. TRABALHOS RELACIONADOS

A utilização da robótica tem crescido muito, não só em função dos números de tecnologias disponíveis, o que tem causado uma diminuição no preço de se montar um laboratório de robótica, mas sobre tudo, pelas pesquisas feitas por universidades. Nesses sentido, alguns trabalhos de pesquisa tem sido desenvolvidos, com o objetivo de proporcionar acesso remoto a laboratórios. O uso de laboratórios remotos para a robótica educacional ainda é um campo vasto a ser explorado. Esta seção apresenta trabalhos que abordam estudos relacionados ao acesso remoto a robótica voltada para a educação.

3.1 LABVAD: DESENHO E IMPLEMENTAÇÃO DO LABORATÓRIO VIRTUAL DE ATIVIDADES DIDÁTICAS COM ROBÓTICA

O trabalho desenvolvido por Souza [39] na Universidade do Rio de Janeiro, propõe o LabVad⁵ que é o desenvolvimento e validação conceitual de um laboratório virtual de acesso remoto para atividades didáticas com Robótica. Trata-se de um ambiente que foi construído sobre a plataforma eletrônica de baixo custo do projeto Arduino conectada a alguns dispositivos eletrônicos, tais como displays, motores, sensores, LEDs e câmeras. O design da plataforma é completamente baseado em softwares livres, utilizando tecnologias como: PHP, HTML, CSS, JavaScript e MySQL. Esta plataforma consiste num ambiente online, de acesso livre, multiplataforma e compatível com a maioria dos navegadores, via internet.

O trabalho de Souza [39] no LabVad, o gene da sua construção surgiu devido ao alto custo dos kits didáticos de hardware disponíveis para o ensino de Robótica, bem como a carência de laboratórios equipados com tais Kits. Diante deste panorama, o uso de laboratórios remotos acessados via internet disponível para professores e alunos 24 horas, 7 dias por semana pode ser uma alternativa didática viável.

Segundo Souza [39] o LabVad procura atender quatro requisitos considerados importantes pela equipe que projetou:

- Similaridade com um ambiente real de RE - todo o ambiente de programação do LabVad deve ter correspondência nas funcionalidades existentes no ambiente local. Deve ser possível também ao usuário visualizar a execução dos seus experimentos no LabVad de forma muito semelhante ao que visualizaria, caso tivesse um hardware Arduino conectado ao seu computador. O hardware do LabVad deve ser semelhante a um Kit educacional de Robótica, do mesmo modo que a plataforma deve fornecer toda a funcionalidade da linguagem Wiring ao usuário.
- Requisito de portabilidade: Ambiente Multiplataforma - levado em consideração no desenvolvimento do LabVad foi a existência de diferentes sistemas operacionais disponíveis para alunos

⁵<http://labvad.nce.ufrj.br/labvad2/>

e professores nos diferentes equipamentos utilizados por eles. Para equacionar tal problema, foi implementado um ambiente multiplataforma que fosse executado na nuvem e que possa ser acessado de qualquer navegador web.

- Requisitos de eficiência e desempenho - As funcionalidades do sistema devem ser adequadas às limitadas especificações de hardware e periféricos do laptop Classmate, que é um laptop de configuração simples, comprado pelo Governo Federal e presente em boa parte das escolas.
- Ambiente Multiusuário - Ao se considerar que o LabVad é uma ferramenta educativa, foi determinado que o mesmo fosse multiusuário em todas as suas funções, exceto pela transmissão dos experimentos, porque as imagens são geradas por um servidor de transmissão de vídeo, contendo um IP fixo.

O LabVad é constituído por dois componentes distintos: um hardware e uma plataforma WEB que é controladora das funcionalidades desse hardware. O hardware é composto por circuito eletrônico acoplado a uma placa Arduino que é mostrado na figura 1.

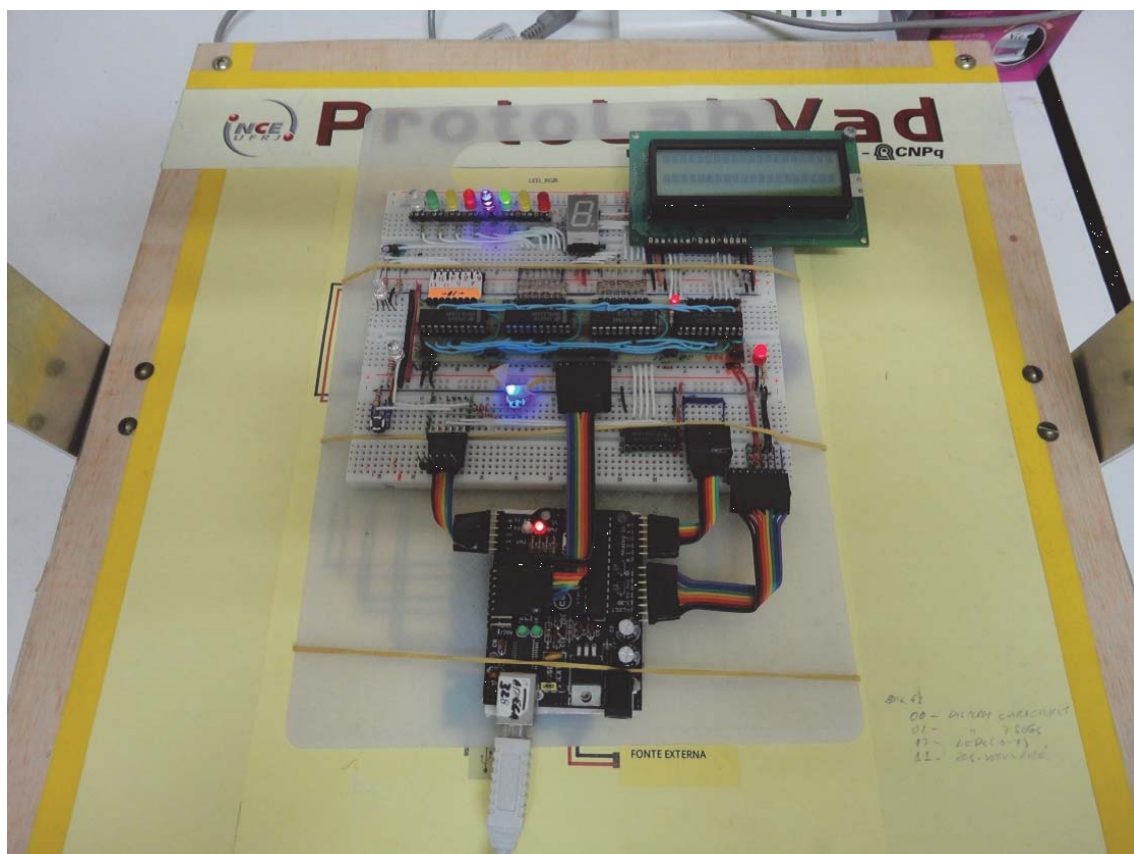


Figura 1. Parte física do LabVad [39].

A eletrônica cumpre o papel de permitir, de forma transparente para o usuário, a multiplexação de diferentes experimentos no Arduino, organizando-os em grupos e ampliando as possibilidades pedagógicas do projeto. A conexão entre os dispositivos periféricos e os pinos da placa Arduino já estão previamente definidos.

O funcionamento do hardware LabVad depende de um programa criado pelo usuário com instruções que permitam o funcionamento dos dispositivos eletromecânicos e/ou eletroeletrônicos (LED, RGB, servo-motor, display, etc.). A visualização da execução de um programa do usuário se efetua por intermédio da transmissão de imagens capturadas por uma câmera acoplada ao hardware.

O módulo de Experimentos tem vários itens, sendo que uma delas é a área de codificação (local onde os programas são escritos), muito similar a IDE nativa do Arduino, contendo destaque na sintaxe de comandos e controles interativos para as duas principais funções da linguagem Wiring: void setup() e void loop(). A área de código ainda oferece a numeração de linhas, recurso que a IDE oficial do Arduino não oferece ao usuário, como podemos ver na figura 2.

```

1 ▾ /* Este exemplo de código é de domínio publico
2   O nosso LED 13 é o último LED do canto esquerdo da tela do LabVad.
3   Vamos declarar este LED */
4   int led = 13;
5
6   // Na função setup escrevemos parte do código que será executado uma vez:
7 ▾ void setup() {
8     // Inicializando LED como saída.
9     pinMode(led, OUTPUT);
10  }
11
12 // O loop rodará parte do código até que o mesmo seja interrompido ou zerado.
13 ▾ void loop() {
14     digitalWrite(led, HIGH); //Acende o LED
15     delay(1000); // Espera um segundo. Para esperar meio segundo o valor atribuído seria 500
16     digitalWrite(led, LOW); // Apaga o LED
17     delay(1000); // Espera um segundo
18  }

```

Figura 2. Área de codificação [39].

3.2 LERP: LABORATÓRIO DE ENSINO DE ROBÓTICA E PROGRAMAÇÃO

Nesta seção apresenta-se o Laboratório de Ensino de Robótica e Programação -LERP⁶ baseado em Sistemas Multiagente (SMA) concebido por Almeida [47].

O LERP desenvolvido por Almeida [47], encontra-se disponível em um AVA acessado via Web, que permite que mesmo pessoas geograficamente distantes tenham experiência com programação utilizando robôs. O laboratório se caracteriza por possibilitar a aprendizagem colaborativa, com apoio de um SMA, para auxiliar no processo de aprendizagem de programação. Desta maneira, usuários que possuem acesso a laboratórios de informática, mas que não contém kits robóticos, podem usufruir de experiências desta natureza.

Para acesso aos experimentos robóticos do LERP, é necessário que o usuário esteja cadastrado no sistema. Para realização dos experimentos é necessário também um agendamento prévio tendo em vista a disponibilidade dos kits de robótica para uso. Neste ambiente os alunos têm desafios propostos a serem superados com o apoio de um SMA para auxiliar na realização dos mesmos. A programação dos robôs deve ser realizada na linguagem de programação NXT-Python [47]).

⁶<http://gaire.icomp.ufam.edu.br/ava/>

O SMA desenvolvido oferece um serviço de ajuda no cadastramento dos usuários, confirmação de acesso à sala virtual, realização de reservas, agendamento de experimentos, liberação e controle da seção na sala de desafios, e programação remota do robô. A escolha deste paradigma de programação, justifica-se pela natureza distribuída do LERP, e por este servir de apoio a problemas semelhantes e de mesmo grau de complexidade.

Para acesso à sala virtual do LERP e experimentos robóticos, é necessário que o usuário efetue seu cadastro. Feito o processo de cadastro o candidato deve se submeter a um agendamento no ambiente para realizar seus experimentos. No horário agendado o usuários tem acesso a sala de desafios que possibilita a realização de experimentos com o kit robótico Lego Mindstorms. O LERP possui cinco desafios a serem realizados pelos usuários, e podem ser acessados após a realização de um agendamento. Na figura 3 pode-se ver a sala de desafios da plataforma LERP [47].



Figura 3. A Sala de Desafios [47].

Na figura 4 tem um programa escrito em NXT-Python, o código que mantém acesa a luz azul do sensor de luz por 5 segundos e após esse tempo apaga, na tabela 1 mostra uma descrição das funções da linguagem. O servidor e o robô estão ligados por meio da tecnologia Bluetooth. Após o programa ser enviado pelo usuário para o servidor, o agente Robô busca o código na base de dados, gera o arquivo executável e faz a análise sintática do código, verificando se há erros ou não. Caso não haja nenhum erro sintático, o programa em execução enviado pelo LERP controlar o robô remotamente.

```

from nxt. blue sock import BlueSock
from funcoes import *

ID = ' 00:16:53:11:25:73'
sock = BlueSock (ID)

brick = sock.connect ( )

luz ( brick , " azul " )
tempo_de_espera ( 5 )
apaga_luz (brick )

sock.close ( )

```

Figura 4. Código NXT-python [47].

Tabela 1. Funções e Significados do NXT-Python [47].

Funções	Significado	Descrição
luz(brick, string_cor)	Acende sensor de luz.	O sensor de luz verifica a cor passada como parâmetro e acende a mesma. As cores podem ser: "vermelho", "azul", ou "verde".
apaga_luz (brick)	Desliga sensor de luz.	É possível desligar o sensor de luz utilizando a função apaga_luz e passando como argumento apenas o nome do brick que está sendo utilizado.
tempo_de_espera(tempo)	Tempo de espera.	A função de tempo de espera, tem como parâmetro apenas o argumento de tempo que deve ser informado em segundos. Essa função faz com que o robô espere por um tempo em segundos, até executar uma nova ação.
correr_por_tempo(brick, velocidade, tempo)	Percorre distância	A função correr_por_tempo percorre uma distância com base no tempo velocidade determinados. Também é necessário passar o nome do brick como argumento.
rotaciona(brick velocidade, numero_rotacao)	Rotaciona o robô.	A função rotaciona recebe três argumentos como parâmetro: o nome do brick, velocidade e número de rotação. Através desta função é possível rotacionar o robô.

3.3 PROJETO SYROTEK

Kulich *et al* [48] apresentam o SyRoTek⁷, uma plataforma de e-learning para robótica móvel, Inteligência Artificial (IA) e engenharia de controle. Esse ambiente gerencia e disponibiliza um conjunto de robôs móveis autônomos em uma arena, para que os usuários possam executar os experimentos. Os robôs podem ser controlados remotamente via tecnologia ActiveX, ou por um programa escrito em C++, Delphi ou Java.

O projeto SyRoTek de Kulich *et al* [48] consiste em um laboratório virtual de robótica no qual um conjunto de robôs móveis estão disponíveis para que alunos programem o controle dos robôs móveis autônomos colocados em uma área restrita com obstáculos dinamicamente reconfiguráveis.

⁷<https://syrotek.felk.cvut.cz/>

Porém a interface de programação exige instalação de ambientes de programação e utilização de linguagem de mais baixo nível.

A plataforma SyRoTek de e-learning para robótica móvel é uma abordagem moderna e eficaz para o treinamento em diversas áreas e em diferentes níveis de ensino. SyRoTek oferece acesso remoto a um conjunto de robôs móveis totalmente autônomos colocados em uma área limitada e com obstáculos que podem ser reconfigurados dinamicamente, que permite resolver uma enorme variedade de problemas. Um usuário é capaz de controlar os robôs em tempo real por seus próprios algoritmos desenvolvidos, bem como ser capaz de analisar os dados recolhidos e observar a atividade dos robôs por interfaces fornecidas. O sistema é atualmente utilizado para a educação na Universidade Técnica Checa, em Praga, República Checa, e na Universidade de Buenos Aires, Argentina, e tem acesso livre para outras instituições. Além da visão geral do sistema, este trabalho apresenta a experiência adquirida com a implementação real do sistema em atividades de ensino.

O projeto SyRoTek é um laboratório virtual de robótica no qual um conjunto de robôs móveis está disponível para os alunos criar um programa que irá controlar um robô móvel usando dados sensoriais reais. Na figura 5 podemos observar a arena com os robôs do projeto.



Figura 5. Arena com os robôs do projeto SyRoTek [49].

A visualização no SyRoTek compreende várias exibições reais para a arena e dados. Os dados são valores de sensores dos robôs como sonares, scanner a laser etc. Além disso, a posição do robô é fornecida pelo sistema de localização global baseada em processamento de imagem de uma câmera colocada no topo da arena. Os arquivos de vídeo e dados são apresentados de três

formas no SyRoTek. Na primeira forma, uma única imagem da cena real é fornecida através de uma página web. O segundo formato são arquivos de vídeo com dados sensoriais das cenas reais que são criados no servidor SyRoTek chamado de computador de controle. A principal vantagem desta forma de entregar os vídeos permite que os mesmos podem ser abertos por um leitor regular para mostrar os mesmos na estação de trabalho do usuário. O último tipo de apresentação é chamado de visualização de dados sensoriais em uma aplicação específica na qual os dados sensoriais podem ser combinados com arquivos de vídeos. O pedido dedicado pode ser usado para visualizar os robôs reais e dados sensoriais em um ambiente virtual que é vantajoso para uma conexão com baixa largura de banda para o servidor SyRoTek e para visualização online. Neste tipo de visualização, dados reais são apresentados de forma simulada ao ambiente e, portanto, este tipo de visualização é chamada de Realidade Mista.

