UNIVERSIDADE DO ESTADO DO RIO GRANDE DO NORTE – UERN

FACULDADE DE CIÊNCIAS EXATAS E NATURAIS – FANAT

DEPARTAMENTO DE INFORMÁTICA – DI

Thalia Katiane Sampaio Gurgel

RepositORE: Um Repositório de Objetos de Aprendizagem para Robótica Educacional

MOSSORÓ - RN

2019

Thalia Katiane Sampaio Gurgel

RepositORE: Um Repositório de Objetos de Aprendizagem para Robótica Educacional

Monografia apresentada à Universidade do Estado do Rio Grande do Norte como um dos pré-requisitos para obtenção do grau de bacharel em Ciência da Computação, sob orientação do Prof. Dr. Sebastião Emídio Alves Filho.

MOSSORÓ – RN

2019

Catalogação da Publicação na Fonte.
Universidade do Estado do Rio Grande do Norte.

G979r Gurgel, Thalia Katiane Sampaio
RepositORE: Um Repositório de Objetos de Aprendizagem para Robótica Educacional. / Thalia Katiane Sampaio Gurgel. - UERN, 2019.
113p.

Orientador(a): Prof. Dr. Sebastião Emidio Alves Filho.
Monografia (Graduação em Ciência de Computação). Universidade do Estado do Rio Grande do Norte.

1. Objetos de Aprendizagem. 2. Robótica Educacional. 3. Metadados. 4. Dublin Core. 5. Sistemas Colaborativos. I. Filho, Sebastião Emidio Alves. II. Universidade do Estado do Rio Grande do Norte. III. Título.

O serviço de Geração Automática de Ficha Catalográfica para Trabalhos de Conclusão de Curso (TCC´s) foi desenvolvido pela Diretoria de Informatização (DINF), sob orientação dos bibliotecários do SIB-UERN, para ser adaptado às necessidades da comunidade acadêmica UERN.

Thalia Katiane Sampaio Gurgel

RepositORE: Um Repositório de Objetos de Aprendizagem para Robótica Educacional

Monografia apresentada como pré-requisito para obtenção do título de Bacharel em Ciência da Computação da Universidade do Estado do Rio Grande do Norte – UERN, submetida à aprovação da banca examinadora composta pelos seguintes membros:

Aprovada em: 06/02/2019

Banca Examinadora

Prof Dr. Sebastião Emídio Alves Filho
Universidade do Estado do Rio Grande do Norte - UERN

Prof. Alysson Mendes de Oliveira
Universidade do Estado do Rio Grande do Norte - UERN

Prof Dr. Rommel Wladimir de Lima
Universidade do Estado do Rio Grande do Norte - UERN

À aqueles de maior importância em minha vida, Deus, meus pais: Agábio e Júlia, e meu irmão Agabio Filho..

AGRADECIMENTOS

Tenho a agradecer primeiramente a Deus, por iluminar e abençoar minha trajetória, mesmo diante de muitas dificuldades consegui chegar até aqui e concretizar mais um objetivo. Levo comigo sempre a fé das palavras de Deus, em especial a palavra do Salmos 119:105 "*Tua palavra é lâmpada para os pés e luz para o meu caminho*".

Tenho muito a agradecer a todos que passaram em minha vida durante essa trajetória, pois não é fácil morar longe do conforto de casa, porém as dificuldades se tornam insignificantes diante dos objetivos que desejamos alcançar. Obrigado a Rejane Gurgel e toda sua família por ter me dado o suporte inicial desde quando cheguei em Mossoró, agradeço infinitamente aos meus pais por sempre terem me ensinado o caminho certo a seguir, me repassando valores que eu levarei para o resto da vida, e que tudo na vida vem através de muito esforço e dedicação.

Agradeço a todos os amigos que adquiri durante a graduação, sempre vou lembrar de todos os momentos de dificuldades e perrengues que passamos, tenho muito a agradecer a DEAD- Diretoria de Educação a Distância da Uern por ter me aberto as portas para estágio na instituição, o tempo decorrente nesse ambiente foi de muito aprendizado e conhecimento, muito obrigado a todos os meus amigos que adquiri nesse local, em especial Thiago Alefy e Ed Carlos que foram meus companheiros de estágio na área de TI, não podendo esquecer também dos outros que lá estiveram sempre dividindo o ambiente e os conhecimentos, Zaira, Arlene, Pedro, Luiza, Tasso e Edymara, obrigada a todos os coordenadores de TI que passaram por essa diretoria: Heitor Liberalino, Dênis, Ceres e Alysson.

Não posso esquecer das minhas fiéis amigas de graduação, Lígia Maria e Elisa Andrade, obrigada por todos os momentos divertidos que passamos, e por me abrigar na casa de vocês quando precisei, risos. Elisa sempre muito agoniada e falante, está sempre por perto para resolver os problemas, Lígia a mulher com visão empreendedora, desejo muito sucesso na vida para vocês. Também agradeço a

Thiago Oliveira e Leonardo Bandeira, por toda ajuda que me foi dada durante o período da graduação. Agradeço ao meu amigo e conselheiro que sempre esteve ao meu lado me apoiando em todos os momentos, Pe. Chagas Costa, suas palavras sábias sempre foram de muita relevância para o meu crescimento, agradeço de coração. A toda equipe do LORDI em especial a pessoa de Mizael, obrigada por toda ajuda nos conhecimentos de eletrônica, e por todos os conselhos. A todos meus sinceros agradecimentos.

Dedico especial agradecimento à meu orientador Sebastião Alves, uma pessoa dedicada no que faz que consegue repassar os seus conhecimentos com muita sabedoria, obrigado por ter me ajudado no direcionamento das ideias para a concretização deste trabalho. Muito obrigada pela oportunidade que me foi de dada de participar dos seus projetos de pesquisa, e por ter me apresentado uma área tão gratificante de ser abordada. Sua força, determinação e esforço são inspiradores para qualquer pessoa que queira se destacar profissionalmente. Agradeço também seu auxílio, disponibilidade. Obrigada por cada incentivo e orientação.

Gostaria de agradecer também a todos os professores do Departamento de Informática da UERN, muito obrigada por todos os conhecimentos que foram repassados durante esse período de graduação, todo o aprendizado me fez crescer tanto pessoalmente como profissionalmente. Todos foram de muita contribuição para minha vida acadêmica.

Finalizo os agradecimentos destacando que é difícil agradecer a todos que fizeram parte dessa trajetória, então obrigado a todos que contribuíram para a conclusão desta etapa em minha vida.

“Julgue seu sucesso pelas coisas que você tem que renunciar para conseguir.”

Dalai Lama

RESUMO

A educação é uma área viável e motivadora para o uso de tecnologias pois o processo de aprendizagem pode-se tornar mais dinâmico e interessante. Dentre os recursos tecnológicos utilizados na educação destaca-se a Robótica Educacional como um meio de possibilitar o desenvolvimento de projetos tecnológicos dentro e fora do ambiente de ensino superior, onde pode envolver técnicas de construção e manipulação de robôs e viabilizar o desenvolvimento do processo criativo, raciocínio lógico e a interdisciplinaridade. Entretanto adquirir conhecimentos nesta área pode ser uma tarefa árdua uma vez que os objetos de aprendizagem estão dispersos devido ao aumento da produção de conteúdo advindos com a massificação do uso da internet. Assim, o presente trabalho tem como objetivo principal apresentar o RepositORE, um repositório onde objetos de Robótica Educacional possam ser armazenados e buscados por usuários que necessitam adquirir determinadas habilidades. Para que os objetos de Robótica Educacional sejam encontrados mais rapidamente e possa atingir os seus objetivos técnicos e pedagógicos o sistema usa metadados para a correta descrição dos objetos armazenados baseados no padrão Dublin Core Metadata. Visando melhorar e tornar simplificada a busca pelos objetos foi feita uma adaptação e extensão do padrão Dublin Core para representar as informações específicas para Robótica Educacional. O RepositORE permite aos usuários a inserção de novos conteúdos e a colaboração através da complementação de informações de objetos cadastradas por outros usuários.

Palavras-chave: Objetos de Aprendizagem, Robótica Educacional, Metadados, Dublin Core, Sistemas colaborativos.

ABSTRACT

Education is a viable and motivating area for the use of technologies because the learning process can become more dynamic and interesting. Among the technological resources used in education, Educational Robotics stands out by enabling the development of technological projects inside and outside the higher education environment, where it can involve techniques of construction and manipulation of robots and enable the development of the creative process, logical reasoning and interdisciplinarity. However, acquiring knowledge in this area can be a hard task since the learning objects are spread due to the increase of content production brought from the massive use of the Internet. Thus, the present work has as general objective present RepositORE, a repository where these objects of Educational Robotics can be stored and searched by users who need to acquire certain abilities. In order for Educational Robotics objects to be found faster and achieve their technical and pedagogical goals, the system uses metadata for the correct description of stored objects based on the Dublin Core Metadata standard. In order to improve and simplify the search for objects, an adaptation and extension of the Dublin Core standard was made to represent the specific information for Educational Robotics. RepositORE allows users to insert new content and collaborate through complementing information from objects registered by other users.

Keywords: Learning Objects, Educational Robotics, Metadata, Dublin Core, Collaborative Systems.

.