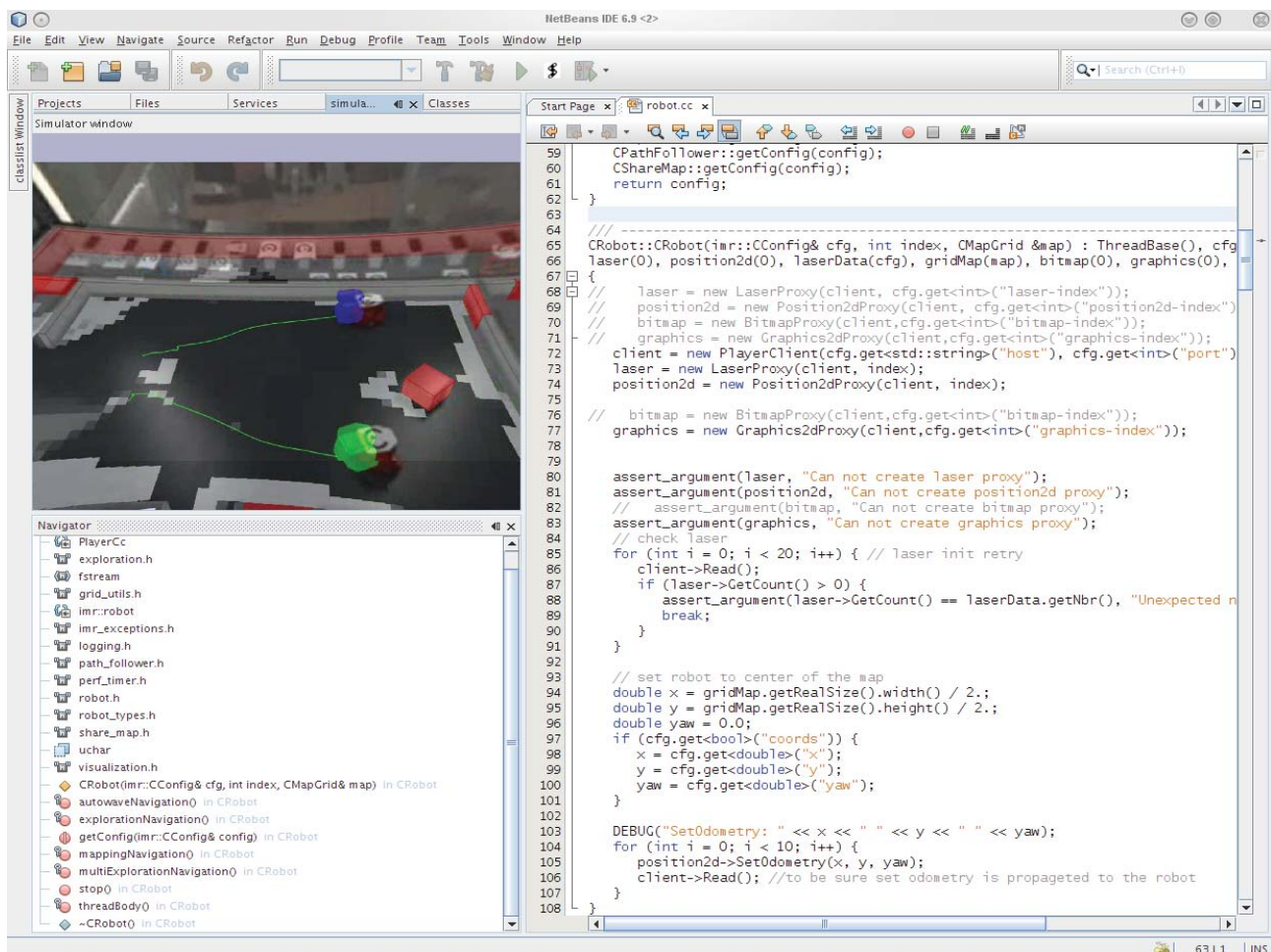


Figura 6. Ambiente de acesso e desenvolvimento do SyRoTek [49].

A visualização ao vivo é fornecida através de fluxos, enquanto os arquivos de dados de vídeo e sensores são fornecidos como arquivos para download e posterior visualização na máquina do cliente ou como fluxos para visualização sob demanda. O principal objetivo do sistema de visualização do SyRoTek é apoiar os usuários no desenvolvimento de aplicações que fazem parte de suas atribuições e também para apoiar a avaliação das aplicações pelos professores, que irão funcionar conforme segue. O aluno desenvolve um programa para controlar o comportamento de um robô autônomo, por

exemplo, desenvolver uma atividade de programação de um robô móvel para exploração de ambiente desconhecido. Então, no tempo livre, onde os alunos não usam ativamente os robôs na arena, o programa do aluno é executado e um vídeo com o comportamento desejado do robô é capturado. Todos os dados necessários são registrados e de acordo com a configuração desejada arquivos de vídeo com os resultados são criados. Nestes arquivos de vídeo, capturados com várias câmaras são combinadas com dados sensoriais e resultados específicos de aplicação do usuário para fornecer uma visão geral do desempenho do robô. Esses arquivos de vídeo são então entregues a um professor como um material de apoio para avaliação do desempenho do aluno. A figura 6 mostra o ambiente de desenvolvimento NetBeans⁸ utilizado para o projeto SyRoTek

O recurso necessário é uma combinação de várias capturas em uma única imagem, portanto, a linha de base de processamento de vídeo é composto por combinação de pontos de captura de várias câmeras em uma imagem que é então utilizada na imagem final. A visão da câmera pode ser vista em uma cena real ou visão virtual em um ambiente simulado. Esta linha de processamento é usado nos tipos de apresentações acima mencionadas, para on-line e apresentação de vídeo off-line, e na aplicação dedicada para visualização de dados sensoriais e vista para ambientes reais ou virtuais na estação de trabalho do usuário. A fim de combinar dados sensoriais e vídeos reais de uma forma significativa, as imagens das cenas reais têm de ser temporizadas para coincidir com a informação sensorial.

⁸<https://netbeans.org/>

4. MATERIAIS E MÉTODOS

Este capítulo apresenta os materiais e métodos utilizados para o desenvolvimento da plataforma robótica onde são apresentados os critérios e ferramentas utilizados. As próximas seções apresentam as principais tecnologias utilizadas para o desenvolvimento da plataforma, em seguida são apresentados alguns requisitos do sistema. Na seção seguinte são apresentadas as funcionalidades de cada módulo do sistema. Depois são apresentados detalhes da implementação da plataforma onde mostra tecnologias utilizadas em cada módulo do sistema. Por fim são apresentados os resultados e discussões relacionados aos testes da plataforma.

4.1 TECNOLOGIAS UTILIZADAS

Foram analisadas algumas tecnologias para implementação da plataforma. Após essas análises optou-se por tecnologias que permitissem simplicidade no desenvolvimento da plataforma, bem como fosse possível escalabilidade e ampliação do sistema sem muito esforço e problemas de integração dos módulos. As tecnologias listadas a seguir nos permitiram desenvolver um sistema com módulos fracamente acoplados sendo que a alteração em um módulo causa pouco ou nenhum impacto nos demais. Todas as tecnologias são *open source* ou com funcionalidades gratuitas que permitem a execução do projeto. Nesta seção é apresentado uma breve descrição e as funcionalidades das principais tecnologias utilizadas nesse trabalho. A seção 4.4 apresenta com mais detalhes a implementação de cada módulo e onde cada tecnologia foi utilizada.

4.1.1 Blockly

Blockly é um conjunto de bibliotecas destinado à construção de editores para programação visual. Sua aparência e funcionamento foram inspirados no ambiente Scratch⁹, desenvolvido com o propósito de ensinar programação para crianças, por meio da criação de animações e jogos interativos [50]. Assim como Scratch, a biblioteca Blockly oferece um conjunto de blocos que representam instruções de linguagens de programação como: declaração de variáveis; operações matemáticas, lógicas e relacionais; estruturas de controle (condicionais e de repetição) entre outras. Estes blocos podem ser empilhados formando pequenos programas, que ao serem executados pelo ambiente, são traduzidos em código nas linguagens JavaScript, Python ou Dart [51].

Como um ambiente de programação visual, Blockly aplica metáforas gráficas para aproximar conceitos de linguagens de programação a tarefas familiares aos usuários. Os programas são construídos encaixando blocos que representam trechos de código sintaticamente corretos. Dessa forma, o ambiente apoia o programador inexperiente em uma das suas principais dificuldades, os erros de sintaxe. Ele assegura, também, a construção de programas coerentes, limitando os encaixes possíveis

⁹<https://scratch.mit.edu/>

entre os blocos. Somente blocos graficamente compatíveis podem ser encaixados, como peças de quebra-cabeças, e estas limitações são estabelecidas durante a definição dos blocos, permitindo-se determinar até o tipo de dado que cada instrução deve receber [51].

A biblioteca também permite que novos blocos sejam adicionados, incorporando capacidades específicas do ambiente, e conferindo novas instruções para a construção de programas. A definição destes blocos acontece pela adição de dois arquivos ao diretório de blocos [52]. Estes descrevem, respectivamente, a aparência (conexões, compatibilidade e comportamento na interface) e o código que resulta da tradução do bloco durante a execução do programa. A Block Factory é uma página construída utilizando a biblioteca e disponibilizada junto desta, que oferece blocos para os aspectos e comportamento de um novo bloco [53]. De maneira que, mesmo sem conhecer os métodos da API, o desenvolvedor pode encaixar blocos que representam as características que deseja (cor, formato, comportamento, retorno, opções, campos de texto, blocos compatíveis, entre outros) e receber da página um modelo para a definição do novo bloco, que pode então ser customizado conforme as necessidades do ambiente que está sendo construído. A biblioteca Blockly é aplicada largamente em projetos de programação didática [51].

A biblioteca Blockly nesse projeto foi utilizada para criar o editor da interface web onde o usuário desenvolve seu programa. A opção por programação em blocos se deu em virtude do objetivo da plataforma abstrair a complexidade de uma linguagem de programação e fazer com que usuário fique com seu foco na lógica. Na seção 4.4.1 é apresentado o desenvolvimento da interface web e mostrado com mais detalhes a utilização dessas bibliotecas.

4.1.2 Framework JavaScript AngularJS

O AngularJS consiste em um framework JavaScript que implementa o padrão arquitetural MVC (Model-View-Controller), o qual separa unidades lógicas e responsabilidades no desenvolvimento de aplicações de grande porte [54]. Com isso, o framework facilita a criação rápida de aplicações que executam adequadamente em qualquer plataforma desktop ou móvel [55]. O AngularJS é considerado um framework robusto porque permite que o desenvolvedor foque no núcleo da aplicação e ele cuida de recursos como autenticação, persistência de dados e hospedagem da aplicação. Ainda, no lado do servidor, o framework permite ao desenvolvedor aplicar boas práticas de engenharia de software e no lado do cliente, o framework aumenta a produtividade dos desenvolvedores [56].

Em uma definição mais técnica, esse framework oferece uma estrutura consistente e escalável que facilita desenvolver aplicações complexas e de grande porte como parte de uma equipe [54]. Além disso, os templates no AngularJS são escritos utilizando HTML puro, o que auxilia os profissionais de design na tarefa de criar as páginas. Comumente, as aplicações desenvolvidas com AngularJS seguem a tendência do mercado no que se refere a SPA (Single Page Application, ou Aplicações de página única) principalmente por conta do uso intenso de requisições assíncronas possibilitados pela tecnologia Ajax [56].

O AngularJS é um framework de código aberto mantido pela empresa Google e atualmente está em sua versão 2.0. Vale salientar que o AngularJS passou rapidamente de um framework desconhecido para um dos mais populares e utilizados na atualidade. Devido a esta popularidade, o framework possui um rico conjunto de documentação de usuários e um ótimo suporte da comunidade [56].

Em nossa plataforma o AngularJS serviu como base para o desenvolvimento tanto da aplicação do lado do servidor como no lado da interface web. Por utilizar o conceito MVC facilitou a criação e integração das aplicações com as demais ferramentas.

4.1.3 Firebase

O Firebase é um BaaS (*backend as a service*) que oferece diversos serviços para facilitar o desenvolvimento de aplicações no lado do servidor. Com integração a várias plataformas como Angular, Javascript, Node.js, Android e IOS, o Firebase ajuda desenvolvedores a focar no desenvolvimento frontend mobile e web [57].

O núcleo do Firebase é um banco de dados NoSQL de tempo real que armazena os dados na nuvem. A manipulação dos dados do banco é feita através de uma API REST, mas todas as chamadas podem ser feitas através das bibliotecas específicas de cada linguagem, o que facilita bastante a utilização. O Firebase também disponibiliza a hospedagem dos arquivos da interface e também recursos para autenticação e identificação dos usuários utilizando suas redes sociais, como por exemplo Facebook e Google Plus [58].

A utilização do Firebase se justifica pois ele reduz a complexidade ou até elimina a necessidade de desenvolver uma aplicação *server side* específica; o tempo de resposta baixíssimo, o que torna o Firebase ideal para aplicações *real time* como chats e jogos multiplayer e permite um crescimento progressivo da aplicação [57].

A empresa Firebase foi fundada em 2011 por Andrew Lee e James Tamplin. Ela está localizada na cidade de São Francisco na Califórnia. Em outubro de 2014, a empresa foi comprada pelo Google [57].

Nesse projeto, o Firebase é responsável pela hospedagem do módulo de interface web e pelo armazenamento de todos dados. Como dados do sistema consideramos o cadastro dos usuários, os programas submetidos por esses usuários e pelo link para o vídeo gravados dos programas já executados.

4.1.4 NodeJS

O aparecimento de projetos como Node.js permitiram tornar o JavaScript uma linguagem possível de ser usada na implementação de componentes de um servidor de uma aplicação Web [59]. Deste modo, sendo o JavaScript uma das linguagens mais utilizadas na programação de aplicações

Web do lado do cliente, a possibilidade de implementação de servidor através de Node.js permite assim um maior potencial de integração entre os cliente e o servidor [60].

Node.js é uma plataforma de desenvolvimento de aplicações Web, desenvolvida utilizando o JavaScript Engine V8 do Google, com o objetivo de proporcionar uma forma fácil de construir aplicações rápidas e escaláveis [61]. Trata-se de um ambiente de desenvolvimento de JavaScript, desenvolvido por Ryan Dahl em 2009, majoritariamente implementada em C e C++, focando a performance e o baixo consumo de memória e visando suportar processos de servidor de longa duração [62].

Ao contrário do que acontece com outros ambientes de desenvolvimento, o Node.js não depende de multithreading para a execução de processos concorrentes na lógica de negócio do servidor. Na realidade, deve-se entender o Node.js como uma plataforma de desenvolvimento de servidores Web que possibilita a construção de sistemas altamente escaláveis, sem as complexidades de gestão de um sistema multithreading [59], operando apenas em single-thread, através de uma abordagem baseada em eventos e de input e output não bloqueante [63].

Por meio do gerenciador de pacotes Node Package Manager (NPM), algo similar ao gerenciador de pacotes APT-GET do Linux, é possível agregar diferentes funcionalidades ao programa desenvolvido, com por exemplo, acesso direto aos periféricos do sistema, drivers para conectividade com diferentes bancos de dados, frameworks para desenvolvimento, entre outros [61].

Dentro deste projeto o Node.js fica no servidor responsável por monitorar a fila de programas submetidos pelos usuários. O sistema verifica se tem algum programa na fila que fora submetidos pelos usuários, pega esse programa e passa as instruções para o Arduino acionar os mecanismos do braço robótico e executar as funções. Outra função executada no servidor é a gravação do vídeo e o envio do mesmo para a Amazon S3. As seções 4.4.3 e 4.4.4 descrevem a implementação dos módulos de vídeo e de integração e mostram com mais detalhes as funções do Node.js.

4.1.5 Amazon S3

Disponibiliza uma interface simples de serviço na Web que pode ser usada para armazenar e recuperar arquivos, a qualquer momento, de qualquer lugar na Web. Com o uso deste serviço, podemos facilmente enviar os vídeos gravados durante a execução dos programas e carregar os vídeos na interface Web.

Amazon Simple Storage Service (S3) é um armazenamento persistente baseado em nuvem. Ele opera independentemente de outros serviços da Amazon. As aplicações que você escreve para hospedar em seu próprio servidor podem utilizar o serviço da Amazon S3, sem necessidade estar na nuvem [64]

Segundo a própria AWS(Amazon Web Services) [65], o Amazon S3 fornece uma interface simples de serviço web que pode ser usada para armazenar e recuperar qualquer quantidade de dados, a qualquer momento, de qualquer lugar na web. A figura 7 mostra a interface do Amazon S3.

The screenshot shows the Amazon S3 console interface. At the top, there are navigation tabs for 'Services', 'Resource Groups', and a user profile 'Velcir Barcaroli'. Below this, the breadcrumb path is 'Amazon S3 > plataforma-robotica.s3'. There are four tabs: 'Objects', 'Properties', 'Permissions', and 'Management'. A search bar is present with the text 'Type a prefix and press Enter to search. Press ESC to clear.' Below the search bar are buttons for 'Upload', 'Create folder', and 'More'. The region is set to 'US East (N. Virginia)'. A table lists objects with columns for Name, Last modified, Size, and Storage class. The table shows a folder named 'videos' and four video files.

Name	Last modified	Size	Storage class
videos	--	--	--
-KVfVbL2a43iXuL6Z-zn.ogv	May 6, 2017 9:31:00 AM	960.2 KB	Standard
-KVf_Scf6qq13JF7gPcH.ogv	May 6, 2017 9:30:59 AM	857.8 KB	Standard
-KVgfqamr0qF_qIOIMJQ.ogv	May 6, 2017 9:31:01 AM	279.8 KB	Standard
-KiglUhG3vBSmi311VAQ.ogv	May 6, 2017 9:30:55 AM	2.1 MB	Standard

Figura 7. Interface do Amazon S3.

Veras [66] diz que o S3 fornece uma interface web service que pode ser usada para armazenar e recuperar qualquer quantidade de dados, a qualquer momento, de qualquer lugar na web. Pode-se gravar, ler e deletar objetos contendo de um byte até cinco terabytes de dados cada, e o número de objetos que você pode armazenar em um bucket¹⁰ do S3 é ilimitado. O S3 também é altamente escalável, permitindo o acesso simultâneo à leitura e à gravação dos dados por diferentes clientes ou threads de aplicativo.

Para acessar as APIs fornecidas pelo S3, o cliente pode utilizar qualquer linguagem de programação, já que o acesso é feito via SOAP ou REST [66].

As funcionalidades do S3, segundo o site da AWS(Amazon Web Services) [68], são as seguintes:

- Grave, leia e exclua objetos que contenham entre 1 byte e 5 terabytes de dados cada um. O número de objetos que você pode armazenar é ilimitado.
- Cada objeto é armazenado em um bucket e é recuperado através de uma chave específica que é atribuída a um desenvolvedor.
- Um bucket pode ser armazenado em uma das diversas regiões. Escolha a região em que deseja otimizar a latência, minimizar custos ou atender aos requisitos regulatórios.
- Os objetos armazenados em uma região nunca saem dela, exceto se você desejar transferi-los. Os objetos armazenados na região da UE (Irlanda), por exemplo, nunca saem da UE.

¹⁰Buckets são containers para objetos S3. Cada objeto armazenado no S3 está contido em um bucket, e ele funciona como um diretório em um sistema de arquivos. Uma das principais distinções entre uma pasta de arquivos e um bucket é que cada bucket e seu conteúdo podem ser acessados usando uma URLLIMA [67]

- Os mecanismos de autenticação são fornecidos para garantir que os dados permaneçam livres de acesso não autorizado. Os objetos podem ser públicos ou privados, e direitos podem ser atribuídos a usuários específicos.
- Existem opções para upload/download seguro de dados e criptografia de dados para proteção adicional dos dados.
- Utilize interfaces REST e SOAP com base padrão projetadas para trabalhar com qualquer ferramenta de desenvolvimento de Internet.
- Projetado para ser flexível para que camadas funcionais ou de protocolo possam ser facilmente adicionadas. O protocolo padrão de download é HTTP.
- Fornece funcionalidade para simplificar a capacidade de gerenciamento de dados durante sua vida útil. Inclui opções para separação de dados por buckets, monitoramento e controle de despesas e arquivamento automático de dados para opções de armazenamento de custo ainda mais baixo.

Os vídeos gravados durante a execução das atividades na arena são enviados para o Amazon S3, a recuperação desses vídeos pode ser feita diretamente por uma URL fornecida pela Amazon, sendo que essa URL é colocada na interface do usuário permitindo que o mesmo visualize suas atividades executadas a qualquer momento.

4.1.6 Ferramenta FFmpeg

FFmpeg é uma solução completa de multimídia, capaz de decodificar, codificar, transcodificar, transmitir, filtrar e gravar praticamente todos os tipos de vídeos criados. Ele suporta os formatos de vídeo mais antigos até os mais recentes e modernos. Não importa se esses vídeos foram projetados por algum comitê de padrões, a comunidade ou uma corporação. Também é altamente portátil: o FFmpeg compila, executa e transfere vídeos nas plataformas Linux, Mac OS X, Microsoft Windows, BSDs, Solaris, etc., em uma ampla variedade de ambientes de compilação, arquiteturas de máquinas e configurações [69].

O projeto FFmpeg tenta oferecer a melhor solução tecnicamente possível para desenvolvedores de aplicativos e usuários finais. Para conseguir isso, são combinadas as melhores opções de software livre disponíveis. Existe um certo favorecimento no desenvolvimento próprio para manter uma baixa dependência de outras bibliotecas e para maximizar o compartilhamento de código entre as diversas funcionalidades da FFmpeg [69].

O projeto é formado por diversos componentes:

- ffmpeg: Linha de comando para converter arquivos multimídia entre vários formatos diferentes.
- ffmpeg-server: É um multimídia streaming server para transmissões ao vivo via HTTP.

- `ffplay`: É um simples media player com biblioteca FFmpeg.
- `libavcodec`: É uma biblioteca que contém decoders e encoders para codec de áudio e vídeo.
- `libavformat`: É uma biblioteca que contém demultiplexador e multiplexador para formatos diversos formatos multimedia.
- `libavutil`: É uma biblioteca de ajuda que contém rotinas comuns para diferentes partes do FFmpeg.
- `libpostpro`: É uma biblioteca que contém rotina de processamento pós vídeo.
- `libswscale`: É uma biblioteca que contém rotina de escala de imagem.
- `libavfilter`: Permite o vídeo ser modificado ou examinado entre o decoder e o encoder.

A seção 4.4.3 mostra com mais detalhes a utilização da FFmpeg. Em nossa plataforma a ferramenta FFmpeg tem a função de capturar o vídeo no momento em que o programa esta sendo executado e o braço robótico trabalhando. Esse vídeo é gravado em um arquivo no próprio servidor. Em seguida o programa administrador que está no servidor envia esse vídeo para o Amazon S3.

4.1.7 Arduino

O projeto Arduino foi criado na Itália em 2005 com o objetivo de oferecer uma plataforma de prototipagem eletrônica de baixo custo e de fácil manuseio por qualquer pessoa interessada em criar projetos com objetos e ambientes interativos [70]. A plataforma Arduino é composta de uma placa eletrônica (hardware) e de um ambiente de desenvolvimento (software) para criação dos projetos pelos usuários. O Arduino é um projeto open source onde a documentação para elaboração do hardware (placa eletrônica) e o código fonte do ambiente de desenvolvimento estão disponíveis para os usuários.

Segundo David Mellis [71] eles tinham o objetivo de que outras pessoas estendessem a plataforma para adequá-la às suas necessidades. Para isso, elas deveriam ter acesso ao código-fonte do software e ao projeto do hardware. Além disso, como era uma plataforma nova, ser de código aberto deu confiança às pessoas. Elas sabiam que poderiam continuar expandindo a plataforma mesmo que o desenvolvedor original desistisse dela.

A placa eletrônica do Arduino contém várias entradas e saídas, analógicas e digitais, além de interface serial via conexão USB para comunicação com o computador. O elemento inteligente desta placa é um microcontrolador ¹¹ da família AVR que permite milhares gravações e regravações em sua memória de programa. Existem diversos modelos oficiais e não oficiais da placa eletrônica (hardware) do Arduino [72].

Em função da característica open source do projeto Arduino, muitos outros modelos da placa eletrônica surgiram, desenvolvidos pela comunidade Arduino em todas as partes do mundo. Esses

¹¹Microcontrolador circuito integrado que contém todas as funções básicas de um computador (CPU, memória RAM, ROM, dispositivos de entrada e saída).

modelos não oficiais também estão listados no site do projeto. Entre eles destacamos as versões norteamericanas Boarduino e Roboduino, a canadense Freeduino e as brasileiras, Severino e Brasuino [70].

Com o objetivo de aumentar as funcionalidades da placa Arduino, várias empresas de hardware desenvolveram placas eletrônicas adicionais para conexão nos terminais do Arduino. Estas placas eletrônicas adicionais são denominadas Shields e acrescentam várias funções específicas ao Arduino, desde controle sobre de motores até sistemas de rede sem fio [72].

Para Pinto [72] a plataforma Arduino tem se popularizado ao redor do mundo em uma infinidade de aplicações, possibilitando que pessoas não especialistas em eletrônica e programação possam colocar em prática suas idéias de interação com objetos e ambientes fazendo uso de recursos da eletrônica e da programação. Dentro do conceito de Hardware Livre, o projeto Arduino possibilita que qualquer pessoa (mesmo com conhecimentos mínimos de eletrônica) possa confeccionar a placa eletrônica.

Em nossa plataforma o Arduino é responsável única e exclusivamente pelo acionamento dos motores responsáveis da movimentação do braço robótico. O Arduino fica conectado fisicamente com o servidor por um cabo USB.

4.2 REQUISITOS DO SISTEMA

Nesta seção serão apresentados os procedimentos realizados para o levantamento dos requisitos da plataforma robótica. Esta etapa levou em conta a pesquisa realizada nos capítulos anteriores considerando principalmente os trabalhos relacionados.

Buscou-se criar um sistema modular onde tivesse um acoplamento fraco entre os módulos, essa fraca dependência tem por objetivo permitir agregar novas funcionalidades no ambiente sem muita interferência nos módulos. Para atingir esse objetivo foi dividido o sistema em quatro módulos onde o primeiro compreende os mecanismos físicos e microcontroladores responsáveis pelo controle dos robôs, o segundo é a interface com o usuário, também chamada de interface web, que pode ser acessada por qualquer computador conectado a internet, o terceiro módulo é responsável por pegar os programas submetidos na interface e fazer sua execução no ambiente físico e por fim um módulo de imagem responsável pela captura das imagens das atividades executadas pelo robô e disponibilizá-las aos usuários.

Na figura 8 é possível observar o delineamento esquemático do projeto, onde o usuário acessa uma interface web, faz sua programação e a submete ao sistema, que analisará os comandos a fim de verificar, por motivos de segurança, se os mesmos não infringem alguns limites existentes no conjunto arena e robô e posteriormente repassar os comandos ao dispositivo robótico. Ao iniciar a execução do programa, o sistema de imagem inicia a captura das imagens e ao final envia uma URL ao usuário para que o mesmo possa conferir se a sua programação atingiu o objetivo previamente proposto.

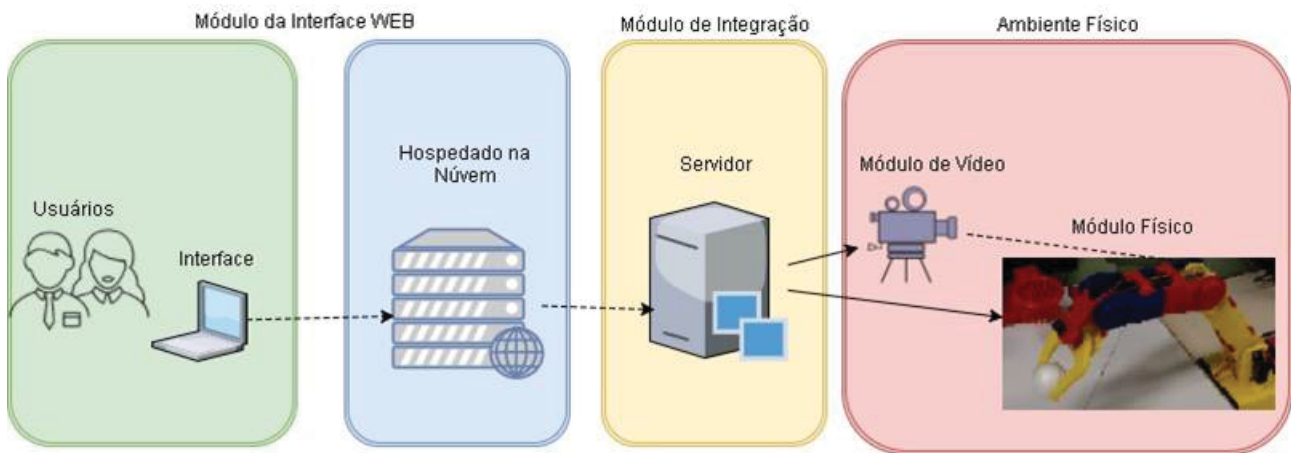


Figura 8. Delineamento esquemático do projeto.

4.3 MÓDULOS DA PLATAFORMA

A plataforma foi dividida em módulos para facilitar a implementação e a integração entre as diversas tecnologias. Foram criados quatro módulos, sendo que o primeiro é o módulo de interface web, e é responsável pela interface com o usuário e armazenamento dos dados dos usuários como autenticação, programas submetidos e resultados. O Módulo físico corresponde a estrutura física do robô que compreende o braço robótico, arena, microcontrolador e programação embarcada. O terceiro módulo, definido como módulo de vídeo tem a função da gravação e envio dos vídeos para posterior reprodução na interface web. O Módulo de integração tem a função de integrar e coordenar os demais módulos a fim de tornar funcional a plataforma. A seguir serão descritos as funcionalidades de cada módulo, suas divisões e sua relação com os demais dentro da plataforma.

4.3.1 Módulo de interface web

A interface Web é uma aplicação que foi desenvolvida com Angular.js e Google Blockly. A interface com o usuário disponibiliza uma programação visual similar ao Scratch¹², onde o foco principal do usuário é nas funcionalidades do braço robótico. A interface foi projetada para ser intuitiva e com instruções claras de como utilizar o sistema. Os vídeos gerados durante a execução do programa são disponibilizadas em local visível dentro da interface. O acesso ao sistema pode ser feito utilizando contas do Facebook ou Google. Os programas submetidos pelos usuários deverão ficar em uma fila aguardando disponibilidade do módulo físico para a execução. Essa fila corresponde em guardar o programa no Firebase com um ID sequencial e o módulo integrador é responsável por executar o primeiro programa e marcar como já executado. A seguir serão descritos detalhadamente como o usuário deve utilizar o sistema envolvendo desde o processo de login, ajuda, desenvolvimento e submissão do programa bem como a visualização do resultado. A plataforma pode ser acessada na URL <https://plataforma-robotica-7d010.firebaseio.com>

¹²<https://scratch.mit.edu>

4.3.1.1 Sistema de Login

Ao acessar a interface web o usuário deverá se identificar. Para simplificar o processo foram utilizadas APIs do Google e do Facebook que permitem o acesso ao sistema com as credenciais desses sistemas, evitando a necessidade de criação de um cadastro para essa finalidade. Na figura 9 podemos ver que o sistema dá liberdade de escolher uma conta da rede social preferida do usuário para acessar a plataforma.



Figura 9. Tela de login.

Após logar-se no sistema, como podemos observar na figura 10, o usuário terá a sua disposição no lado esquerdo as "funções" que poderão ser usadas no desenvolvimento das tarefas, no centro a área de trabalho onde o código é montado pelo usuário e na direita as instruções de como utilizar o sistema e os desafios que poderão ser submetidos.

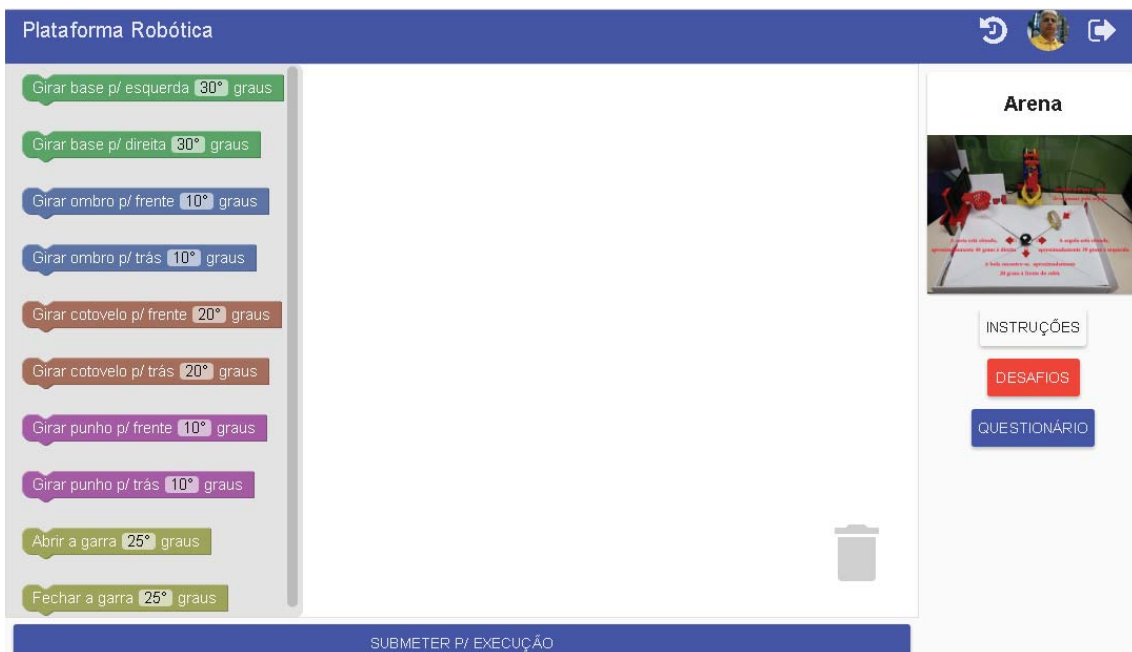


Figura 10. Tela de inicial do sistema.