LISTA DE SIGLAS

ADL	*Advanced Distributed Learning Initiative*
API	*Application Programming Interface*
AVAS	Ambientes Virtuais de Aprendizagem
CRUD	*Create, Read, Update, Delete*
CSS	*Cascading Style Sheets*
DC	*Dublin Core*
DCMI	*Dublin Core Metadata Initiative*
DCMES	*Dublin Core Metadata Element Set*
EAD	Educação a Distância
EML	*Educational Modeling Language*
ESR	*Firefox Extended Support Release*
HTML	*HyperText Markup Language*
IDE	*Integrated Development Environment*
IMS	*Global Learning Consortium*
LOM	*Learning Object Metadata Working Group*
MIT	*Massachusetts Institute of Technology*
MVC	*Model View Controller*
NOSQL	*Not Only SQL*
OA	Objeto de Aprendizagem
OAS	Objetos de Aprendizagem

RCX	*Robotic Command Explore*
RDF	*Resource Description Framework*
RE	Robótica Educacional
ROAS	Repositórios de Objetos de Aprendizagem
SAAS	Software como um serviço
SCORM	*Sharable Content Object Reference Model*
SGML	*Standard Generalized Markup Language*
UERN	Universidade do Estado do Rio Grande do Norte
VSCODE	*Visual Studio Code*
XML	*Extensible Markup Language*

LISTA DE FIGURAS

Figura 1: Elementos de metadados do padrão DC Simples.....34
Figura 2: Evolução dos kits de robótica educacional LEGO 46
Figura 3: Exemplo de programação no software NXT.....48
Figura 4: Interface de programação Lego EV3.....49
Figura 5: Programa seguidor de linha NXT e EV3.....49
Figura 6: Aplicativo Free Lego Boost Creative Toolbox.....50
Figura 7: Modelo Vernie The Robot LEGO BOOST.....50
Figura 8: Kit de Robótica Modelix Ensino Médio e Superior.....52
Figura 9: Modelix com Microcontrolador.....52
Figura 10: Exemplo cenário interativo Modelix.....53
Figura 11: Tabela comparativa dos modelos de placas Arduino.....56
Figura 12: Ambiente de programação do Arduino.....57
Figura 13: Modelo HEXBUG Vex IQ Robotics.....58
Figura 14: IDE EasyC Vex Robotics.....58
Figura 15: Cyberbox.....59
Figura 16: GoGo Widget.....60
Figura 17: Tinker, ambiente de programação GoGo Board.....61
Figura 18: Placa Go-Go Board.....61
Figura 19: Super Robby.....62
Figura 20: Tela RoboEduc 1.0.....63
Figura 21: Telas RoboEduc 2.0.....63
Figura 22: Telas RoboEduc 2.1.....64
Figura 23: Telas RoboEduc 3.0.....64
Figura 24: RoboEduc Inbox.....66
Figura 25: Robokit.....66
Figura 26: Tela de Início Robolab.....68
Figura 27: Linguagens cadastradas no ambiente Weduc.....69
Figura 28: Ambiente de desenvolvimento weduc.....70
Figura 29: Ambiente de desenvolvimento Megalogo.....71

Figura 30: Mapeamento de Elementos Metadados Dublin Core para Descrição e Gerenciamento da Biblioteca Digital de Teses da USP.....73
Figura 31: Adaptação do AACR2 ao DCMI.....74
Figura 32: Metadados utilizados pela Brasiliana Digital na Plataforma Corisco.....75
Figura 33: Diagrama de Caso de Uso do sistema.....83
Figura 34: Arquitetura da aplicação.....84
Figura 35: Desempenho do angular relacionado aos seus concorrentes.....87
Figura 36: Arquitetura do Angular.....88
Figura 37: Controle de versionamento da aplicação utilizando Github.....91
Figura 38: Ambiente de desenvolvimento Visual Studio Code.....92
Figura 39: Tela de Início.....95
Figura 40: Tela Pesquisar Objetos.....95
Figura 41: Visualização de Descrição do Objeto.....96
Figura 42: Funcionalidade Ver metadados.....97
Figura 43: Tela de Login.....97
Figura 44: Tela de Colaboração/Cadastro.....98
Figura 45: Interface do Banco de Dados com objetos cadastrados.....99
Figura 46: Objetos descritos no Banco de Dados em formato Json.....100

LISTA DE TABELAS

Tabela 1: Descrição de um objeto de robótica disponível no site nxtprograms.....35

SUMÁRIO

LISTA DE SIGLAS 11

LISTA DE FIGURAS 13

SUMÁRIO 16

1 INTRODUÇÃO 18

1.1 Contextualização 19

1.2 Objetivos 21

1.3 Justificativa 21

1.4 Metodologia 22

1.5 Estrutura do Documento 22

2 REFERENCIAL TEÓRICO 24

2.1Objetos de Aprendizagem 24

2.2 Padrões de Identificação de metadados 29

2.3 Padrão de Metadados Dublin Core 32

2.4 Bases de objetos educacionais 37

3 ROBÓTICA EDUCACIONAL 40

3.1 Conceito de Robótica Educacional 40

3.2 Objetivos da Robótica Educacional 42

3.3 Kits de Robótica Utilizados na Educação 44

3.3.1 LEGO MINDSTORMS 45

3.3.2 Modelix Robotics 50

3.3.3 Arduino 53

3.3.4 VEX 57

3.3.5 Cyberbox 58

3.3.6 GoGoBoard 59

3.3.7 Super Robby 61

3.3.8 RoboEduc 62

3.3.9 Robokit 66

3.4 Ambientes de programação utilizados na Robótica Educacional 66

4 USO DO PADRÃO DUBLIN CORE PARA OBJETOS DE ROBÓTICA EDUCACIONAL 72

4.1 Trabalhos Relacionados 72

4.2 Adaptação da proposta ao padrão Dublin Core para Robótica Educacional 76

5 RepositORE 81

5.1 Modelagem do Sistema 81

5.2 Tecnologias utilizadas 84

5.3 Resultados 94

6 CONCLUSÃO 101

6.1 Trabalhos Futuros 101

REFERÊNCIAS 103

APÊNDICE 111

1 INTRODUÇÃO

Com o desenvolvimento de novas Tecnologias de Informação e Comunicação (TIC's), elas têm passado por intensas transformações, aumentando sua aplicação na educação.

> A tecnologia disponível aos estudantes tem por objetivo principal desenvolver as possibilidades individuais, tanto cognitivas como estéticas, através de sua empregabilidade variada, que o docente pode realizar para interagir com o grupo. Ignorar o fato de que a tecnologia, o saber tecnológico e as produções tecnológicas fizeram e ainda podem fazer grandes mudanças na vida cotidiana dos estudantes, seria um retrocesso a um ensino que, contraditoriamente, não poderia ser mais considerado tradicional, e sim, ficcional (LITWIN, 1997).

De acordo com Papert (PAPERT 1994), "o uso da tecnologia na educação, é caracterizada por um paradoxo existente, pois, através da utilização de métodos tecnológicos ocorrerá uma mudança que virá para eliminar a natureza técnica da aprendizagem tradicional na Escola".

> Logo, através do uso das tecnologias, torna-se possível a inovação de métodos e de técnicas do professor, ampliando as possibilidades de aprendizagem. Essa ampliação das possibilidades de ensino configura uma grande mudança de paradigma, já que o modelo de escola existente, nos moldes como conhecemos, surgiu na Antiguidade (CASTRO, 2008).

A partir dessa inclusão de tecnologias agregadas a educação, a robótica educacional surgiu como um meio de possibilitar o desenvolvimento de projetos tecnológicos dentro e fora do ambiente de ensino superior, onde pode envolver técnicas de construção e manipulação de robôs e viabilizar o desenvolvimento do processo criativo, raciocínio lógico, a interdisciplinaridade em diferentes áreas.

A robótica é uma área multidisciplinar, onde pode envolver diversas áreas, como a engenharia elétrica, engenharia mecânica e inteligência artificial. A robótica também agrega um papel muito importante na indústria de automação, pois essa parte da indústria é responsável pela fabricação de robôs capazes de substituir o ser humano em diversas atividades, ou até mesmo tarefas que o ser humano não consegue realizar.

E de acordo com Castro (2008) além de ser uma simples ferramenta, a robótica educacional tornou-se uma metodologia que veio possibilitar o avanço de tecnologias de maneira efetiva, fazendo com que as pessoas possam obter uma melhoria no

desenvolvimento de suas habilidades e competências. Ao ser inserida no meio escolar, esse avanço vem com o objetivo de intensificar a capacidade crítica, o raciocínio lógico e uma melhor percepção para resolução de problemas.

A autora afirma que por suas formas e pela possibilidade de executar tarefas com autonomia, os robôs fascinam tanto os adultos quanto as crianças. Observando esse quadro, a robótica na educação propicia um estímulo a mais entre os estudantes, permitindo que eles busquem os conhecimentos necessários para realizarem determinada atividade.

1.1 Contextualização

“A Internet é atualmente a expressão maior da utilização dos computadores e dos meios eletrônicos para o armazenamento, a busca e a recuperação de informações armazenadas em meio eletrônico” (GRACIO, 2002). Nela podemos ter acesso a diversos tipos de recursos. O acesso à informação nos dá uma visão abrangente diante dos diversos conteúdos que são apresentados diante de pesquisas feitas na *web.* Esses recursos informacionais podem estar caracterizados em e-mails, listas de discussão, *websites*, artigos em revistas eletrônicas, informações comerciais, culturais, artísticas,e bibliotecas virtuais e digitais. Dessa forma os conteúdos digitais passaram a ser ferramentas imprescindíveis na conduta de processos educacionais, fazendo com que objetos de aprendizagem possam oferecer um conteúdo digital que auxilie no processo educacional.

Os objetos de aprendizagem(OA) são como ferramentas estratégicas para o processo de ensino e aprendizagem. Eles podem ser utilizados para que determinadas questões no âmbito do aprendizado e da perspectiva científica possam ser incrementadas e conduzidas a partir de outras alternativas além do livro didático e estratégias tradicionais em sala de aula. De acordo com o IEEE (2002) um objeto de aprendizagem é definido como qualquer entidade digital ou não digital que pode ser usada para aprendizado, educação ou treinamento.