4.3.1.2 Ajuda do sistema e desafios

Para tornar simples a utilização da plataforma foi desenvolvido um sistema de ajuda que se encontra no lado direito da tela conforme visto na figura 10.

Na seção anterior quando descrevermos a arena podemos observar na figura 14 que são informadas aos usuários algumas instruções de posicionamento para facilitar o desenvolvimento das tarefas. Nas instruções esta descrito que a cesta se encontra aproximadamente a 40 graus a direita do centro da arena onde se encontra a bola, o posicionamento da bola esta á aproximadamente 20 graus a frente do braço robótico e que a argola esta posicionada 30 graus a esquerda. Essas instruções podem ser obtidas ao clicar na imagem principal de ajuda, onde a mesma será ampliada no centro da tela.

Ao clicar no botão *instruções* o usuário terá acesso uma série de animações que simulam todos os movimentos possíveis do braço robótico. Essas simulações foram criadas para evitar interpretações equivocadas dos movimentos do braço. Por exemplo o movimento de girar a direita, pode ser entendido como a direita do usuário ou a direita do braço. Na figura 11 pode-se observar os botões para acionar os recursos bem como a tela de instruções aberta.

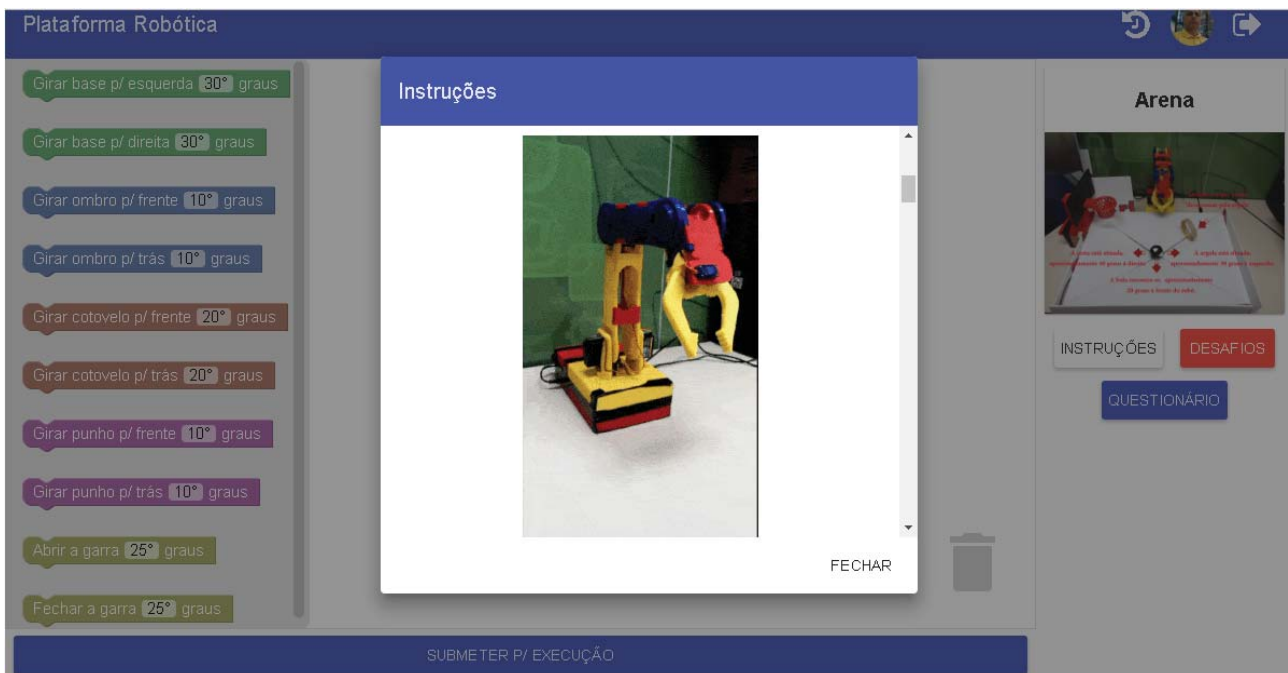


Figura 11. Tela de inicial do sistema.

Outro recurso do módulo de ajuda são instruções sobre os desafios que o aluno pode submeter. O sistema permite a execução de dois desafios conforme segue:

1. Desafio da Cesta de Basquete: o usuário é convidado a desenvolver um código para que o braço robótico movimente-se do seu ponto inicial e vá até o encontro da bola, posicionada no centro da arena e leve a mesma até uma cesta de basquete, que encontra-se ao lado direito do robô no cenário.

2. Desafio da Argola: o usuário é convidado a desenvolver uma sequência de códigos para que o braço recolha a bola do seu ponto inicial e passe a mesma por dentro de um argola no sentido "fora para dentro", ou seja que a bola passe por entre a argola e vá ate o seu ponto de origem. A argola está situada ao lado esquerdo do robô.

Nesses desafios o usuário deverá desenvolver uma sequencia de códigos que realize cada um independente ou também os dois desafios podem ser executados em uma única programação.

4.3.1.3 Área de desenvolvimento e submissão dos programas

A construção do programa ocorre por meio de uma linguagem de programação visual, através do agrupamento de blocos lógicos. Foi desenvolvida com uma interface amigável e atrativa, tornando possível a programação por pessoas leigas em programação de computadores, diferente das outras linguagens de programação que exigem conhecimento específico dessa ciência. Por exemplo, a linguagem de programação Java.

Como pode ser observado na figura 12, o usuário deve arrastar os blocos da esquerda da tela agrupando-os no centro formando uma sequência de instruções necessárias para a execução das tarefas. A simplicidade da programação é perceptível pelo pequena variedade de instruções que o usuário pode utilizar.

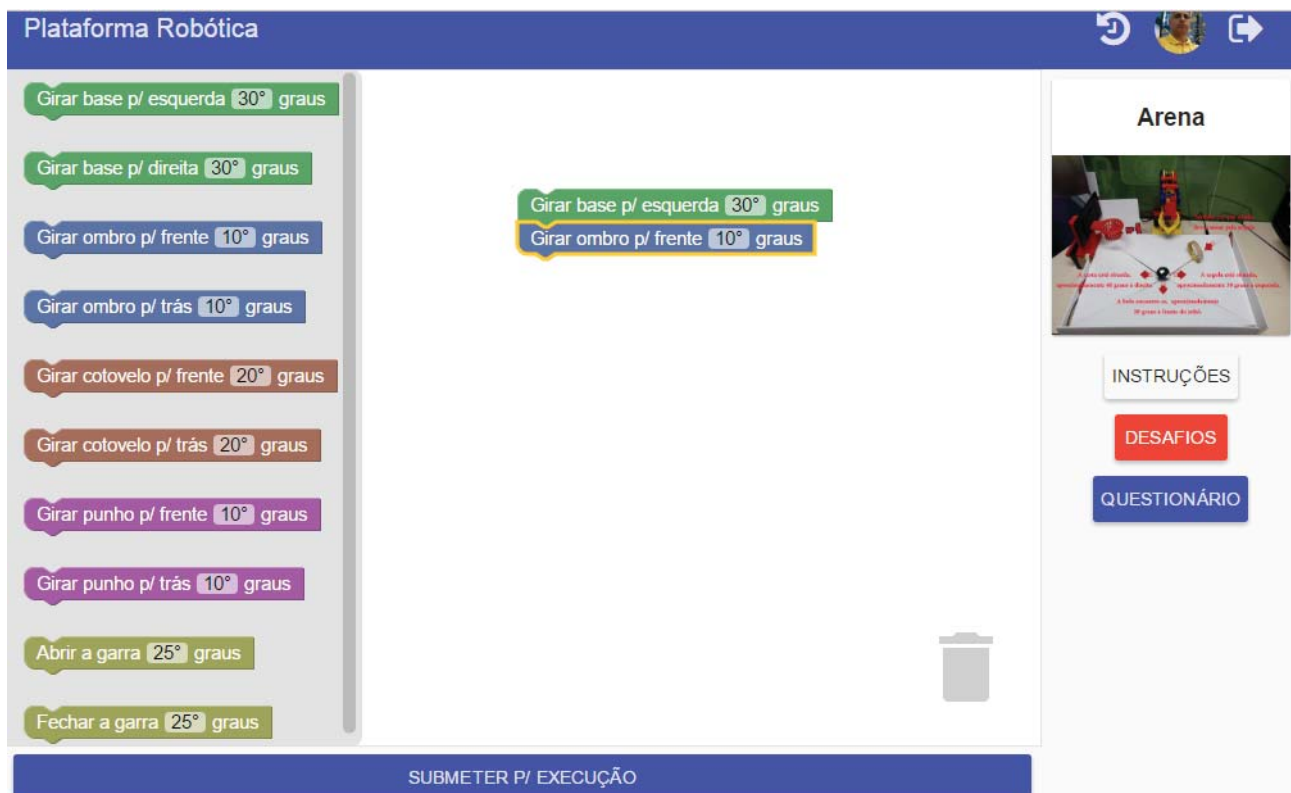


Figura 12. Área de desenvolvimento dos programas.

Com a combinação de cerca de dez instruções o usuário pode executar as tarefas propostas pelo ambiente. As instruções são agrupadas a cada duas permitindo ações contrárias uma da outra, como frente e traz, cima e baixo, abre e fecha. A seguir estão os grupos de instruções:

- Girar a base para a direita e para a esquerda: Essas instruções tem a função de girar a base do braço robótico em n graus permitindo posicionar o mesmo em determinada área da arena.
- Girar ombro para frente e para trás: Instruções que fazem a junta da base chamada de ombro inclinar-se para frente ou retornar, permitindo que a pinça possa se aproximar do ponto desejado na arena.
- Girar cotovelo para cima e para baixo: Ao acionar segunda junta do braço, chamada de cotovelo, faz com que a pinça se aproxime ou distancie da base da arena. Esse acionamento pode ser programado usando as instruções de girar cotovelo.
- Girar punho para frente e para trás: Ao se aproximar do alvo, ainda pode ser necessário alguns ajustes finos, esses ajustes podem ser conseguidos coma o utilização dessas instruções que fazem com que a terceira junta chamada de punho possa ser movimentada.
- Abrir e fechar a pinça: Instruções que programam a pinça para pegar ou soltar a bolinha que esta na arena.

O programa desenvolvido na tela principal no momento da sua submissão é validado com alguns testes, para evitar que o robô se choque com algum obstáculo e acabe se danificando. Se o programa for válido ele é salvo na base de dados do Firebase e colocado em uma fila para ser executado no sistema físico do robô.

4.3.1.4 Visualização dos resultados

A interface com o usuário ainda disponibiliza uma área onde se tem todos os históricos dos programas submetidos. O histórico possui uma listagem com os programas enviados recentemente, mostrando alguns dados referente a validação e execução de cada programa. No histórico tem também um link para visualização do vídeo da execução das atividades pelo robô. Esse vídeo fica armazenado no amazon S3. Na figura 13 podemos observar na direita os vídeos das execuções dos programas. No lado esquerdo pode-se ver a lista de programas submetidos bem como um detalhamento desses programas. Esses vídeos podem ser visualizados a qualquer momento.

4.3.2 Módulo Físico

Após várias discussões e considerações, definiu-se que a plataforma física não poderia ter uma complexidade muito alta no seu desenvolvimento em virtude do tempo disponível para a execução do projeto já que não se pretende nesse trabalho criar atividades muito complexas, pois o objetivo



Figura 13. Tela de históricos dos programas submetidos.

principal é o desenvolvimento de uma plataforma que permita a programação remota de robôs. Uma vez a plataforma desenvolvida, a complexidade e diversidade dessas atividades podem ser ampliadas.

A escolha por um braço robótico se deu devido a praticidade que o mesmo proporciona. Por ser um dispositivo de base fixa, o braço é a solução ideal para essa plataforma, a qual necessita que o dispositivo, ao estar disponível, esteja posicionado sempre em sua posição inicial para receber uma nova sequência de comandos, além de não necessitar de um transmissor sem fio na comunicação entre o servidor e o dispositivo robótico. Na figura 14 podemos observar a estrutura física e a seguir serão apresentados os materiais e métodos utilizados na confecção do robô.

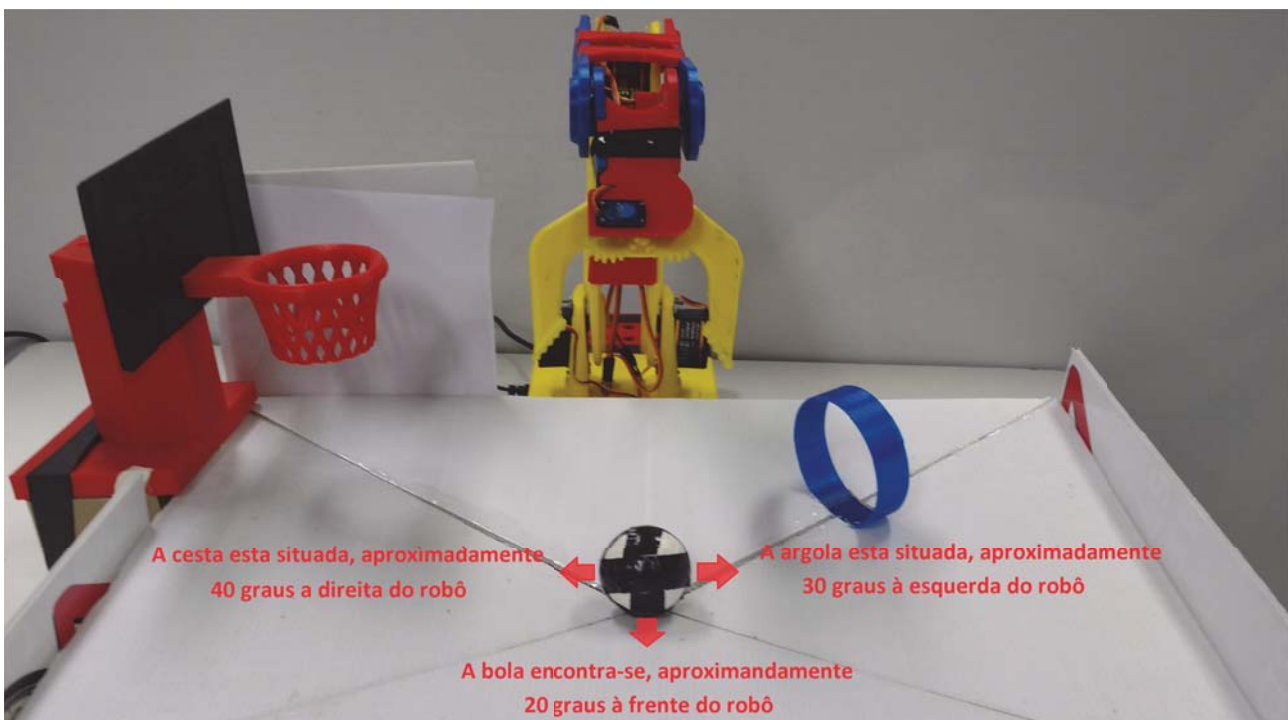


Figura 14. Plataforma física do sistema.

4.3.2.1 Arena

Como cenário da plataforma, desenvolveu-se uma arena com (40 x 50) cm de dimensão e em um formato conhecido por duas águas com 4 caimentos. Esse formato permite, graças às elevações laterais, que a bola que deverá ser capturada pelo braço robótico, ao final de cada execução permaneça sempre no seu ponto de origem, ou seja, no centro da arena. Dessa forma, a arena, sem nenhum auxílio humano, estará sempre disponível ao próximo usuário que vir a utilizar o aparato robótico.

Dois desafios estão presentes nessa arena: o desafio do basquete e o desafio da argola. No primeiro, o usuário deve desenvolver um código para que o braço robótico movimente-se do seu ponto inicial e vá até o encontro da bola, posicionada no centro da arena, e leve a mesma até uma cesta de basquete, que possui 18 cm de altura, 6,5 cm de diâmetro, também modelada e impressa na impressora 3D, e que encontra-se ao lado direito do robô no cenário. Semelhante ao desafio do basquete, no desafio da argola o usuário deve novamente codificar no ambiente web uma sequência de códigos para que o braço recolha a bola do seu ponto inicial. Porém, neste, ao invés de acertar a cesta, o usuário para completar o desafio deve fazer com que a bola recolhida pelo robô passe por dentro de um argola, de 7,5 cm de diâmetro, posicionada no seu lado esquerdo. A figura 15 mostra a arena, seus desafios e o robô em atuação.

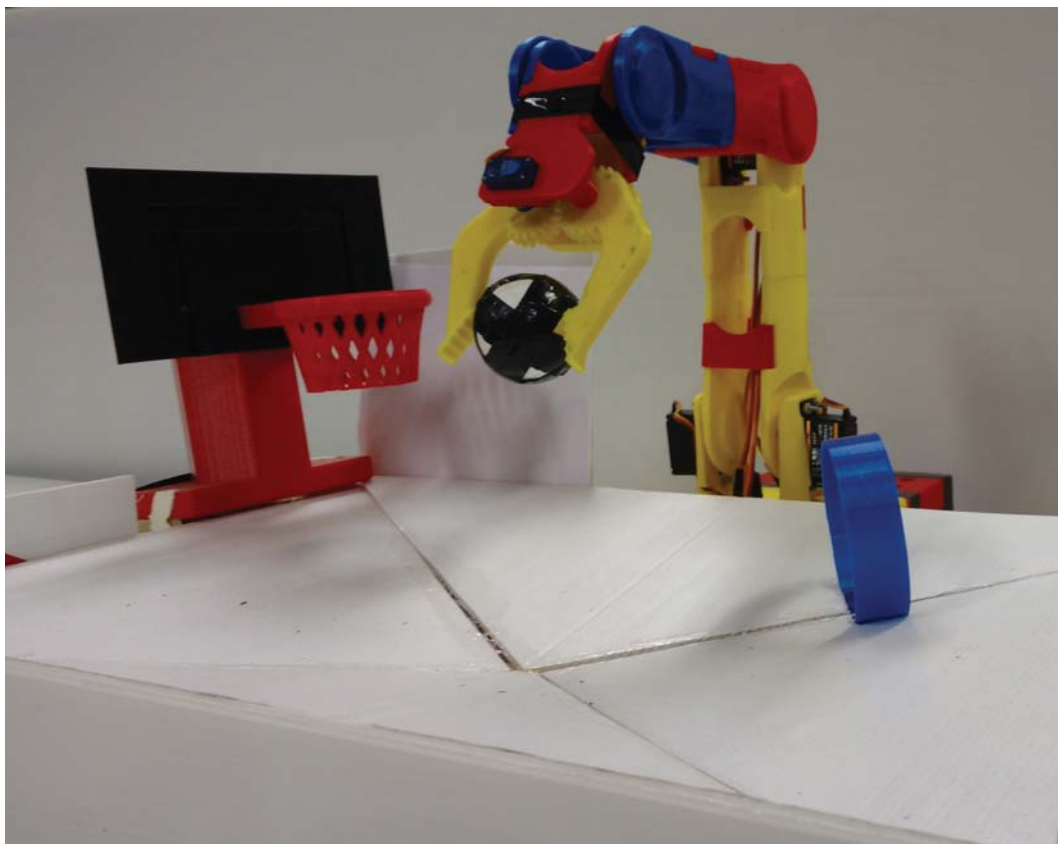


Figura 15. Plataforma em funcionamento.

4.3.2.2 Braço Robótico

O robô criado para essa pesquisa possui 57 cm de altura e 46 cm de alcance em sua extensão máxima. A seguir serão apresentados maiores detalhes sobre o mesmo:

- A estrutura do braço robótico é composta de vinte e seis peças, conforme figura 16, as quais foram em sua totalidade impressas em uma impressora 3D utilizando filamento 100 PLA. O modelo utilizado foi adaptado de um projeto disponível no repositório Instructable¹³; Na figura 16 são apresentadas as peças utilizadas na fabricação do braço robótico.
- Quatro servos-motores Emax Es3001 de 4,2 kg de torque, operando a uma tensão de 5V de tensão;
- Dois micro servos-motores Tower Pro de 1,5 kg de toque, operando também, em 5V de tensão;
- Micro controlador Arduino Nano baseado no ATmega328 32 KB.
- Fonte de alimentação com 5V de tensão e 2A de corrente;
- Conectores.

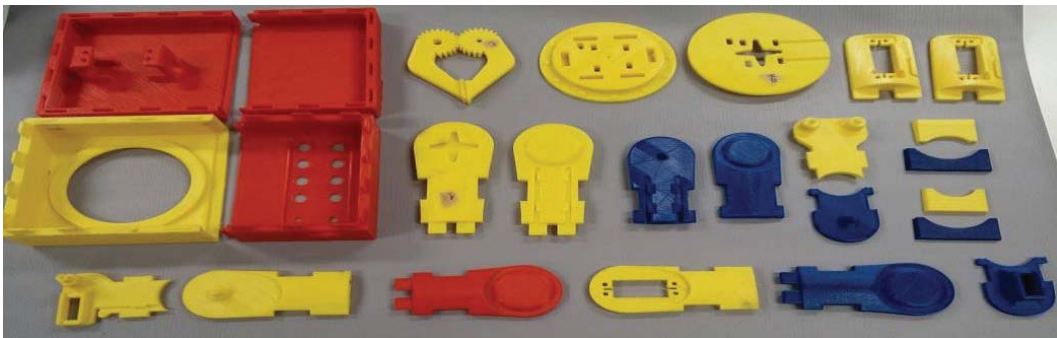


Figura 16. Peças que compõem o braço robótico.

Na figura 17 é mostrado o esquemático utilizado para conectar os servos-motores ao Arduino. Nesse esquemático são apresentadas as conexão de dois servos-motores. Os demais servos-motores seguem as mesmas conexões sendo que cada um tem a conexão de controle em um pino diferente do Arduino.

4.3.2.3 Programação Embarcada

O firmware embarcado no aparato foi desenvolvido na linguagem C++ para o controle de periféricos robóticos, a qual é comumente utilizada na programação da placa de prototipagem Arduino.

¹³Projeto disponível sobre a licença creative commons (BY-NC- SA), disponível em <http://www.instructables.com/id/3D-Printed-Robot-Arm/>

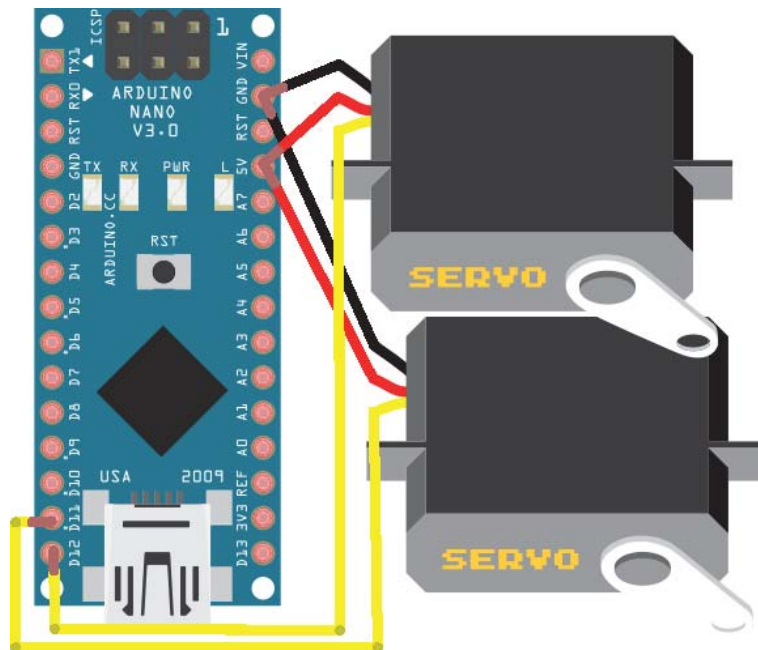


Figura 17. Esquemático do micro controlador.

Para compilação e carga do algoritmo na placa controladora do braço robótico utilizou-se o ambiente de desenvolvimento (IDE) Arduino IDE 1.6.9. Esse código embarcado é responsável por receber as instruções enviadas pelo usuário via interface web e executá-las no robô, realizando assim os movimentos solicitados.



Figura 18. Braço robótico e comandos presentes na interface web.

Dividido em cinco módulos denominados: garra, pulso, cotovelo, ombro e base, o braço recebe da interface web, por intermédio de um servidor, uma sequência de comandos que dita quais módulos movimentar, em qual sequência e qual a quantidade de graus que cada módulo deve mover a cada movimento. A figura 18 demonstra a interação entre interface web, seus comandos e os módulos existentes no aparato robótico.

Com o intuito de deixar os movimentos mais suaves e ao mesmo tempo menos robotizados, desenvolveu-se com base na cinemática inversa uma função capaz de acionar os motores, a fim de movimentar os módulos do dispositivo robótico da sua posição inicial até uma posição meta em um determinado tempo, gerando um valor variável de pausa a cada ângulo movimentado, tomando-se como base os conceitos robóticos de geração de trajetórias [73]. A equação e o algoritmo gerado, são descritos nas figuras 19 e 20.

$$Ax^3 + Bx^2 + Cx + D = 0 \text{ onde:}$$

$$A = \frac{-2\Delta p}{tf^3}; B = \frac{3\Delta p}{tf^2}; C = 0; D = PosInicial - PosAtual ;$$

sendo que:

x é tempo total em segundos para a realização do movimento;

tf é a posição meta desejada;

Δp a diferença angular a ser percorrida;

$PosAtual$ será incrementada gradativamente a cada intervalo.

Figura 19. Equação base para os movimentos suaves dos motores.

O intervalo necessário para a movimentação do próximo grau de cada servo-motor é determinado pela diferença do último intervalo de tempo (iniciando em 0) com a terceira raiz da equação acima. Dessa forma, o algoritmo implementado pode ser descrito conforme a figura 20.

ultimo = 0

Para i de PosInicial+1 at PosFinal passo 1

MoverMotor i

$$raiz = \frac{B}{3A} + \frac{2}{3} + \frac{B}{A} \cdot \cos\left(\frac{\arccos\left(\frac{-27DA^2}{2B^3} - 1\right) + \pi}{3}\right) ;$$

intervalo = *raiz* - *ultimo*

Espere intervalo segundos

ultimo = *raiz*

Fim_para

Figura 20. Algoritmo para o movimento suave dos motores.

4.3.3 Módulo de vídeo

Como a arena onde as atividades executadas pelos robôs está localizada remotamente dos seus usuários, faz-se necessário mostrar para o usuário o comportamento dos robôs na execução das suas tarefas. Para essa finalidade criou-se um sistema que é responsável pela captura das imagens feitas durante a execução do programa pelo robô e as disponibiliza para quem submeteu a atividade na plataforma. Esse módulo é composto por uma câmera que fica localizada junto a arena para a obtenção das imagens. Essa câmera fica conectada fisicamente no computador onde está rodando o módulo servidor. O módulo servidor que é responsável por colocar em execução o programa submetido pelos usuários, também é responsável por acionar a ferramenta FFmpeg que faz a captura do vídeo durante a execução do programa e o salva no servidor. Na figura 21 podemos ver a câmera responsável pela captura dos vídeos durante a execução dos programas.



Figura 21. Câmera responsável pela filmagem das atividades.

Finalizada a execução do programa e feita a gravação do vídeo, o mesmo é enviado para o Amazon S3. Como comentado nas seções anteriores, o Amazon S3 disponibiliza uma URL que permite esse vídeo ser visualizado por um browser, pelo usuário que submeteu o código. Essa URL é disponibilizada no módulo de Interface Web onde o usuário poderá visualizar esse vídeo a qualquer momento.

4.3.4 Módulo de integração

O módulo integrador, como o próprio nome diz, tem a finalidade de integrar e gerenciar os demais módulos. Esse módulo roda em um servidor que está localizado próximo a arena, pois a câmera que faz o vídeo bem como o Arduino responsável pelo acionamento dos motores que controlam o braço robótico estão conectados fisicamente nesse servidor.

Esse módulo tem quatro funções principais que deve executar seguindo uma certa sincronia. A primeira função é monitorar a fila de programas submetidos pelos usuários na interface web. O programa servidor fica rodando de forma contínua buscando no banco de dados hospedado na plataforma Firebase por um novo programa para ser executado. Na existência de um programa o mesmo é copiado para o servidor para posteriormente submeter o mesmo a execução.

A segunda função desempenhada por esse módulo é o envio da sequência de comandos do programa do usuário para o Arduino que controla o braço robótico. O módulo faz uma validação da programação recebida, verificando se a mesma está dentro dos limites existentes tanto por parte do robô quanto da arena. Essa verificação é de suma importância para aplicação, pois a mesma garante integridade do robô, impedindo que alguma sequência de comandos danifique o mesmo como, por exemplo, forçar um braço do robô a exceder o seu deslocamento para além da arena. Ao encontrar algum erro ou comando inválido o módulo de verificação impede que a programação chegue até o robô e retorna à aplicação web informando os motivos pelos quais a programação não foi executada. Se a programação passar pela validação, a sequência de comandos será enviada para o Arduino via porta serial para ser executada.

Outra função desse módulo é a gravação do vídeo durante a execução das atividades do robô. Antes de iniciar o envio dos comandos para o robô, o sistema aciona a ferramenta FFmpeg responsável pela gravação do vídeo. A gravação fica ativa até o fim da execução do programa. O vídeo gerado é gravado no servidor.

A quarta função é o envio do vídeo e a gravação de dados de encerramento na plataforma web. Após a execução do programa e geração do vídeo, o sistema envia o arquivo do vídeo gerado para o Amazon S3 e captura a URL gerada. Em seguida faz o fechamento do programa na plataforma web, gravando nela a URL para a visualização do vídeo e definindo o mesmo como executado.

Após o término da execução das funções relacionadas a um programa que foi submetido, o sistema volta a checar novamente o módulo web para ver se tem novos programas na fila para execução.

A tecnologia escolhida para o desenvolvimento do programa responsável por administrar as execuções foi NodeJs por se tratar de um interpretador de código JavaScript que funciona do lado do servidor, e que permite a criação de aplicações de alta escalabilidade, com códigos capazes de manipular muitas conexões simultâneas, em uma única máquina física. A implementação desse módulo esta descrita na seção 4.4 e a figura 29 mostra parte do código responsável por essas funcionalidades.

4.4 IMPLEMENTAÇÃO DOS MÓDULOS

Nesta seção serão descritos os procedimentos utilizados na implementação da plataforma detalhando cada módulo separadamente e o processo de integração entre os módulos. São apresentados detalhes da implementação e trechos de códigos com o objetivo de facilitar o entendimento do processo de construção da plataforma. Todo o código utilizado para o desenvolvimento da plataforma pode ser encontrado no repositório do GitHub¹⁴. A figura 22 mostra a tela principal do projeto onde pode ser observado três pastas principais onde a primeira chamada "administrador" tem os códigos do módulo integrador e de vídeo. Na segunda "arduino" tem os programas que são embarcados na plataforma física. Na pasta "interface" são encontrados todos os programas que compõem a interface web.