Os objetos de aprendizagem transmitem informações que são encarregadas de produzir um determinado conhecimento. Porém para que eles possam atingir seu objetivo, é necessário que sejam compartilhados, registrados e documentados, e o usuário possa ter acesso a essa informação de forma simplificada. Para que o usuário

possa ter acesso aos objetos, e eles atendam a sua necessidade, é preciso que a informação esteja bem descrita e bem representada para que facilite a sua busca. Esse objetos geralmente são armazenados em Repositórios, que tem como principal objetivo facilitar a localização e o reuso de OAs.

Os repositórios não armazenam apenas os objetos educacionais, mas também os descritores estruturais e semânticos para esses objetos. É recomendado que seja adotado um padrão de descrição dos seus dados, que são chamados de Metadados. Metadado é frequentemente identificado como dado acerca de dado, ou informação sobre informação que está no espaço digital e virtual. É um sumário de informações sobre a forma e o conteúdo de um recurso eletrônico, ou não.

Porém a quantidade de dados que são gerados diariamente na internet tem evoluído de forma gradativa e significativa, e a partir desse crescimento pode-se perceber uma amplificação exponencial na quantidade dos dados e informações. Esse aumento de dados pode dificultar diretamente na obtenção de resultados em pesquisas específicas, gerando um grande número de informações, fazendo com que eventualmente não obtenha-se resultados esperados nas filtragens de conteúdo.

> Para recuperar a informação armazenada na Internet e transformá-la em conhecimento, são utilizadas atualmente ferramentas de busca, que consistem em programas de computadores com bancos de dados que armazenam descritores de recursos disponíveis na Internet, como Yahoo, Google entre outros. Tais ferramentas não possuem mecanismos de busca iguais e, de acordo com a característica de cada uma, o número e a qualidade das informações recuperadas podem variar enormemente (GRÁCIO, 2002).

As ferramentas de busca do tipo diretório, que organizam as informações em categorias, realizam a indexação de documentos utilizando especialistas, gerando informações com mais conteúdo, mas tornam esse trabalho muito demorado.

> As ferramentas do tipo motores de busca, ao contrário, utilizam software (chamado robô de busca) para buscar automaticamente as informações na Web, tornando a indexação mais ágil, mas acabam gerando um número muito grande de informações, deixando algumas páginas da Internet fora do catálogo (CENDÓN, 2001).

Esse fenômeno também pode ser visto com relação ao conteúdo de Robótica Educacional. Dessa forma, visando a necessidade de que que os conhecimentos sobre Robótica Educacional estejam organizados e indexados como objetos de aprendizagem com metadados descritivos para facilitar a sua busca, este trabalho foi feito para que os objetos de aprendizagem de Robótica Educacional sejam melhor

descritos e disponibilizados em um repositório específico destinado apenas para objetos relacionados a esta temática.

1.2 Objetivos

O principal objetivo deste trabalho é facilitar a localização e o reuso de OAs com conteúdo destinado à Robótica Educacional, através da criação de um Repositório de Objetos de Aprendizagem (ROA) que armazena não só os objetos educacionais, mas também metadados.

Esta monografia visa a construção de um repositório de objetos de aprendizagem com descrição de seus dados baseados no padrão Dublin Core Metadata, e de uso exclusivo para o conteúdo destinado a Robótica, como por exemplo montagem de objetos de acordo com modelos de kits específicos, programação de movimentos, novos ambientes de programação destinados a esse conteúdo, entre outros.

O trabalho também apresenta uma proposta de adaptação do padrão Dublin Core para a descrição dos Objetos de Robótica Educacional, fazendo um mapeamento de todos os campos descritivos que são necessários para que o objeto esteja bem representado no repositório.

1.3 Justificativa

Foi feita uma pesquisa bibliográfica acerca dos repositórios de objetos de aprendizagem existentes, tanto nacionais quanto internacionais, onde foi visto que esses repositórios não possuem um conteúdo voltado apenas para o conteúdo de Robótica. Eles disponibilizam diversos tipos de conteúdos destinado às mais diversas áreas de conhecimento, mas os OAs voltados para o conhecimento de Robótica são encontrados em sua maior parte em páginas na internet, blogs ou websites que não possuem descrição de metadados.

A ideia deste trabalho justifica-se pela necessidade de fácil obtenção de Objetos de Aprendizagem voltados para Robótica Educacional, pois de acordo com os dados descritos anteriormente, a dificuldade em encontrar esses conteúdos na *web* se torna cada mais difícil, pois a quantidade de páginas que fornecem essas

informações não utilizam padrões descritivos para que o objeto seja melhor encontrado de acordo com a necessidade específica do usuário.

1.4 Metodologia

Foi visto que em repositórios de OAs, em sua maioria, utilizam padrões de metadados descritivos para inserção dos OAs nesses repositórios. Os dados descritivos de objetos de aprendizagem precisam ser descritos de acordo com um padrão de metadados. Então foi feita uma busca a respeito dos padrões de metadados existente e a partir desse estudo foi a feita a escolha pelo padrão Dublin Core, devido sua simplicidade na utilização dos recursos e também o fato de sua extensibilidade ser bastante acessível e descomplicada, ou seja ele oferece uma melhor adaptação às necessidades do usuário.

Foi realizado um estudo mais aprofundado sobre Robótica Educacional e suas principais características. A partir disso, foi feita uma adaptação do padrão Dublin Core para representar as informações de OA's de Robótica Educacional visando a construção de um repositório. Também foi realizada uma pesquisa a respeito das tecnologias atuais que estão sendo mais utilizadas para a construção de páginas web visando a implementação do repositório. Para a construção do protótipo dessa aplicação foram utilizados *frameworks*, banco de dados em tempo real dentre outras ferramentas que auxiliaram no desenvolvimento, todas as ferramentas utilizadas foram citadas neste trabalho.

1.5 Estrutura do Documento

O restante desta monografia encontra-se estruturada da seguinte forma. O capítulo 2 apresenta os principais conceitos de objetos de aprendizagem, os principais padrões de identificação de metadados. É feito um aprofundamento acerca do padrão utilizado, o Dublin Core, logo após são descritos os principais objetivos das bases de objetos educacionais, os repositórios.

O capítulo 3 apresenta as principais definições sobre robótica educacional e seus objetivos, bem como também descreve os kits de robótica educacional que são utilizados na educação e seus ambientes de programação.

O capítulo 4 relata os trabalhos relacionados que serviram como base para construção deste, e mostra o mapeamento que foi feito para adaptação do repositório de Robótica Educacional ao Padrão Dublin Core. O capítulo 5 é caracterizado pela modelagem da aplicação, seus principais diagramas, as tecnologias que foram utilizadas para o desenvolvimento e os resultados obtidos.

Por fim no capítulo 6 temos as considerações finais sobre o desenvolvimento deste trabalho, apresentando os principais benefícios para a construção deste repositório bem como as propostas para trabalhos futuros.

2 REFERENCIAL TEÓRICO

No âmbito educacional de aprendizagem e conhecimento, pode-se perceber a busca de novas tecnologias para obter conteúdos de forma simples e didática. A partir dessa demanda, foi observado um aumento na quantidade de materiais dos mais diversos tipos e temas que são disponibilizados na internet. Nesse contexto temos uma variedade de materiais, como software, imagens, vídeos, cursos e jogos, cabe à pessoa que busca por esse conhecimento escolher a melhor ferramenta que irá auxiliar no seu processo de aprendizado.

2.1Objetos de Aprendizagem

Com base nessas necessidades e com o que já era utilizado, surgiu uma formulação denominada de Objeto de Aprendizagem (OA). Segundo Wiley (2002) “o OA é qualquer recurso digital que possa ser reutilizado em diferentes contextos de aprendizagem”.

> Um objeto de aprendizagem é um arquivo digital interativo produzido a partir de softwares específicos, os quais podem ser disponibilizados na internet com a finalidade pedagógica de facilitar o processo cognitivo e com possibilidade de ser reaproveitado e adaptado de acordo com a necessidade dos aprendizes (BEHAR; TORREZAN, P. 33-65 2009).

Segundo o IEEE (2002) “um objeto de aprendizagem é definido como qualquer entidade digital ou não digital que pode ser usada para aprendizado, educação ou treinamento”.

Saraiva e Netto (2010) citam que “com o objetivo de facilitar estes entendimentos abstratos, acredita-se que o uso de objetos de aprendizagem que permitam a exploração dos conteúdos de forma digital e com recursos midiáticos como simulações, gráficos, desenhos, dentre outros, possa fazer com que os alunos tenham uma melhor compreensão dos conteúdos e ainda se sintam mais motivados ao estudo com a utilização destes recursos”.

Braga (2014, P. 29) fala que “um OA é tanto mais interativo quanto maior a capacidade de intervenção do aluno no conteúdo ensinado por esse OA. Um OA com alta interatividade possibilita a ação do aluno e o estabelecimento de uma relação de reciprocidade”. Ou seja, quanto mais o OA permite que o aluno se aproprie de

informações, reflita e seja ativo em seu processo de aprendizagem, mais interativo ele é.

Os objetos de aprendizagem são como ferramentas estratégicas para o processo de ensino e aprendizagem. Eles podem ser utilizados para que determinadas questões no âmbito do aprendizado e da perspectiva científica possam ser incrementadas e conduzidas a partir de outras alternativas além do livro didático e estratégias tradicionais em sala de aula. Isto traz um processo mais autônomo para o aluno, que pode estar incluído nos mais diversos tipos de recursos digitais especificados a seguir:

- **Vídeoaula:** A videoaula é uma ferramenta pedagógica importante, pois nela o participante tem a possibilidade de visualizar o conteúdo em audiovisual, seja por uma aula de um professor, depoimento de um profissional da área ou ainda uma demonstração de técnica. (Portal Educação , 2013).

- **Animação:** Técnica cinematográfica que se baseia em fotografar uma sequência de imagens fixas de desenhos ou bonecos, visando a sugestão de movimentos (Infopedia, 2003).

- **Simulação:** As simulações são animações que representam um modelo da natureza e, devido a isso, podem ser muito utilizadas como objetos de aprendizagem. Sistemas computacionais para simulação auxiliam os desenvolvedores e pesquisadores, na medida em que permitem estudar o modelo em ambientes controlados, possibilitando a análise de itens tais como: a dinâmica do modelo, detalhes de sua estrutura, execução variada da simulação, alterando parâmetros de entrada para verificar os resultados obtidos etc. (NASCIMENTO et al., 2013).

- **Software:** São programas de computadores que permitem executar determinadas tarefas e resolver problemas de forma automática (PIMENTEL; BRAGA, 2013).

- **Hipertexto:** Corpus textual em suporte eletrónico que, por meio de hiperligações, contém remissões para outros blocos textuais disponíveis em

rede, de tal modo que as possibilidades de leituras interativas e não sequenciais se tornam múltiplas (Infopedia, 2013).

Destacando que OAs não se resumem apenas a esses recursos que foram citados. Eles são utilizados como instrumentos para o processo de construção de objetos de aprendizagem, além disso podemos citar alguns aspectos pedagógicos que são considerados importantes para a construção de um OA segundo (GALAFASSI et al.,2013.):

- **Interatividade**: indica se há suporte às consolidações e ações mentais, requerendo que o aluno interaja com o conteúdo do OA de alguma forma, podendo ver, escutar ou responder algo.
- **Autonomia**: indica se os objetos de aprendizagem apoiam a iniciativa e tomada de decisão
- **Cooperação**: indica se há suporte para os alunos trocarem opiniões e trabalhar coletivamente sobre o conceito apresentado.
- **Cognição**: refere-se às sobrecargas cognitivas alocadas na memória do aluno durante o processo de ensino-aprendizagem.
- **Afetividade**: refere-se aos sentimentos e motivações do aluno com sua aprendizagem e durante a interação com o OA.

Com o crescimento gradativo da utilização da tecnologia de objetos de aprendizagem, surgiram diversas especificações para regulamentar o desenvolvimento destes conteúdos digitais. A utilização de um padrão para o desenvolvimento e compartilhamento de um objeto é de suma importância e é imprescindível que um OA tenha que atender os requisitos necessários.

Dentre esses requisitos necessários podemos especificar algumas características técnicas dos OAs citadas por (BRAGA et al., 2012):

- **Disponibilidade:** indica se o objeto está disponível para ser utilizado.
- **Acessibilidade:** indica se o objeto pode ser acessado por diferentes tipos de usuários (ex: idosos, deficientes visuais etc), em diferentes lugares (ex:.

lugares com acesso a Internet, lugares sem acesso a Internet etc.) e por diferentes tipos de dispositivos (ex.: computadores, celulares, tablets etc.).

- **Confiabilidade:** indica que o OA não possui defeitos técnicos ou problemas no conteúdo pedagógico.
- **Portabilidade:** indica se o OA pode ser transferido (ou instalado) para diferentes ambientes, como, por exemplo, diferentes tipos de AVAs ou sistemas operacionais.
- **Facilidade de instalação**: indica se o OA pode ser facilmente instalado caso ele exija esse recurso.
- **Interoperabilidade:** medida de esforço necessário para que os dados dos OAs possam ser integrados a vários sistemas.
- **Usabilidade**: indica a facilidade de utilização dos Oas por alunos e professores.
- **Manutenibilidade:** é a medida de esforço necessária para alterações do OA.
- **Granularidade:** de maneira geral, a palavra granularidade origina-se da palavra grão, sendo que quanto maior o número de grãos de um sistema maior a sua granularidade. Trazendo esse conceito para o âmbito dos objetos de aprendizagem, a granularidade é a extensão à qual um OA é composto por componentes menores e reutilizáveis.
- **Agregação:** indica se os componentes do OA (grãos) podem ser agrupados em conjuntos maiores de conteúdos como, por exemplo, as estruturas tradicionais de um curso.
- **Durabilidade:** indica se o OA se mantém intacto quando o repositório em que ele está armazenado muda ou sofre problemas técnicos.
- **Reusabilidade:** indica as possibilidades de reutilizar os OAs em

  diferentes contextos ou aplicações. Essa é a principal característica do OA e pode ser influenciada por todas as demais.

Dalziel (2002) cita que entre os passos de criação e armazenamento de um OA em um determinado banco de dados (repositório), questões como a administração das licenças de uso e dos direitos autorais devem ser abordadas em um estágio intermediário. O autor salienta que essas questões serão posteriormente utilizadas durante o estágio de busca e recuperação/entrega dos recursos em que as licenças e as condições de uso deverão ser aceitas por alguém se o mesmo deseja utilizar os materiais recuperados. O autor também descreve cinco atores diferentes envolvidos no ciclo de vida de um objeto de aprendizagem, sendo eles:

1. **Autoridade**: responsável por prescrever os objetivos de aprendi zagem e resultados;

2. **Criador:** o autor do objeto de aprendizagem e/ou responsável por submeter o mesmo para a publicação;

3.**Organizador:** responsável por projetar atividades de aprendizagem e revisar as licenças e direitos autorais e de uso;

4. **Buscador de Informação (*Infoseeker*)**: tem o papel de buscar por recursos de acordo com os metadados fornecidos;

5. **Aprendiz:** aquele que irá utilizar os OAs e realizar as avaliações.

Com a diversa gama de recursos tecnológicos existentes, muitas vezes pode-se confundir os conceitos de Objetos de Aprendizagem e Recursos Educacionais Abertos. O conceito de recursos educacionais abertos (REA) foi definido pela Unesco como “qualquer material suportado por mídia que esteja sob domínio público ou com uma licença aberta e que possa ser utilizado e adaptado por terceiros” (RNP, 2019).

“Para ser considerado um REA, é importante que os materiais possam ser retidos, reutilizados, revisados, recombinados e distribuídos sem necessidade de autorização adicional dos autores” (RNP, 2019).

De acordo com os principais conceitos abordados sobre OAs, que também são recursos, digitais ou não, com fins educacionais. Enfatiza-se que a principal diferença entre esses dois conceitos está no formato aberto, tendo em vista que “os REA precisam estar totalmente disponíveis para acesso e edição. Podendo citar alguns

exemplos de formatos abertos sólidos como o PNG para imagens, o WebM para vídeo e o HTML para páginas web” (RNP, 2019).

2.2 Padrões de Identificação de metadados

Um aspecto muito importante para a criação e disponibilização de objetos de aprendizagem em ambientes virtuais é a sua padronização por meio de metadados, que serve para disponibilizar o objeto de forma detalhada de acordo com seus principais aspectos descritivos e informativos, a partir da importância e necessidade dessa padronização, podemos definir alguns conceitos para exemplificar o que o metadado representa.

> Metadado é frequentemente identificado como dado acerca de dado, ou informação sobre informação que está no espaço digital e virtual. É um sumário de informações sobre a forma e o conteúdo de um recurso eletrônico, ou não, que pode ser um objeto bibliográfico (livros, seriados, mapas, etc.), catálogo de registros bibliográficos, inventários e registros de arquivos, objetos geoespaciais (imagens de satélites, etc) , recursos de museus e visuais, ou implementações de software (Ercegovac, 1999).

A biblioteca digital de teses da USP utilizou os conceitos de descrição de metadados para a descrição bibliográfica dos seus documentos, no trabalho é apresentado algumas definições sobre metadados e como foi feita a descrição dos dados no referido trabalho, em (ROSETTO, 2002) também podemos encontrar esse conceito sobre definição de metadados:

> Dempsey & Heery (Sutton, 1999), também afirmam que os metadados podem ser informações descritas tradicionalmente (ex: catálogos de bibliotecas), ou informações de recursos eletrônicos (arquivos automatizados, textos eletrônicos ou páginas web), cuja forma de recuperação está baseada nos motores de busca da web para a localização. Da mesma forma que um catálogo de biblioteca fornece informação para localizar um livro, um metadado é usado para localizar uma informação (registro), que poderá ser um endereço Internet, recursos web, livros, e-mail, eventos, artefatos, conjunto de dados, projetos, organizações, etc. É um poderoso instrumento normativo para identificar e descrever recursos de informação e auxiliar os pesquisadores a localizarem e obterem essas informações.

O conjunto de elementos tem o objetivo principal de representar o conteúdo do recurso descrito, ou seja, as informações necessárias para possibilitar a identificação do elemento, e qual o seu conteúdo. Cada conjunto de elementos pode variar de acordo com o padrão de metadados que é utilizado, e para uma descrição clara e objetiva dos objetos, os elementos devem conter alguns tipos de informações, que de acordo com Gilliland-Swetland (1998) os metadados podem ser divididos em 5 tipos de acordo com os aspectos de sua funcionalidade em um sistema digital:

- **Administrativos**: Usados na gestão e administração de recursos de informação.
- **Descritivos:** Usados para descrever informação sobre recursos.
- **De conservação:** Relacionados com a conservação de recursos de informação.
- **Técnicos:** Relacionados com o funcionamento dos sistemas e o comportamentos dos metadados.
- **De uso:** Relacionados com o nível e o tipo do uso dos recursos de informação.

Segundo GILLILAND-SWETLAND (1998), os metadados devem ser elaborados cuidadosamente, seguindo padrões nacionais ou preferencialmente internacionais. O objetivo é permitir o tratamento adequado das informações contidas em um recurso, permitindo maior acessibilidade a ele, mantendo a relação entre um recurso digitalizado e o original, podendo descrever direitos e restrições do recurso e conservando-o disponível, independente das alterações de software e hardware.