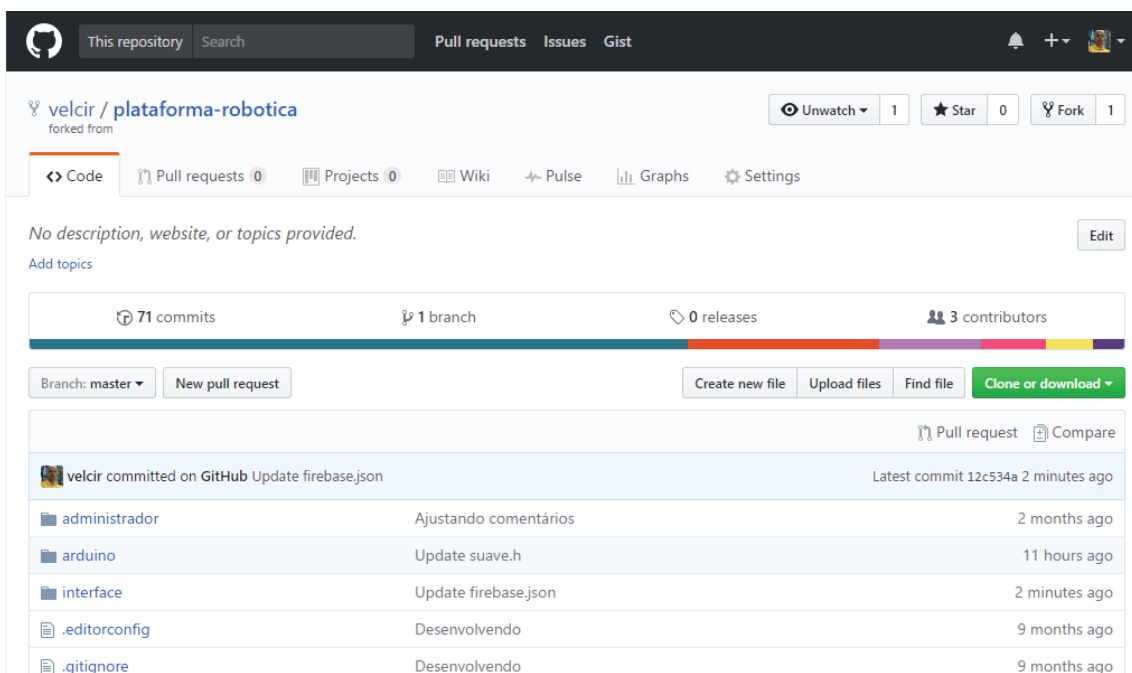


Figura 22. Repositório do GitHub.

4.4.1 Módulo de Interface WEB

Como descrito anteriormente o Firebase é uma plataforma com vários recursos. Nessa plataforma utilizamos três recursos. O primeiro recurso utilizado é o de *Authentication*. O Firebase permite várias formas de autenticação. Nesse projeto optou-se pela autenticação com contas do Gmail ou Facebook como mostrado na figura 23.

Outro recurso utilizado é o *Realtime Database* onde ficam armazenados os dados da plataforma no formato JSON¹⁵. Na figura 24 podemos ver as três estruturas de dados utilizadas. O *histórico*,

¹⁴<https://github.com/velcir/plataforma-robotica>

¹⁵JavaScript Object Notation - Notação de Objetos JavaScript. <http://www.json.org/json-pt.html>

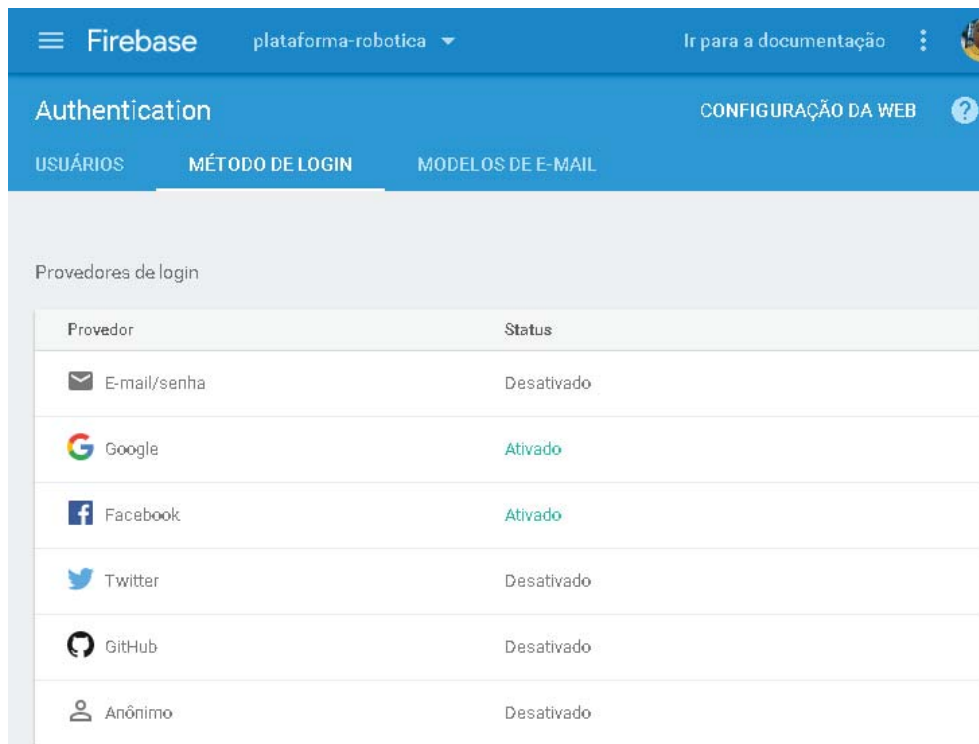


Figura 23. Recurso *Authentication* do Firebase.

onde ficam registrados os programas executados, os *programas* que são os programas que estão na fila para ser executados e o *usuários* que representa a lista de usuários do sistema.

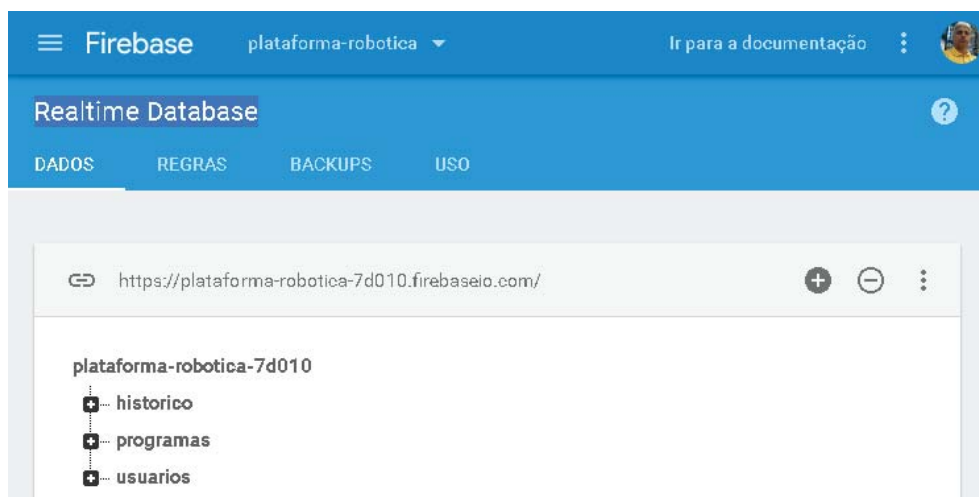


Figura 24. Estrutura de dados usadas no Firebase.

Ainda dentro do firebase utilizamos o serviço de *Hosting*. Esse serviço permite a hospedagem da aplicação a nível de produção.

O desenvolvimento da interface web teve como framework o Angular JS e a linguagem foi o TypeScript¹⁶. A aplicação é composta de três telas principais, a tela de login, a tela de histórico e a

¹⁶<https://www.typescriptlang.org/>

tela de edição do programa. Na tela de edição de programa fazem parte também os recursos de ajuda do sistema.

Na figura 25 é apresentado trecho do código responsável por carregar todos os módulos necessários. Na linha 1 é importado os recursos do framework angular, na linha 3 a importação das bibliotecas do firebase. A biblioteca blockly é carregada na linha 32. Existem também três componentes principais do sistema que correspondem as três telas da interface. Esses componentes são o editor carregado na linha 29, o componente histórico carregado na linha 27 e por fim o componente login carregado na linha 26. O componente app, carregado na linha 25 é responsável por manter a seção do usuário por toda aplicação.

```

1. import * as angular from 'angular';
2.
3. import * as firebase from 'firebase';
4. (<any>window).firebase = firebase;
5.
6. import 'angular-material';
7. import 'angularfire';
8. import 'angular-ui-router';
9.
10. import './index.scss';
11.
12. const ngModule = angular.module('app', [
13.   'ngMaterial',
14.   'firebase',
15.   'ui.router'
16. ]);
17. import routesConfig from './config/routes.config';
18. import themingConfig from './config/theme.config';
19. import blocklyConfig from './config/editor.config';
20.
21. import Firebase from './common/services/firebase.service';
22. import Usuario, runOnRouteError from './common/services/usuario.service';
23. import Editor from './common/services/editor.service';
24.
25. import app from './components/app/app.component';
26. import login from './components/login/login.component';
27. import historico from './components/historico/historico.component';
28.
29. import editor from './components/editor/editor.component';
30. import sidenav from './components/editor/sidenav/sidenav.component';
32. import blockly from './components/editor/blockly/blockly.component';

```

Figura 25. Código responsável pela carga dos módulos da interface.

4.4.2 Módulo Físico

A implementação do módulo físico ficou restrita a programação do Arduino. Na figura 26 mostra o código embarcado no Arduino.

```

1.  include <Servo.h>
2.  include "suave.h"
3.  void setup() {
4.    Serial.begin(9600);
5.    garra.conecta(2,50);
6.    pulso.conecta(4,70);
7.    cotovelo.conecta(6,15);
8.    ombroI.conecta(8,90);
9.    ombroII.conecta(10,180 - ombroI.read());
10.  base.conecta(12,97);
11.  }
12.  int ler (Suave servo){
13.    return servo.read();
14.  }
15.  char aguardarValorChar()
16.    while (Serial.available() <= 0)
17.      delay(1);
18.    return Serial.read();
19.  }
20.  void loop() {
21.    porta = aguardarValorInt(); // Lê byte do buffer serial;
22.    switch (porta) {
23.      case 2: //garra
24.        sentido = aguardarValorChar();
25.        graus = aguardarValorInt();
26.        if (sentido == '-') graus = graus * (-1);
27.        if( graus + ler(garra) < 0 || graus + ler(garra) > 180)
28.          break;
29.        garra.mover(graus + ler(garra),10);
30.        break;

```

Figura 26. Programação do Arduino.

Nesse trecho de código pode-se ver na linha 1 é feita a carga da biblioteca responsável para o controle dos motores e na linha 2 é carregado a biblioteca "suave" que será descrita a seguir. No bloco iniciado na linha 3 são declaradas as variáveis do sistema onde é definido a velocidade de comunicação da porta serial na linha 4 e nas linhas subsequentes são criadas as variáveis de cada movimento do braço robótico e já definidas as posições iniciais. Na linha 12 é definida a função de leitura da posição de cada movimento e aplicadas as regras de temporização da biblioteca "suave". Na linha 20 inicia o laço que fica lendo os comandos enviados pelo servidor e baseado nos comandos recebidos executa a função. Na linha 23 temos um bloco com o comando para execução de uma atividade com a garra. Após verificar se o movimento é abrir ou fechar e validar os limites de movimento sem danificar o

equipamento o comando é executado. Por fim é finalizado o bloco e feita a leitura da nova instrução. O programa segue com blocos de instruções de cada movimento que é permitido no braço.

Além da programação principal do módulo físico, foi desenvolvido uma biblioteca em C++ que foi chamada de *suave.h*. Essa biblioteca tem por finalidade suavizar os movimentos do braço robótico evitando movimentos bruscos conforme citado na seção 4.3.2.3.

4.4.3 Módulo de Vídeo

O módulo de vídeo é composto por dois programas, o primeiro com a função da gravação do vídeo e o segundo com a função de envio do mesmo para o Amazon S3.

Na figura 27 é apresentado o código cuja finalidade é a gravação do vídeo. Na linha 3 apresenta-se a definição dos parâmetros que a ferramenta FFmpeg necessita. A função da linha 7 serve para adicionar um tempo no início e no final do vídeo. A função responsável para iniciar a gravação está na linha 11. A função da linha 15 tem a finalidade de terminar a gravação.

```

1. import {spawn} from 'child_process';
2.
3. const FFmpeg_PARAMS ≡ [
4.   '-y', '-i', '/dev/video1', '-s', '320x240', '-codec:v', 'libtheora',
5.   '-qscale:v', '7', '-codec:a', 'libvorbis', '-qscale:a', '5'];
6.
7. export async function delay(ms) {
8.   return new Promise(resolve => setTimeout(resolve, ms));
9. }
10.
11. export function iniciarGravacao(arquivo) {
12.   return spawn('ffmpeg', [...FFmpeg_PARAMS, arquivo]);
13. }
14.
15. export function finalizarGravacao(ffmpeg){
16.   ffmpeg.stdin.write('q');
17. }

```

Figura 27. Código para gravação do vídeo.

Na figura 28 é apresentada a principal função responsável pelo envio do vídeo para o Amazon S3. Essa função recebe o caminho onde o vídeo se encontra no servidor como pode se observar na linha 1 e envia o mesmo para o Amazon S3 pela instrução da linha 7. Na linha 12 é retornado a URL do vídeo que será enviada para a interface web onde o usuário poderá visualizar o mesmo.

```

1. export async function enviarVideo(path) {
2.   return new Promise<string>((resolve, reject) => {
3.     fs.readFile(path, (e1, file) => {
4.       if (e1) {
5.         reject(e1);
6.       } else {
7.         s3bucket.upload({Body: file,Key: path,ACL: 'public-read' }, (e2,data)=>
8.           if (e2) {
9.             reject(e2);
10.          } else {
11.            fs.unlink(path);
12.            resolve(data.Location);
13.          }
14.        });
15.      }
16.    });
17.  });
18. }

```

Figura 28. Código para envio do vídeo.

4.4.4 Módulo de Integração

Como descrito em seções anteriores o módulo de integração tem quatro atividades principais que são responsáveis pelo funcionamento da plataforma. A figura 29 mostra parte do código executado no módulo de integração.

```

1. async function iniciarExecutor() {
2.   while (true) {
3.     const snapshot = await firebaseService.obterProximoPrograma(Status.Validado);
4.     console.log('Iniciando: ', snapshot.key);
5.     await firebaseService.atualizarPrograma(snapshot, status: Status.Executando);
6.     let nmArquivo = `videos/${snapshot.key}.ogv`;
7.     let ffmpeg = camera.iniciarGravacao(nmArquivo);
8.     await camera.delay(1000);
9.     await braco.executarPrograma(snapshot.val().programa);
10.    console.log('finalizou execucao');
11.    await camera.delay(1000);
12.    camera.finalizarGravacao(ffmpeg);
13.    console.log('finalizou gravacao');
14.    let dsUrlS3 = await s3.enviarVideo(nmArquivo);
15.    console.log('enviou video');
16.    await firebaseService.atualizarPrograma(snapshot, {
17.      status: Status.Finalizado,
18.      video: dsUrlS3
19.    });
20.  }

```

Figura 29. Código para envio do vídeo.

A primeira funcionalidade é de monitoramento da fila de programas em execução. Na linha 2 um laço infinito que fica monitorando a chegada de novo programas na fila. Assim que chegar um novo programa pega o ID do mesmo e gera um arquivo de vídeo com esse nome conforme instrução da linha 7. Na linha 8 é iniciado o processo de gravação do vídeo. Na linha 12 tem um *delay* de 1 segundo que é adicionado no início do vídeo. A linha 10 é responsável pelo envio do programa para o braço robótico. As instruções são enviada uma por vez. Terminada a execução do programa é adicionado um delay de 1 segundo no final do vídeo e feito a finalização da gravação na linha 12. A linha 15 é responsável pelo envio do vídeo para o AMazon S3. Já na linha 17 é atualizado o status do programa e gravado a url do vídeo na base de dados da plataforma web.

4.5 RESULTADOS E DISCUSSÃO

Terminada as etapas de desenvolvimento e implementação da plataforma a mesma foi submetida a alguns testes práticos para validação do seu funcionamento. Divulgou-se o endereço da URL da plataforma em uma rede social e com alunos de turmas iniciantes do curso de redes de computadores. A participação na avaliação da plataforma se deu de forma voluntária. Os usuários tinha a sua disposição somente as instruções oferecidas na interface web da plataforma.

A plataforma apresentou-se promissora quanto à sua utilização. Os testes foram realizados em duas semanas. Sendo uma semana em novembro de 2016 e outra semana em abril de 2017. Nesse período, a plataforma para prática de robótica a distância recebeu 389 programas provenientes de 64 usuários cadastrados, totalizando em média 6,1 execuções por usuário. Desses 389 programas 28 obtiveram sucesso na execução de pelo menos uma das atividade ou ambas. Desses programas, 6 conseguiram em um único conjunto de instruções, executar as duas tarefas sugeridas. A primeira tarefa que foi colocar a bola na cesta foi executada com sucesso por 13 programas. Já a tarefa de passar a bola pelo arco foi realizada com sucesso por 9 programas. Os programas executados com sucesso foram submetidos por 8 usuários diferentes.