Gill (1998) aponta que qualquer solução para busca e recuperação de recursos na Web passa pela utilização de um catálogo distribuído e afirmava: *"*A existência de descritores consistentes, exatos e bem distribuídos dos recursos da Web permitirá maior precisão na busca e uma classificação mais rigorosa dos resultados obtidos segundo sua relevância*".*

Levando em consideração esses requisitos, é indispensável que o objeto garanta a sua reutilização, possua uma uniformidade na descrição do conteúdo e contenha um melhor acesso a mais conteúdos de mais fontes fazendo com que a sua busca seja facilmente identificada, tornando a implementação dos metadados extremamente importantes para a identificação, organização e recuperação das informações digitais. Para isso são utilizados padrões de construção, esses padrões possibilitam a inclusão do objeto em repositórios, tanto nacionais quanto internacionais, fazendo com que ele seja reutilizável.

Um padrão de metadados pode ser descrito como um conjunto de elementos descritores que segue um determinado modelo de dados com o objetivo de descrever recursos de um domínio específico. Podemos entender como um modelo de dados um conjunto de conceitos e regras (BARRETO apud GRACIO, 2002).

Esses padrões existem para que o objeto seja construído de forma eficaz, e possa contribuir de forma significativa para a aprendizagem da pessoa que irá utilizá-lo, pois de acordo com Kruchten (2003), um produto de qualidade deve ter ausência de defeitos e, principalmente, deve atender aos propósitos desejados. Os objetivos também devem ser atendidos da melhor forma tanto no quesito técnico quanto pedagógico.

Podemos citar a seguir alguns dos padrões mais aplicados:

- **IMS (Global Learning Consortium, Inc):** O Learning Design da IMS, ou IMS Learning Design, é um modelo para especificação de objetos e atividades de aprendizagem baseada no EML (Educational Modeling Language) da Universidade Aberta da Holanda (OUNL - Open Universiteit Nederland). O desenvolvimento do EML iniciou-se em 1998 com a idéia de criar uma modelagem que pudesse representar uma unidade de aprendizagem em sua totalidade, englobando não só o conteúdo como também os diversos processos envolvidos (Koper, 2002);

- **LOM (Learning Object Metadata):** O IEEE-LTSC *Learning Object Metadata Working Group* (LOM) tem o objetivo de especificar a sintaxe e semântica de metadados para *learning objects*. O padrão de metadados para learning objects compreende um conjunto de características que permitem o gerenciamento e busca de learning objects. As propriedades relevantes a serem descritas para os learning objects incluem tipo do objeto, autor, propriedade, termos de distribuição e formato. Quando aplicável, o metadado para learning objects pode incluir propriedades pedagógicas, tais como: tipo de interação, classificação do nível e pré-requisitos (IEEE, 2002).

- **ADL(Advanced Distributed Learning Initiative):** É uma organização composta por órgãos do governo norte-americano, vendo que existiam muitos padrões diferentes no mercado e que eles só tratavam de assuntos específicos, ela criou um modelo de referência juntando os padrões AICC, IMS, LOM e Ariadne, para tentar padronizar os padrões (Gonçalves, 2011);

- **SCORM (Sharable Content Object Reference Model):** O SCORM é um modelo de referência, ou seja, conjunto unificado de especificações para a disponibilização de conteúdos e serviços de *e-learning.* Este conjunto de especificações define um modelo de agregação de conteúdo, um modelo de sequenciamento e um ambiente de execução para objetos de aprendizagem baseados na web (ADL, 2004);

- **Dublin Core Metadata Element Set (DCMES):** O Dublin Core é uma especificação de metadados, definido pelo padrão NISO Z39.85-2007, que é especialmente usado para descrever materiais digitais como, por exemplo, vídeo, som, imagem, texto e páginas web, por ter um conjunto de atributos simples e robusto, o qual ajuda na hora de recuperar alguma informação ou na hora de catalogar esses objetos em um repositório. Esse padrão se caracteriza pela sua estrutura e flexibilidade, podendo ser usado para descrever qualquer tipo de recurso. (Gonçalves, 2011).

2.3 Padrão de Metadados Dublin Core

O padrão Dublin Core (DC) se originou em 1994, na 2ª conferência internacional *World Wide Web* e visa descrever diversos elementos digitais, especificamente elementos na web, tais como imagens, vídeos, sons, textos, sites, softwares e jogos educacionais. Suas aplicações utilizam XML e o RDF *(Resource Description Framework)*, tendo como principal objetivo promover a interoperabilidade dos metadados, fazendo com que o retorno dos dados em sistemas de busca se tornem mais ágeis e flexíveis. Atualmente o padrão é dividido em dois níveis, o simples e o qualificado. O simples é constituído de 15 elementos, já o qualificado possui 3 elementos adicionais (Audiência, Proveniência e Detentor de Direitos).

Dos padrões de metadados apresentados na seção 2.2, o Dublin Core é o padrão mais aplicável, pois ele foi desenvolvido com o objetivo principal de trabalhar com os recursos que são disponibilizados na web, onde atualmente há um grande número de informações que são disponibilizadas de forma despadronizada dificultando a busca de dados específicos.

O DC possui algumas características, que equivalem também os seus objetivos, dentre os quais são as seguintes (Grácio 2012; Hillman 2005a):

- **Simplicidade** – O conjunto de elementos DC tem sido mantido pequeno e simples, de modo que sejam compreensíveis semanticamente. Um pesquisador ou responsável da coleção, não especialista, pode gerar facilmente registros descritivos para os recursos de informação.

- **Interoperabilidade semântica** – A existência de esquemas distintos de descrição intervém na capacidade das pesquisas e das descobertas entre áreas. Um conjunto de elementos comum, universalmente inteligível e apoiado, propicia maior interoperabilidade entre estas áreas.

- **Consenso internacional** – A participação de especialistas, órgãos, ou representantes de vários países na DCMI (*Dublin Core Metadata Initiative*) para fins de internacionalização na Web e de uma estrutura apropriada (que reflita a natureza multilíngue e pluricultural do universo da informação eletrônica), colabora para um consenso internacional e uma contínua melhora do DC (junto ao avanço da Web).

- **Extensibilidade** – O padrão DC é um esquema simplificado de descrição, que na criação de esquemas, permite conjuntos de metadados adicionais para complementar as demandas de descrição e recuperação precisa de um recurso digital particular numa determinada comunidade. Estes elementos de metadados adicionais junto com os elementos DC, possibilitam que múltiplas comunidades em várias áreas usem o padrão DC, permutem e acessem as informações, podendo ter acesso a elas no guia do usuário.

- **Flexibilidade** – Os elementos são opcionais e repetíveis, não há ordem definida e podem ser alterados adotando um conjunto limitado de qualificadores, que são atributos utilizados para refinar (não estender) o significado do elemento.

Para que se possa entender o funcionamento do padrão de metadados DC é feita uma análise da versão 1.1, composta de 15 elementos. Essa versão é uma iniciativa de recomendação de metadados, e o DCMI (DCMI, 2001) entende por recomendação que as especificações são fixas e suportadas para adoção da comunidade DC. Na figura 1 podemos observar o conjunto de elementos definidos pelo Dublin Core para a descrição dos metadados em sua versão simples.

Os elementos são descritos de acordo com os padrões da ISSO/IEC 1117 (DCMI, 1999) e as definições dos elementos do padrão de metadados DC estão definidas na RFC 2413 (IETF, 2019).

• Título	- um título dado ao recurso
• Criador	- uma entidade principal responsável pela elaboração do conteúdo do recurso
• Assunto	- assunto referente ao conteúdo do recurso
• Descrição	- uma descrição sobre o conteúdo do recurso
• Editor	- a instituição responsável pela difusão do recurso
• Contribuinte	- uma entidade responsável pela contribuição ao conteúdo do recurso
• Data	- data associada com um evento no ciclo de vida do recurso
• Tipo	- a natureza ou gênero do conteúdo do recurso
• Formato	- manifestação física ou digital do recurso
• Identificação	- identificação não ambígua do recurso dentro de um dado contexto
• Fonte	- uma referência para um outro recurso o qual o presente recurso é derivado
• Idioma	- idioma do conteúdo intelectual do recurso
• Relação	- uma referência a um outro recurso que se relaciona com o recurso
• Cobertura	- a extensão ou cobertura espaço-temporal do conteúdo do recurso
• Direitos	- Informações sobre os direitos do recurso e seu uso

Figura 1: Elementos de metadados do padrão DC Simples
Fonte: Dublin core (1999)

Na Tabela 1 a seguir é apresentado um exemplos de aplicação do padrão Dublin Core em sua forma simples com 15 elementos padrões em uma página web, ela tem a referência para uma descrição de uma página web, onde foi depositado um objeto de robótica educacional para construção de uma Guitarra a partir de um kit de robótica da Lego Mindstorms, a partir da descrição dos dados na página foram extraídos os principais pontos onde se encaixam no padrão e feita a tabela de metadados.

Tabela 1 - Descrição de um objeto de robótica disponível no site nxtprograms
Fonte: Autoria Própria

Identificador	Conteúdo
dc.title	Guitar Challenge game
dc.creator	Dave Parker
dc.subject	Nxt, lego, guitar
dc.description	This upgraded version of the Electric Guitar project allows to you play in two different modes. In "Free Play" mode, you can play tones or chords similar to the Electric Guitar project, except that the guitar is upgraded to use a touch sensor lever for the "strum" sensor, and also includes a "whammy bar" so that you can add whammy effects to your guitar notes when playing tones.
dc.publisher	NXT PROGRAMS
dc.contributor	
dc.date	2011
dc.type	Interactive Resource
dc.format	Text/html
dc.identifier	https://www.guitarhero.com/game
dc.source	http://www.nxtprograms.com/NXT2/guitar_game/index.html
dc.language	en-us(Inglês-Estados Unidos)
dc.relation	É um jogo de Guitar Hero com lego
dc.coverage	2007-2011
dc.rights	Copyright © 2007-2011 by Dave Parker. All rights reserved.