Afim de obter um feedback dos voluntários que se propuseram avaliar a plataforma, foi disponibilizado na interface web um questionário para que fosse relatado um parecer sobre a experiência de utilizar mesma. Das 64 usuários diferentes que submeteram programas para a execução, forma obtidas 24 respostas.

Compilando as respostas do questionário obteve-se os dados a seguir que serviram para traçar um perfil dos usuários. Esses dados servem de base para mostrar que é viável a utilização da plataforma como instrumento para disseminação do pensamento computacional. A figura 30 mostra a escolaridade dos usuários participantes que responderam o questionário. Pode-se observar que a maioria dos participantes tem ensino superior incompleto. Em segundo lugar aparecem usuários com ensino superior completo e em terceiro estudantes do ensino médio.

Também foi perguntado aos participantes se os mesmos já tiveram algum contato com programação de computadores. A figura 31 mostra que 58% dos usuários que responderam o questionário

Escolaridade (24 respostas)

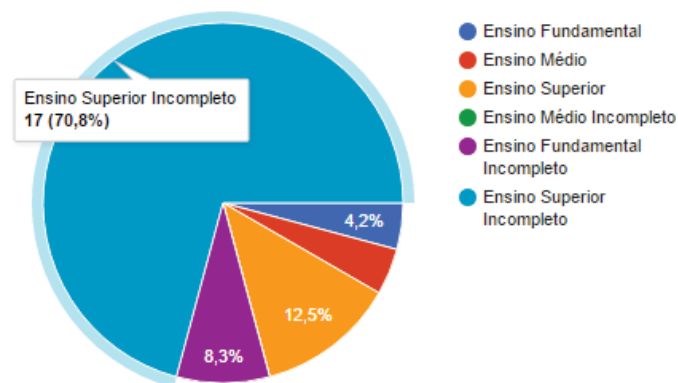


Figura 30. Escolaridade dos participantes dos testes.

havia tido algum contato com programação. Esse número nos mostra um interesse pela plataforma de usuários que nunca tiveram contato com programação.

Havia tido algum contato com programação antes de utilizar a plataforma de robótica?

(24 respostas)

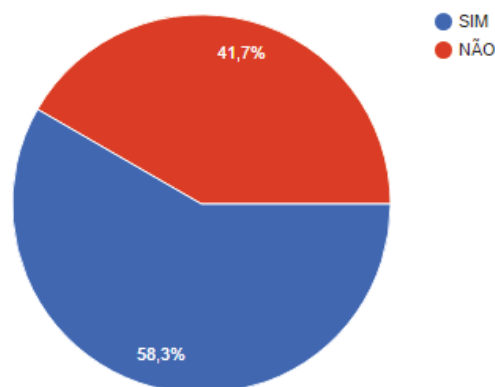


Figura 31. Conhecimento de programação dos usuários.

Outra pergunta feita no questionário foi relacionada ao grau de satisfação do usuário na utilização da plataforma. O usuário escolhia um valor entre 1 e 5 sendo 1 totalmente insatisfeito e 5 totalmente satisfeito. A figura 32 mostra os resultados onde mostra um grau de satisfação muito alto sendo que 54% se disseram totalmente satisfeitos com a plataforma utilizada. Os demais usuários avaliaram como 4 o grau de satisfação.

Também foi perguntado aos usuários se os mesmos tiveram alguma dificuldade na utilização da plataforma. O usuário encolheu entre *sim* e *não*. Nesse questão 58% dos usuários responderam *sim*. Esse resultado mostra que mesmo a plataforma sendo bem aceita tem melhorias a serem feitas. Aos usuários que responderam *sim* foi dada a opção de responder quais foram as dificuldades encon-

Grau de satisfação (de 1 a 5, sendo 1 totalmente insatisfeito e 5 totalmente satisfeito)
(24 respostas)

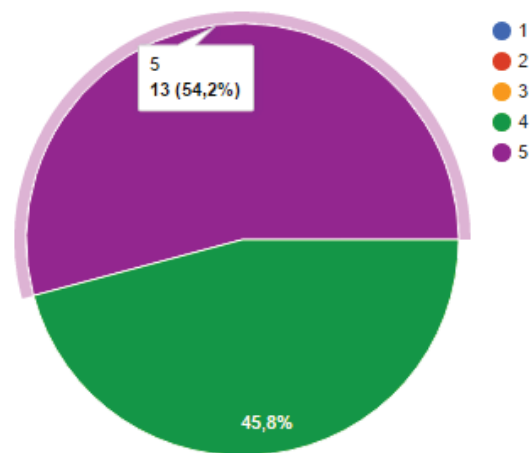


Figura 32. Satisfação do usuário com a plataforma.

tradas. Dentre as principais dificuldades encontradas destacamos as seguintes: *Dificuldade inicial de entender as direções, antes de enviar o primeiro código, se eram na visão da câmera ou levando em consideração o braço; Dificuldades em usar a plataforma pelo smartphone; Difícil percepção do quão longe a bola está do braço; Os limites dos módulos do braço não ficaram claros, foi necessário enviar um código para entendê-los.*

As figuras 33 e 34 mostram respectivamente os resultados das perguntas: *Voltaria a utilizar a plataforma se a mesma possuísse outros desafios?* e *Indicaria a plataforma a outras pessoas?*. Nessas questões é possível verificar o alto grau de aceitação da plataforma por parte dos usuários. Em ambas as questões um percentual muito alto de usuários voltariam a utilizar a plataforma e a recomendariam para outras pessoas. Esse interesse dos usuários ficou ainda mais evidente na última questão Dicas e Sugestões, onde alguns comentários classificaram a plataforma como: *Viciante; De grande entretenimento; Divertida e Desafiadora.*

Além dos comentários positivos da pergunta de *dicas e sugestões*, a última pergunta trouxe várias propostas de melhoria da plataforma. Dentre as principais sugestões destacamos algumas que serão avaliadas e muito provavelmente implementadas em futuras atualizações da plataforma: *A possibilidade dos comandos manterem-se salvos na tela principal, quando o usuário clicar no ícone histórico para visualizar a execução real do robô, não precisando assim, reescrever todos os comandos novamente; A possibilidade da inserção do último vídeo gerado pelo usuário, na tela principal; A inserção de avisos mais chamativos, quando o usuário coloca um comando que extrapola os limites; Uso de indicadores na interface web que indiquem ao usuário, se o robô está ligado ou não.*

Também foram realizados testes por professores do ensino médio de uma escola privada. Para esses professores foi feito uma breve apresentação da plataforma e sugerido para que utilizassem a mesma e fizesse uma avaliação da mesma. Participaram dos testes oito professores. Esses

Voltaria a utilizar a plataforma se a mesma possuísse outros desafios?

(24 respostas)

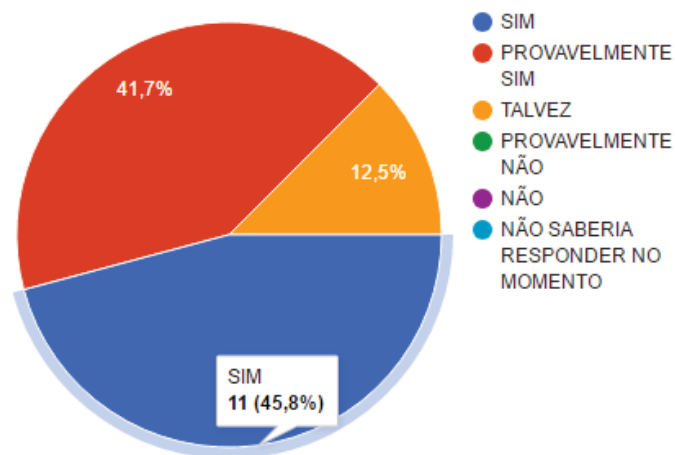


Figura 33. Usuários que voltariam a utilizar a plataforma.

Indicaria a plataforma a outras pessoas? (24 respostas)

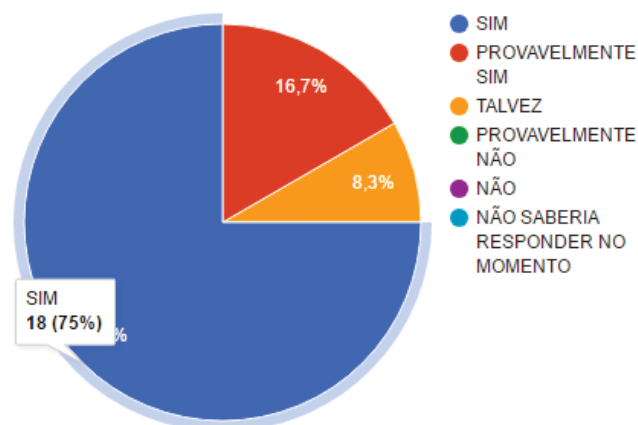


Figura 34. Usuários que indicariam a plataforma.

professores avaliaram como excelente a iniciativa de disponibilizar a plataforma para ser utilizada em atividades escolares. Todos foram unânimes em afirmar que utilizariam a mesma em suas aulas mesmo que a disciplina não tivesse muita relação com lógica, como geografia por exemplo. Nenhum professor conseguiu escrever um programa que executasse as tarefas propostas e acharam que em futuras melhorias na plataforma fosse incluído atividades mais simples onde alunos do ensino fundamental também pudessem utilizar. De modo geral todos consideraram ótima a ideia de isolar a complexidade da confecção do robôs e simplificar a linguagem de programação com a utilização da montagem de blocos.

5. CONCLUSÃO

A robótica está vivendo uma nova fase, como toda a ciência eletrônica e computacional. A cada dia surgem novos conceitos e pesquisas relacionados a robótica. Ligar robôs em todo o mundo é uma tarefa desafiadora além do escopo de qualquer projeto individual, mas já existem pesquisas relacionadas a esse assunto. Seria ousadia demais separar o conceito de robótica dos conceitos de internet das coisas (IOT), e conceitos já utilizados como computação em nuvem já estão se estendendo para a robótica em nuvem. Não podemos esquecer da inteligência artificial (AI) que permite dar autonomia aos robôs.

Com um número cada vez maior de sensores e barateamento dos mesmos tem permitido que as pesquisas em robótica tivessem acontecido com maior frequência fazendo com que novas ideias sejam colocadas em prática. Dentre as várias aplicações da robótica, observou-se um movimento voltado para a sua aplicação em ambientes educacionais. Mas ainda tem muito a ser feito em termos de estudos e pesquisas para que essas tecnologias sejam utilizadas pela comunidade escolar.

Este trabalho descreveu conceitos gerais de Pensamento Computacional, Robótica Educacional e Inteligência Artificial. Abordou também conceitos de laboratórios remotos na educação bem como apresentou trabalhos relacionados ao assunto. Identificou-se nesse sentido que diversas iniciativas e pesquisas vem sendo desenvolvidas a fim de propiciar o acesso a robótica e utilizando-se da mesma como forma de disseminação do pensamento computacional.

Como o objetivo desse trabalho foi a construção de uma plataforma interativa de aprendizagem utilizando robótica remota, realizou-se um estudo de diversas ferramentas que pudessem viabilizar o desenvolvimento desta plataforma. Constatou-se através desse estudo que é viável o desenvolvimento ao integrar os recursos das diversas ferramentas e que as mesmas são disponibilizadas gratuitamente sob o conceito de software livre. Diante disso foi possível desenvolver uma plataforma com baixo custo.

Na realização dos testes e pelo envolvimento dos voluntários, verificou-se que a plataforma se apresenta como desafiadora, provocando nos mesmos um sentimento de competição, divertimento e motivação para a realização dos desafios. Elementos que atuam como potencializadores no aprendizado já que, propiciam um interesse e envolvimento natural dos estudantes. Essa atração é provocada especialmente pelo fato do usuário vislumbrar sua programação em um robô real, o que pode ser considerado um diferencial se comparado com simuladores robóticos, que apresentam uma imagem gerada por computação gráfica, as quais por vezes não proporcionam a quem estiver programando, uma sensação de estar no controle de um mecanismo robótico, mesmo que essas repliquem exatamente o funcionamento de um robô.

Nos testes realizados com um grupo de professores, também despertou o interesse e os mesmos afirmaram que usariam a plataforma em suas aulas como forma de desenvolvimento do raciocínio lógico dos estudantes. Muitos professores se mostraram ansiosos em aplicar metodologias ativas em sala de aula, nesse sentido consideraram a plataforma um meio de diversificar suas aulas.

A tecnologia é hoje parte inerente da vida do ser. O futuro será cada vez mais tecnológico e atividades como essa poderão contribuir para que o pensamento computacional se dissemine hoje e que no futuro, os alunos poderão vir a seguir carreiras nas áreas correlatas a Internet das Coisas.

Esse ambiente permite que alunos aprendam conceitos de robótica e interajam de forma interativa com um robô, programando-o e compreendendo-o, sem ter a necessidade de possuir um dispositivo robótico. Dessa maneira, acredita-se que será possível ampliar a robótica educativa, especialmente a programação de robôs em situações onde não se teriam condições suficientes de se desenvolver tais atividades. Outro ponto a se destacar é que as dificuldades referentes a montagem do robô nessa solução podem ser abstraídas, não necessitando que o aluno domine, inicialmente, alguns conhecimentos preliminares em áreas como eletrônica e mecânica, por exemplo. Isso de certa forma é interessante, pois facilita a prática da programação de robôs, especialmente em escolas públicas e faz com que todos possam adquirir de forma indireta, porém interativa, novos conhecimentos, bem como a obtenção e melhor de benefícios cognitivos que a programação pode proporcionar.

Também através dos testes, verificou-se a necessidade de algumas modificações no projeto. Dessa forma, como trabalho futuro prevê-se a ampliação da plataforma, com a adição de novas arenas munidas de novos desafios com diferentes níveis de complexidade. Outras funcionalidades a serem implementadas é o desenvolvimento de um aplicativo para uma melhor experiência em dispositivos móveis, bem como, a inserção de sensores e atuadores que juntamente com o front-end serão responsáveis por exibir ao usuário se ele cumpriu os desafios propostos.

O projeto modular da plataforma permite agregar novas funcionalidades. Essas funcionalidades podem ser implementadas no lado do usuário ou interface web, adicionando novos desafios bem como simplificando ou aumentando a complexidade da programação. Novas funcionalidades também podem ser adicionadas no módulo físico, criando novos robôs ou mecanismos que possam ser programados remotamente.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] WING, J. M. Computational thinking. *Commun. ACM*, ACM, New York, NY, USA, v. 49, n. 3, p. 33–35, mar. 2006. ISSN 0001-0782. Disponível em: <<http://doi.acm.org/10.1145/1118178.1118215>>.
- [2] BOMBASAR, J. et al. Ferramentas para o ensino-aprendizagem do pensamento computacional: onde está alan turing? *SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO - SBIE*, 2015.
- [3] NUNES, D. J. Licenciatura em computação. *Jornal da Ciência*, may 2008.
- [4] SEEHORN, D. et al. *CSTA K–12 Computer Science Standards: Revised 2011*. New York, NY, USA, 2011. 104111.
- [5] FRANÇA, R. S. de; TEDESCO, P. C. de A. R. Um modelo colaborativo para a aprendizagem do pensamento computacional aliado à autorregulação. *SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO - SBIE*, 2014. Disponível em: <<http://www.br-ie.org/pub/index.php/sbie/article/view/3059>>.
- [6] RAMOS, F.; TEIXEIRA, L. da S. Significação da aprendizagem através do pensamento computacional no ensino médio: uma experiência com scratch. *Anais do XXI Workshop de Informática na Escola 2015*, 2015. Disponível em: <<http://www.br-ie.org/pub/index.php/wie/article/view/5024>>.
- [7] MARQUES, D. L. et al. Atraindo alunos do ensino médio para a computação: Uma experiência prática de introdução a programação utilizando jogos e python. *Anais do WIE 2011*, 2011. Disponível em: <<http://www.br-ie.org/pub/index.php/wie/article/view/1954>>.
- [8] ALMEIDA, M. E. B. de. Tecnologias na educação: dos caminhos trilhados aos atuais desafios. *Boletim de Educação Matemática*, v. 21, n. 29, p. 99–129, 2008.
- [9] FELICIA, P. *Digital games in schools : Handbook for teachers*. 2009. Complements to the study How are digital games used in schools? (European Schoolnet). Disponível em: <<https://hal.archives-ouvertes.fr/hal-00697599>>.
- [10] MEDEIROS, T. J.; SILVA, T. R. da; ARANHA, E. H. da S. Ensino de programação utilizando jogos digitais: uma revisão sistemática da literatura. *Revista Novas Tecnologias na Educação - RENOTE*, v. 11, n. 3, 2013. ISSN 1679-1916.
- [11] MEDEIROS, M. de O.; SCHIMIGUEL, J. Uma abordagem para avaliação de jogos educativos: Ênfase no ensino fundamental. *Revista Novas Tecnologias na Educação - RENOTE*, v. 10, n. 3, 2012. ISSN 1679-1916.
- [12] JUNIOR, O. de O. B.; AGUIAR, Y. P. C. Análise de abordagens objetivas para avaliação de softwares educativos. In: *Proceedings of the 13th Brazilian Symposium on Human*