O Dublin Core é um padrão de descrição de objetos de aprendizagem muito utilizado pois facilita o processo de descrição dos objetos digitais, sendo altamente relevante, pois dentre suas principais características, podemos destacar a simplicidade da utilização dos recursos e também o fato de sua extensibilidade ser bastante acessível e descomplicada, ou seja ele oferece uma melhor adaptação às necessidades do usuário. Vale destacar que como o Dublin Core é adaptável a cada tipo de situação, os identificadores podem aparecer mais de uma vez, como também alguns elementos podem não aparecer na descrição do conteúdo.

O padrão DC tem sido amplamente utilizado como o padrão de descrição de metadados na web, mas dificilmente é aplicado apenas no seu formato padrão com seus 15 elementos. Sempre as entidades que utilizam o padrão estão acrescentando identificadores e elementos extras para descrição dos seus recursos eletrônicos, dependendo da aplicação desejada.

> Contudo, com a evolução do padrão, os 15 elementos de descrição se tornaram mais completos com o surgimento dos qualificadores. Estes qualificadores têm a função de refinar ou tornar mais específico o recurso descrito. Portanto, é decisão do sistema sobre qual formato utilizar (simples ou qualificado) (PIRES, 2012).

Os usuários podem ter a livre escolha de utilizar ou não os qualificadores, podendo também desenvolver os seus próprios qualificadores para sua aplicação. Porém os qualificadores desenvolvidos para cada aplicação não podem ser reutilizados por outras comunidades.

De acordo com (DCMI, 2019) , foram definidos duas classes de qualificadores:

- Elemento de refinamento: esses qualificadores dão mais especificidade a um elemento, detalhando-o melhor.
- Esquema de codificação: esses qualificadores identificam esquemas para o valor do elemento. Incluem vocabulário controlado e notações formais de representação (sistemas de classificação).

Podemos citar algumas aplicações conhecidas que utilizam esse padrão para a descrição dos seus objetos, como por exemplo o Banco Internacional de Objetos

Educacionais do MEC (MEC, 2019), RIVED – Rede Interativa Virtual de Educação (RIVED, 2019), Repositório de objetos de aprendizagem da EAD- UFSC (UFSC, 2019), elas podem ser classificadas como bases de objetos educacionais.

2.4 Bases de objetos educacionais

Os objetos de aprendizagem estão espalhados por toda internet de forma despadronizada, fazendo com eles sejam encontrados em uma enorme variedade de páginas e locais. Tendo em vista essa diversidade de locais para a localização de objetos, foram criados os ROAs (Repositórios de objetos de aprendizagem), se tornando o local mais adequado para realizar a busca por OAs, que tem como principal objetivo facilitar a localização e o reuso de OAs, os repositórios não armazenam apenas os objetos educacionais, mas também os metadados, que funcionam como descritores estruturais e semânticos para esses objetos.

Uma grande vantagem que podemos destacar em realizar a busca de um OA em um ROA, é que nele estão contidas as informações técnicas e pedagógicas, seguindo um padrão específico e aumentando a reusabilidade do recurso educacional pela comunidade. As bases de objetos educacionais ou ROAs são fundamentais para que sejam armazenados os recursos a serem disponibilizados, também são responsáveis pela disponibilização das referências que permitem a localização dos objetos.

Após o desenvolvimento de um OA ser finalizado, ele precisa ser depositado em um ROA, sendo disponibilizado aos usuários para que possa ser buscado, utilizado e compartilhado. Heery e Anderson (2005) citam em seu trabalho algumas características que diferem os ROAs de outras coleções digitais:

- O conteúdo pode ser depositado no repositório, seja pelo criador do conteúdo, proprietário ou terceiros.
- Possuir uma arquitetura capaz de administrar conteúdos e metadados.
- Deve fornecer um conjunto mínimo de serviços básicos, como por ex. inserir, buscar, pesquisar, além de serviços de controle de acesso.
- Ser confiável, bem suportado e bem administrado.

Heery e Anderson (2005) também citam algumas características dos repositórios de acesso aberto:

- O repositório deve fornecer acesso aberto ao seu conteúdo (a menos que haja restrições)
- Repositório deve fornecer acesso aberto aos seus metadados para coleta

O trabalho supracitado também destaca que os repositórios formam uma intersecção de interesse para diferentes comunidades: Bibliotecas digitais, pesquisa, aprendizagem, e-ciência, publicação, gerenciamento de registros, preservação. Dentro dessas comunidades, destacando as principais áreas funcionais:

- Acesso aprimorado aos recursos.
- Novos modelos de publicação e revisão.
- Gerenciamento de informações corporativas (gerenciamento de registros e gerenciamento de conteúdo).
- Compartilhamento de dados (reutilização de dados de pesquisa, reutilização de objetos de aprendizagem).
- Preservação de recursos digitais.

McGreal (2008) fala que os ROAs podem ser categorizados com base em fatores como localização de OAs, especificidade da área, fornecimento de cursos completos, requisitos para participação e uso, e para cada tipo de ROA descrito no seu trabalho ele cita algumas características:

- **Nível do público-alvo:** se o repositório está enfocado em um nível educacional específico ou não.
- **Granularidade dos materiais:** componentes, cursos, lições,recursos;
- **Tamanho do repositório em termos do número de recursos armazenados;**
- **Tipo dos materiais:** se existe um tipo predominante de OA (ex.:applets, vídeos, livros digitais etc.), ou se os materiais estão em formatos variados;

- **Tipo do metadado utilizado:** Dublin Core Metadata, IEEE-LOM, CanCore, alguma taxonomia específica, ou nenhum.

A partir dos principais conceitos apontadas neste tópico pudemos observar os princípios básicos dos ROAs, seus objetivos fundamentais e formas de desenvolvimento, vimos que os repositórios possuem como principais funcionalidades a colaboração das informações, a busca pelos objetos a partir de palavras chave ou tipos de objetos, e a sua visão e implementação de metadados a partir de um padrão específico.

3 ROBÓTICA EDUCACIONAL

O principal objetivo desse capítulo é abordar os principais conceitos e definições da robótica educacional, será feito uma breve descrição dos kits de robótica educacional mais utilizados.

3.1 Conceito de Robótica Educacional

A Robótica pode ser definida como “a ciência dos sistemas que interagem com o mundo real com pouca ou mesma nenhuma intervenção humana” (ARS CONSULT, 1995, p.21). A robótica é uma área multidisciplinar, onde pode envolver diversas áreas, como a engenharia elétrica, engenharia mecânica e inteligência artificial. A robótica também agrega um papel muito importante na indústria de automação, pois essa parte da indústria é responsável pela fabricação de robôs capazes de substituir o ser humano em diversas atividades, ou até mesmo tarefas que o ser humano não consegue realizar.

“No início do século XX, pela necessidade do aumento da produtividade e melhoria da qualidade dos produtos, iniciou-se a construção dos robôs para as indústrias”. (GODOFREDO, ROMANÓ E ZILLI, 2001).

Podemos apresentar uma definição para o termo robô moderno sendo citados no trabalho de (ULRICH apud KLOC et al, 2009)como sendo “um equipamento multifuncional e reprogramável, projetado para movimentar materiais, peças, ferramentas ou dispositivos especializados através de movimentos variados e reprogramáveis”. Ou também “Diferentemente da automação convencional, os robôs são projetados para realizarem, dentro dos limites pré-definidos, um número irrestrito de diferentes tarefas”.

O autor também enfatiza que “os robôs podem ser equipados com sensores para sentir ou perceber calor, pressão, impulsos elétricos e objetos e podem ser usados com sistemas de visão rudimentares”. Dessa forma, podem monitorar as tarefas que realizam. Podem também aprender e se lembrar das tarefas, reagir ao seu ambiente de trabalho, operar outras máquinas e se comunicar quando ocorrem problemas no

seu funcionamento. É uma tecnologia que pode levar à reformulação da maneira de pensar e trabalhar.

Quando se pensa em robótica, automaticamente remete-se esse pensamento para áreas de engenharia elétrica, engenharia mecânica ou mecatrônica, ou seja, áreas de ensino superior. Dificilmente esse pensamento está relacionado a uma ferramenta capaz de auxiliar o aprendizado entre professores e alunos no âmbito educacional. A partir desse pensamento podemos definir o conceito de robótica educacional, que teve sua origem como uma ferramenta que “permite ao professor demonstrar na prática muitos dos conceitos teóricos, às vezes de difícil compreensão”. Motivando o aluno, que a todo momento é colocado em situações de desafios com o objetivo de observar, abstrair e inventar. “Utiliza-se dos conceitos de diversas disciplinas (multidisciplinar) para a construção de modelos, levando o educando a uma gama enorme de experiências de aprendizagem” (BESAFE apud ZILLI, 2004).

A robótica educacional surgiu como um meio de possibilitar o desenvolvimento de projetos tecnológicos dentro e fora do ambiente de ensino superior, onde pode envolver técnicas de construção e manipulação de robôs e viabilizar o desenvolvimento do processo criativo, raciocínio lógico, a interdisciplinaridade em diferentes áreas.

> Robótica educativa é definida como sendo o controle de mecanismos eletro-eletrônicos através de um computador, transformando-o em uma máquina capaz de interagir com o meio ambiente e executar ações definidas por um programa criado pelo programador a partir destas interações (Zilli, 2004).

O autor afirma que com a robótica educacional, “o aluno passa a construir seu conhecimento através de suas próprias observações e aquilo que é aprendido pelo esforço próprio da criança tem muito mais significado para ela e se adapta às suas estruturas mentais”.

O mesmo autor também afirma que a utilização da robótica na educação veio, a princípio, expandir o ambiente Logo de aprendizagem (PAPERT 1994). “Esse novo recurso permite que haja a integração de diversas disciplinas e a simulação do método científico, pois o aluno formula uma hipótese, implementa, testa, observa e faz as devidas alterações para que o seu “robô” funcione” (Maisonnette apud Zilli 2004).