- Factors in Computing Systems*. Porto Alegre, Brazil, Brazil: Sociedade Brasileira de Computação, 2014. (IHC '14), p. 353–356. ISBN 978-85-7669-291-1. Disponível em: <<http://dl.acm.org/citation.cfm?id=2738055.2738116>>.
- [13] FALCÃO, T. P.; BARBOSA, R. "aperta o play!" análise da interação exploratória em um jogo baseado em pensamento computacional. *Anais do SBIE 2015 (Proceedings of the SBIE 2015)*, 2015.
- [14] BEZERRA, R.; ALMADA, M. Construindo um robô com material reciclado: uma proposição de convergência entre robótica educacional e educação socioambiental. *XXI Workshop de Informática na Escola (WIE 2015)*, 2015.
- [15] SILVA, A. F. da. *RoboEduc: uma metodologia de aprendizado com robótica educacional*. Tese (Doutorado) — Universidade Federal do Rio Grande do Norte, Natal, RN, Brasil, 2009.
- [16] VOOGT, J. et al. Challenges to learning and schooling in the digital networked world of the 21st century. *Journal of Computer Assisted Learning*, v. 29, n. 5, p. 403–413, 2013. ISSN 1365-2729. Disponível em: <<http://dx.doi.org/10.1111/jcal.12029>>.
- [17] MELLON, C. Center for computational thinking. 2013. Disponível em: <<http://www.cs.cmu.edu/>>.
- [18] ANDRADE, D. et al. Proposta de atividades para o desenvolvimento do pensamento computacional no ensino fundamental. *Anais do WIE 2013*, 2013. Disponível em: <<http://www.br-ie.org/pub/index.php/wie/article/view/2645>>.
- [19] CODE.ORG. *Organização sem fins lucrativos que ensina programação a pessoas de todas as idades*. Disponível em: <<https://code.org/about>>. Acesso em: Mar. 2017.
- [20] WING, J. M. Computational thinking and thinking about computing. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, The Royal Society, v. 366, n. 1881, p. 3717–3725, 2008. ISSN 1364-503X. Disponível em: <<http://rsta.royalsocietypublishing.org/content/366/1881/3717>>.
- [21] BUNDY, A. Computational thinking is pervasive. *Journal of Scientific and Practical Computing*, v. 1, n. 2, p. 67–69, 2007.
- [22] BARR, V.; STEPHENSON, C. Bringing computational thinking to k-12: What is involved and what is the role of the computer science education community? *ACM Inroads*, ACM, New York, NY, USA, v. 2, n. 1, p. 48–54, fev. 2011. ISSN 2153-2184. Disponível em: <<http://doi.acm.org/10.1145/1929887.1929905>>.
- [23] GROVER, S.; PEA, R. Computational thinking in k12. *Educational Researcher*, v. 42, n. 1, p. 38–43, 2013. Disponível em: <<http://dx.doi.org/10.3102/0013189X12463051>>.
- [24] WOLZ, U. et al. Computational thinking and expository writing in the middle school. *Trans. Comput. Educ.*, ACM, New York, NY, USA, v. 11, n. 2, p. 9:1–9:22, jul. 2011. ISSN 1946-6226. Disponível em: <<http://doi.acm.org/10.1145/1993069.1993073>>.

- [25] CAURIN, G. A. P.; PEDRO, L. M. Hybrid motion planning approach for robot dexterous hands. *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, The Brazilian Society of Mechanical Sciences and Engineering, v. 31, n. 4, p. 289–296, dec 2009. ISSN 1678-5878. Disponível em: <http://www.scielo.br/scielo.php?script=sci_arttext&pid=S1678-58782009000400002&lng=en&nrm=iso&tlng=en>.
- [26] MEDEIROS, A. A. D. A survey of control architectures for autonomous mobile robots. *Journal of the Brazilian Computer Society*, Sociedade Brasileira de Computação, v. 4, n. 3, apr 1998. ISSN 0104-6500. Disponível em: <http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0104-65001998000100004&lng=en&nrm=iso&tlng=en>.
- [27] LUGER, G. F. *Inteligência Artificial - 6.ED.* Perason, 2013. ISBN 9788581435503. Disponível em: <<https://books.google.com.br/books?id=h2GJLP6jQ4kC>>.
- [28] ALVES, I. M.; MARONEZE, B. O. Projeto léxico panlatino da inteligência artificial. 2009.
- [29] HU, G.; TAY, W. P.; WEN, Y. Cloud robotics: architecture, challenges and applications. *IEEE Network*, v. 26, n. 3, p. 21–28, May 2012. ISSN 0890-8044.
- [30] MOHANARAJAH, G. et al. Rapyuta: A cloud robotics platform. *IEEE Transactions on Automation Science and Engineering*, v. 12, n. 2, p. 481–493, April 2015. ISSN 1545-5955.
- [31] GOLDBERG, K. Editorial multiplicity has more potential than singularity. *IEEE Transactions on Automation Science and Engineering*, v. 12, n. 2, p. 395–395, April 2015. ISSN 1545-5955.
- [32] DABREU, J. V. V. Disseminação de robótica em diferentes níveis de ensino. *Revista Educativa*, Revista Educativa, v. 1, p. 11–16, 2004.
- [33] CASTILHO, M. I. Robótica na educação: com que objetivos? *Monografia de Especialização em Informática na Educação*, Porto Alegre, 2002.
- [34] DABREU J. V. V.; MIRISOLA, L. G.; RAMOS, J. J. Ambiente de robótica pedagógica com br_gogo e computadores de baixo custo: Uma contribuição para o ensino médio. Anais do XXII Simpósio Brasileiro de Informática na Educação., v. 1, n. 1, p. 100–109, 2011.
- [35] SANTOS F. L.; NASCIMENTO, F. M. S.; BEZERRA, R. M. Reduc: A robótica educacional como abordagem de baixo custo para o ensino de computação em cursos técnicos e tecnológicos. 2010.
- [36] FILHO D. A. M.; GONÇALVES, P. C. Robótica educacional de baixo custo: Uma realidade para as escolas brasileiras. Anais do Workshop de Informática na Escola, v. 1, n. 1, 2008.
- [37] TRENTIN M. A.; PÉREZ, C. A. S. T. A. C. A robótica livre no auxílio da aprendizagem do movimento retilíneo. Anais do XIX Workshop de Informática na Escola., v. 1, n. 1, p. 51–59, 2013.
- [38] CHAO, K.-M. et al. Cloud e-learning for mechatronics. *Future Gener. Comput. Syst.*, Elsevier Science Publishers B. V., Amsterdam, The Netherlands, The Netherlands, v. 48, n. C, p. 46–59, jul. 2015. ISSN 0167-739X. Disponível em: <<http://dx.doi.org/10.1016/j.future.2014.10.033>>.

- [39] SOUZA, P. R. de A. *LABVAD: DESENHO E IMPLEMENTAÇÃO DO LABORATÓRIO VIRTUAL DE ATIVIDADES DIDÁTICAS COM ROBÓTICA*. Dissertação (Mestrado) — Universidade Federal do Rio de Janeiro, Rio de Janeiro, RJ, Brasil, 2015.
- [40] SILVA, J. B. da et al. Uso de dispositivos móveis para acesso a experimentos remotos na educação básica. *VAEP-RITA*, VAEP-RITA, v. 1, n. 2, p. 129–134, 2013. ISSN 2255-5706.
- [41] ERTUGRUL, N. Towards virtual laboratories: a survey of labview-based teaching/learning tools and future trends. *International Journal of Engineering Education*, v. 16, 2000. Disponível em: <<http://www.ijee.ie/articles/Vol16-3/ijee1116.pdf>>.
- [42] NEDIC, Z.; MACHOTKA, J.; NAFALSKI, A. Remote laboratories versus virtual and real laboratories. In: *33rd Annual Frontiers in Education, 2003. FIE 2003*. [S.l.: s.n.], 2003. v. 1, p. T3E–1–T3E–6 Vol.1. ISSN 0190-5848.
- [43] CATLIN, D.; WOOLLARD, J. Educational robots and computational thinking. *Proceedings of 4th International Workshop Teaching Robotics, Teaching with Robotics 5th International Conference Robotics in Education*, p. 144–151, July 2014. ISSN 978-88-95872-06-3.
- [44] ALMEIDA, C. M. dos S. *A importância da aprendizagem da robótica no desenvolvimento do pensamento computacional: um estudo com alunos do 4º ano*. Tese (Doutorado) — Universidade de Lisboa, Lisboa, Portugal, 2015. Disponível em: <<http://hdl.handle.net/10451/22412>>.
- [45] OLIVEIRA, E. J. S. de; ARAÚJO, A. L. S. O. de. Pensamento computacional e robótica: Um estudo sobre habilidades desenvolvidas em oficina de robótica educacional. *V Congresso Brasileiro de Informática na Educação - CBIE, Anais do XXVII Simpósio Brasileiro de Informática na Educação - SBIE*, 2016. Disponível em: <<http://www.br-ie.org/pub/index.php/sbie/article/view/6734/4621>>.
- [46] ZILLI, S. do R. *A Robótica Educacional no Ensino Fundamental: Perspectivas e Prática*. Dissertação (Mestrado) — Universidade Federal do Santa Catarina, Florianópolis, SC, Brasil, 2004.
- [47] ALMEIDA, T. O. *Laboratório remoto de robótica como apoio ao Ensino de Programação*. Dissertação (Mestrado) — Universidade Federal do Amazonas, Manaus, AM, Brasil, 2016. Disponível em: <<http://tede.ufam.edu.br/handle/tede/5330>>.
- [48] KULICH, M. et al. Syrotek-distance teaching of mobile robotics. *IEEE Trans. on Educ.*, IEEE Press, Piscataway, NJ, USA, v. 56, n. 1, p. 18–23, fev. 2013. ISSN 0018-9359. Disponível em: <<http://dx.doi.org/10.1109/TE.2012.2224867>>.
- [49] SYROTEK. *SyRoTek: System for robotic e-learning*. 2013. Disponível em: <<https://syrotek.felk.cvut.cz/>>. Acesso em: Dez. 2016.
- [50] VANDEVELDE, C. et al. Overview of technologies for building robots in the classroom. In: *International Conference on Robotics in Education, Proceedings*. [S.l.: s.n.], 2013. p. 122–130. ISBN 9788393755196.

- [51] HEINEN, E. et al. Raspiblocos: Ambiente de programação didático baseado em raspberry pi e blockly. *SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO - SBIE*, 2015. Disponível em: <<http://dx.doi.org/10.5753/cbie.sbie.2015.567>>.
- [52] SALEIRO, M. et al. A low-cost classroom-oriented educational robotics system. In: _____. *Social Robotics: 5th International Conference, ICSR 2013, Bristol, UK, October 27-29, 2013, Proceedings*. Cham: Springer International Publishing, 2013. p. 74–83. ISBN 978-3-319-02675-6. Disponível em: <http://dx.doi.org/10.1007/978-3-319-02675-6_8>.
- [53] ITURRATE, I. et al. A mobile robot platform for open learning based on serious games and remote laboratories. In: *2013 1st International Conference of the Portuguese Society for Engineering Education (CISPÉE)*. [S.l.: s.n.], 2013. p. 1–7.
- [54] SESHADRI, S.; GREEN, B. *Desenvolvendo com AngularJS: Aumento de Produtividade com Aplicações Web Estruturadas*. Novatec Editora, 2014. ISBN 9788575224090. Disponível em: <<https://books.google.com.br/books?id=ZEgcBQAAQBAJ>>.
- [55] WILLIAMSON, K. *Introdução ao AngularJS: Um guia para o desenvolvimento com o AngularJS*. Novatec Editora, 2015. ISBN 9788575224304. Disponível em: <<https://books.google.com.br/books?id=W4jDCAAQBAJ>>.
- [56] ROCHA, H. D. Wegoal: Sistema para gerenciamento e compartilhamento de objetivos financeiros individuais e coletivos. *Monografia (Sistemas para Internet)*., GUARAPUAVA, 2015.
- [57] INTRODUÇÃO ao Firebase. 2015. Disponível em: <<http://gdgpato-branco.org/2015/04/21/firebase-introduction/>>. Acesso em: Jul. 2016.
- [58] MAES, J. *Firestore o que é e para que serve?* Disponível em: <<http://digitalprimews.com/google-firebase/>>. Acesso em: Jul. 2016.
- [59] SEVERANCE, C. Javascript: Designing a language in 10 days. *Computer*, IEEE Computer Society, Los Alamitos, CA, USA, v. 45, p. 7–8, 2012. ISSN 0018-9162.
- [60] CHANIOTIS, I. K.; KYRIAKOU, K.-I. D.; TSELIKAS, N. D. Is node.js a viable option for building modern web applications? a performance evaluation study. *Computing*, v. 97, n. 10, p. 1023–1044, 2015. ISSN 1436-5057. Disponível em: <<http://dx.doi.org/10.1007/s00607-014-0394-9>>.
- [61] KARREI V.A.; PAVAN, W. Agronet: Uma plataforma para registro e disponibilização de dados agrícolas. *Programa de pós-Graduação em Computação Aplicada*., 2015.
- [62] BONFIM F.L.; LIANG, M. Aplicações escaláveis com mean stack. *Monografia (Ciência da Computação)*., Paraná, 2014.
- [63] SOUSA, F. P. de. Criação de framework rest/hateoas open source para desenvolvimento de apis em nodejs. *Monografia (Engenharia Informática e Computação)*., 2015. Disponível em: <<http://hdl.handle.net/10216/83505>>.

- [64] REESE, G. *Cloud Application Architectures: Building Applications and Infrastructure in the Cloud*. O'Reilly Media, 2009. (Theory in Practice (O'Reilly)). ISBN 9780596555481. Disponível em: <<https://books.google.co.uk/books?id=j8YO7gVqMqAC>>.
- [65] AWS. *About AWS*. Disponível em: <<https://aws.amazon.com/about-aws/?nc1=hjs>>. Acesso em: Dez. 2016.
- [66] VERAS, M. *Arquitetura de Nuvem - Amazon Web Services*. BRASPORT. ISBN 9788574525686. Disponível em: <<https://books.google.com.br/books?id=RY7kp2dT0jwC>>.
- [67] LIMA, D. J. Proposta de arquiteturas para desenvolvimento de aplicações corporativas voltadas ao ambiente de nuvem. *Monografia (Engenharia de Projetos de Software)*., 2013. Disponível em: <<http://hdl.handle.net/10216/83505>>.
- [68] AWS. *Amazon S3 Product Details*. Disponível em: <<https://aws.amazon.com/s3/details/?nc1=hjs>>. Acesso em: Dez. 2016.
- [69] FFMPEG. *About Ffmpeg*. 2017. Disponível em: <<http://www.ffmpeg.org/about.html>>. Acesso em: Fev. 2017.
- [70] ARDUINO. *Arduino*. 2016. Disponível em: <<https://www.arduino.cc/>>. Acesso em: Dez. 2016.
- [71] MELLIS, D. *O hardware em código aberto*. 2009. Disponível em: <<http://raphaelalves.blogspot.com.br/2009/03/o-hardware-em-codigo-aberto.htmlaxzz4b7fPqaQf>>. Acesso em: Dez. 2016.
- [72] PINTO, M. D. C. *APLICAÇÃO DE ARQUITETURA PEDAGÓGICA EM CURSO DE ROBÓTICA EDUCACIONAL COM HARDWARE LIVRE*. Dissertação (Mestrado) — Universidade Federal do Rio de Janeiro, Rio de Janeiro, RJ, Brasil, 2011. Disponível em: <http://www.nce.ufrj.br/ginape/publicacoes/dissertacoes/d2011/d2011_marcos_de_castro.pdf>.
- [73] CRAIG, J. *Introduction to Robotics: Mechanics and Control*. Pearson/Prentice Hall, 2005. (Addison-Wesley series in electrical and computer engineering: control engineering). ISBN 9780201543612. Disponível em: <<https://books.google.com.br/books?id=MqMeAQAAIAAJ>>.