O dicionário Dicionário Interativo da Educação Brasileira (2015) define robótica educacional como:

> Termo utilizado para caracterizar ambientes de aprendizagem que reúnem materiais de sucata ou kits de montagem compostos por peças diversas, motores e sensores controláveis por computador e softwares.

A robótica educacional (RE) é um paradigma recente. Os primeiros conceitos desse padrão surgiram com (PAPERT, 1986) em meados do ano de 1986 nos Estados Unidos. No cenário brasileiro, a RE começou a ser desenvolvida na segunda metade dos anos de 1990, pelas instituições de ensino UFRJ, UNICAMP e UFRGS, segundo relato de d'Abreu (2014).

> Na prática a ideia tomou forma na metade da década de 80 quando o Massachusetts Institute of Technology (MIT) realizou uma parceria com a Lego, empresa que fabrica blocos de montar para crianças. Aos conjuntos de construção da Lego foram acrescentados motores e sensores, permitindo às crianças construírem modelos que poderiam ser programados utilizando a linguagem Logo (RESMINI et al 2018).

3.2 Objetivos da Robótica Educacional

A partir dos conceitos apresentados anteriormente reflete-se que a RE não se restringe apenas a uma atividade educacional que desenvolve uma programação para o desenvolvimento e funcionamento de objetos físicos, ela possibilita a reflexão, a imaginação e a produtividade que é desenvolvida durante o processo de criação, podendo também desencadear o processo de atividades lúdicas possibilitando a montagem de diversos objetos por meio das peças que são disponibilizadas nos kits de montagem.

Pode-se definir uma classificação dos principais objetivos gerais, psicomotores, cognitivos e afetivos que são desenvolvidos com a RE (Godoy, 1997):

1. **Objetivos Gerais:**
   - Construir maquetes que usam lâmpadas, motores e sensores;
   - Trabalhar conceitos de desenho, física, álgebra e geometria;
   - Conhecer e aplicar princípios de eletrônica digital;
   - Construir ou adaptar elementos dinâmicos como engrenagens, redutores de velocidade de motores, entre outros.
2. **Objetivos Psicomotores:**
   - Desenvolver a motricidade fina;
   - Proporcionar a formação de habilidades manuais;
   - Desenvolver a concentração e a observação;
   - Motivar a precisão de seus projetos.

3. **Objetivos Cognitivos:**
    - Estimular a aplicação das teorias formuladas à atividades concretas;
    - Desenvolver a criatividade dos alunos;
    - Analisar e entender o funcionamento dos mais diversos mecanismos físicos;
    - Ser capaz de organizar suas idéias a partir de uma lógica mais sofisticada de pensamento;
    - Selecionar elementos que melhor se adequem à resolução dos projetos;
    - Reforçar conceitos de matemática e geometria;
    - Desenvolver noções de proporcionalidade;
    - Desenvolver noções topológicas;
    - Reforçar a aprendizagem da linguagem Logo;
    - Introduzir conceitos de robótica;
    - Levar à descoberta de conceitos da física de forma intuitiva;
    - Utilizar conceitos aprendidos em outras áreas do conhecimento para o desenvolvimento de um projeto;
    - Proporcionar a curiosidade pela investigação levando ao desenvolvimento intelectual do aluno.
4. **Objetivos Afetivos:**
    - Promover atividades que gerem a cooperação em trabalhos de grupo;
    - Estimular o crescimento individual através da troca de projetos e idéias;
    - Garantir que o aluno se sinta interessado em participar de discussões e trabalhos de grupo;
    - Desenvolver o senso de responsabilidade;
    - Despertar a curiosidade;
    - Motivar o trabalho de pesquisa;
    - Desenvolver a autoconfiança e a auto-estima;
    - Possibilitar resolução de problemas por meio de erros e acertos.

Zilli (2002) apresenta algumas outras competências que a RE pode desenvolver:

- 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;
- Capacidade crítica.

A partir dessas competências que foram apresentada, é importante salientar o grande potencial que a Robótica Educacional possui como ferramenta interdisciplinar, auxiliando o professor e o aluno no desenvolvimento de projetos e protótipos em sala de aula, fazendo com que se obtenha uma melhor absorção das informações e um melhor raciocínio na resolução de problemas.

3.3 Kits de Robótica Utilizados na Educação

Atualmente no mercado há várias empresas que comercializam kits de Robótica educacional. Esses kits são compostos por hardware e software, onde na parte do hardware os kits são constituídos por microcontroladores, manipuladores, rodas, motores e sensores, já o software possui as linguagem de programação, alguns kits utilizam a linguagem de programação textual, que é uma linguagem que é desenvolvida especificamente para o próprio kit, geralmente as linguagens textuais são baseadas em alguma linguagem já existente, como C ou Java.

Outro tipo de programação que é utilizado nos kits é a linguagem visual ou gráfica, onde os programas são desenvolvidos a partir da interação do usuário com os elementos do software, geralmente esses elementos são compostos por ícones,

imagens, símbolos e cores, esse tipo de interação torna a programação mais acessível para os usuários, facilitando o desenvolvimento de projetos e ideias.

A seguir serão mostrados os kits mais comercializados e popularmente conhecidos, como também algumas de suas principais características.

3.3.1 LEGO MINDSTORMS

As peças LEGO são produzidas pela empresa dinamarquesa Lego Group que existe desde 1949, tendo como principal foco os brinquedos de plástico que se encaixam e permitem diversas combinações. Em 1980 foi criado o departamento de produtos educacionais, e em 1989 o departamento de produtos educacionais mudou de nome para LEGO Dacta. A palavra "dacta" deriva da palavra grega "didático", que significa "o estudo da finalidade, meios e conteúdo de aprendizagem e o processo de aprendizagem". A colaboração entre o LEGO Group e o Massachusetts Institute of Technology deu início ao desenvolvimento de um “tijolo inteligente” que deu vida às criações LEGO através da programação de computadores. Dr. Seymour Papert tornou-se LEGO *Professor of Learning Research* após o seu trabalho fazer a associação da linguagem LOGO com os produtos LEGO (LEGO, 2019).

Na figura 5 pode-se visualizar uma evolução dos kits de robótica educacional fabricados pela LEGO:

1-LEGO MINDSTORMS RCX Intelligent Brick (1998)

2-Robotics Invention System 1.5 (1999)

3- Robotics Invention System 2.0 (2000)

4- Lego Mindstorms NXT 1.0 (2006)

5 – Lego Mindstorms NXT 2.0 (2009)

6- Lego Mindstorms EV3 (2013)

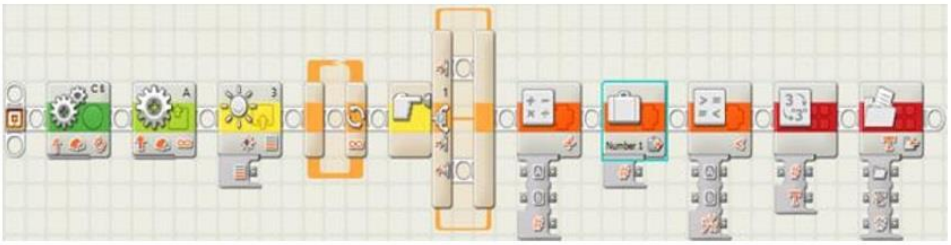
Figura 2: Evolução dos kits de robótica educacional LEGO

Fonte: Imagens adaptadas da internet, montagem de autoria própria

O primeiro kit de robótica educacional lançado no mercado foi o RCX (*Robotic Command Explore*) em 1988 (LEGO, 2019), com o lançamento do RCX houve um grande avanço na introdução de novas tecnologias em sala de aula, sua programação era feita utilizando o software Robolab, arrastando os blocos de comando do software e conectando com outros blocos, de acordo com a lógica inserida no programa, o robô iria obedecer os comandos desejados.

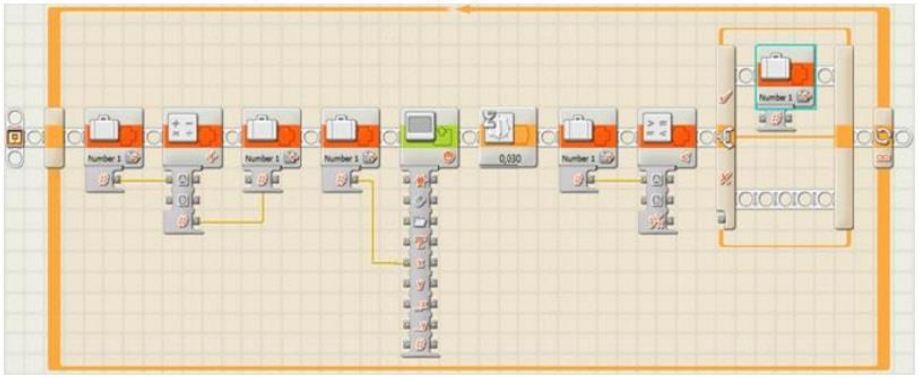
Em 2006, a lego decidiu inovar, fazendo uma parceria com o Media Lab, instituto de pesquisas do MIT, criando o kit Lego Mindstorms NXT, trazendo um diferencial da suas versões anteriores, com uma grande diversidade de sensores, utilizando um melhor processador e inserindo no mercado um novo software para realização da programação dos robôs, o novo software trouxe uma melhor interface gráfica e novas funcionalidades, possibilitando uma construção mais ampla de movimentos para os robôs, na qual a construção do software consiste em arrastar os blocos para a área de trabalho e configurá-los da forma desejada utilizando a lógica, podendo também ser programado em outras linguagens como C, C++ ou Java (LEGO, 2019).

Na figura 3 podemos ter uma visão do software da Lego referente ao modelo apresentado acima.



a) Blocos de Programação NXT 2.0

b) Painel de configuração NXT 2.0



C) Exemplo de codificação NXT 2.0

Figura 3: Exemplo de programação no software NXT
Fonte: Imagens adaptadas da internet, montagem de autoria própria

Em 2013 chegou ao mercado o Kit Lego Mindstorms EV3 (Terceira Versão), possuindo um processador e um hardware superior ao do NXT, processador Sitara AM1808 da Texas Instruments (núcleo ARM9 de 32-bits, instruções ARMv5 e clock de 300 MHz), em um de seus diferenciais, também está inserindo a possibilidade de conexão via Wi-Fi, e ele ganhou mais um OUTPUT que possibilita o controle de um motor a mais, trazendo também uma mudança em seu software, com um novo modelo de blocos e uma interface modificada como podemos ver na figura 4.

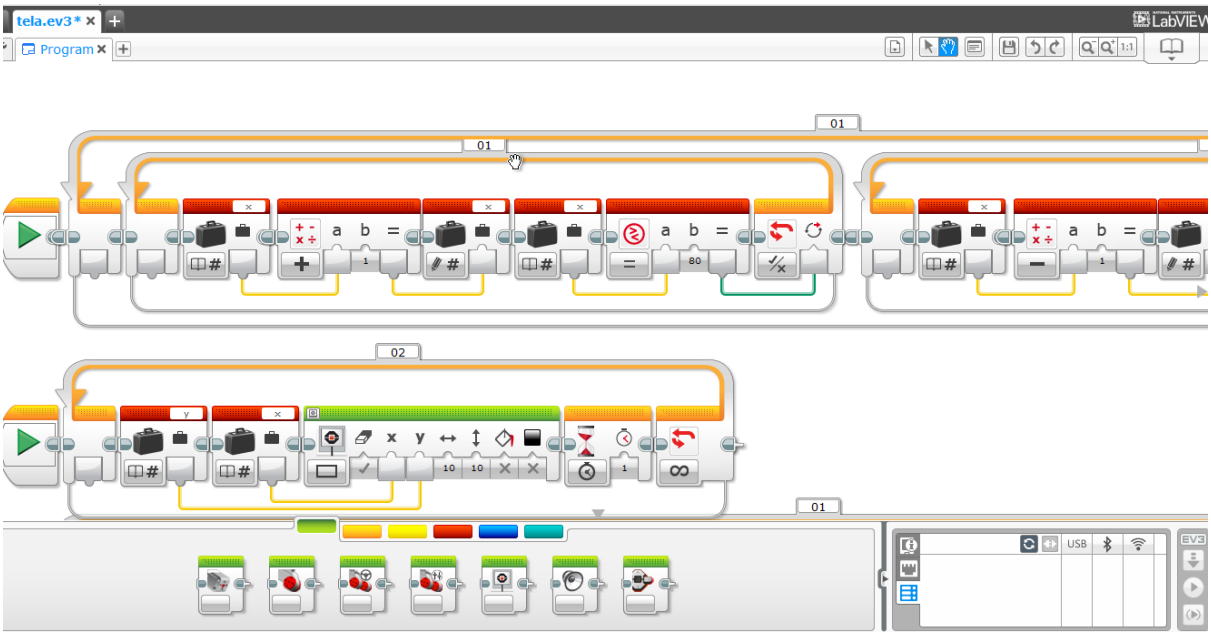


Figura 4: Interface de programação Lego EV3
Fonte: Veiga (2015)

Na figura 5 apresentada a seguir, pode-se ver um exemplo de programação de um robô seguidor de linha feito no software do NXT e no software do Lego EV3, nela pode-se perceber a mudança de um software para o outro, e como as duas programações são bastante semelhantes.

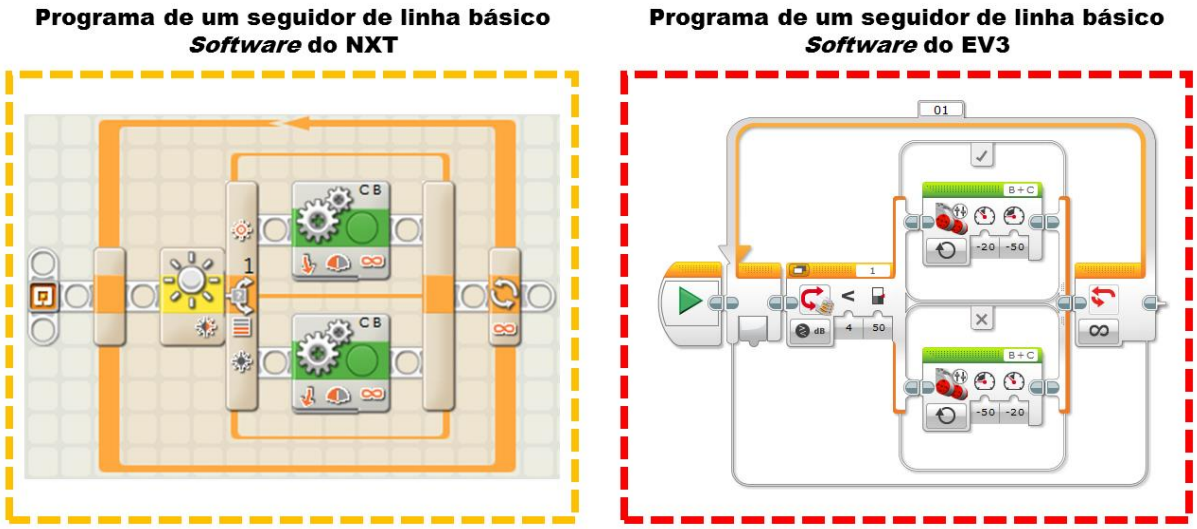


Figura 5: Programa seguidor de linha NXT e EV3
Fonte: Moreira (2015)

Em 2017 a Lego anunciou o Lego Boost, um robô com diferentes formas de montagem e programação, ele é composto por sensores de movimento, sensor de cor

e distância, e motores interativos, sua principal inovação foi a programação simples e intuitiva através do aplicativo *Free Lego Boost Creative Toolbox*, ele possui uma interface divertida. O aplicativo também conta com instruções de construção para os 5 modelos diferentes LEGO BOOST, bem como 60 atividades lúdicas para facilitar a aprendizagem, veja na figura 6 e na figura 7 a representação da interface do aplicativo, bem como é exposta a sua utilização.

Figura 6: Aplicativo Free Lego Boost Creative Toolbox
Fonte: Lego Boost (2019)

Figura 7: Modelo Vernie The Robot LEGO BOOST
Fonte: Lego Boost (2019)

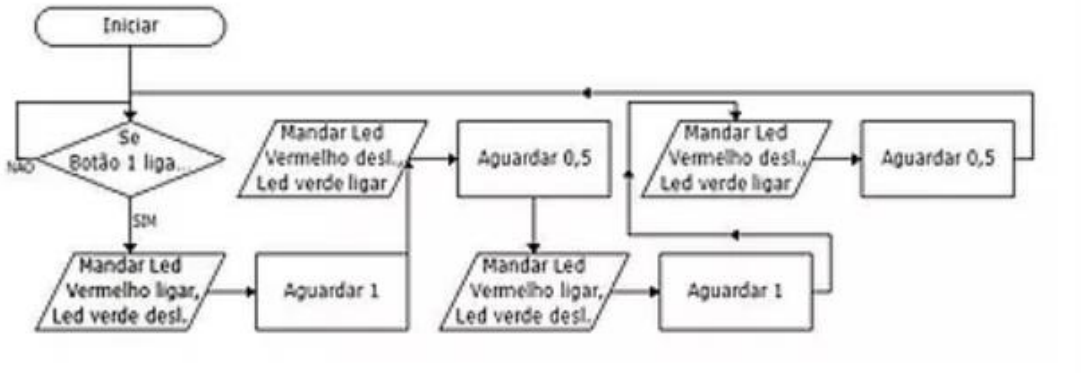
3.3.2 Modelix Robotics

A Modelix Robotics é uma empresa brasileira que disponibiliza kits de robótica educacional atendendo diferentes níveis de escolaridade. O kit Modelix é composto por microcontrolador, peças mecânicas (engrenagens, polias, parafusos e hastes), componentes eletrônicos (sensores e display LCD), motores, botões, sinalizadores luminosos e de som. Os protótipos elaborados a partir do Modelix podem ser controlados por infravermelho (utilizando controle remoto) ou por tecnologia bluetooth. Um exemplo de kit Modelix é apresentado na figura 8. A programação de funcionamento é criada utilizando o software Modelix System. De acordo com Modelix (2019) ele é um software desenvolvido especialmente para o ensino de programação de robótica.

A programação no Modelix System é intuitiva e é feita através de fluxograma, possui ferramentas que vão desde as mais básicas até ferramentas mais avançadas. Dessa maneira é possível atingir crianças de 7 anos até o nível universitário. A facilidade na aprendizagem é o primeiro passo para entrar no mundo da programação, o foco desse software é justamente ser esse passo inicial para o aluno desenvolver raciocínio lógico e começar a programar, e o mais interessante, não é necessário nenhum conhecimento técnico para usar o software Modelix System.

O software ainda conta com 2 versões de utilização para programação, a versão com microcontrolador (figura 9) e a versão com cenários interativos (figura 10) (MODELIX, 2019). Em ambos, a programação é feita da mesma forma, o que difere é que a versão do microcontrolador precisa do componente físico, nela pode-se montar um robô e ver a programação atuar sobre ele, já a versão dos cenários, não precisa de nenhum componente físico, tudo que se programa vai estar na tela, bem como o resultado da programação.

Figura 8: Kit de Robótica Modelix Ensino Médio e Superior
Fonte: Modelix (2019)



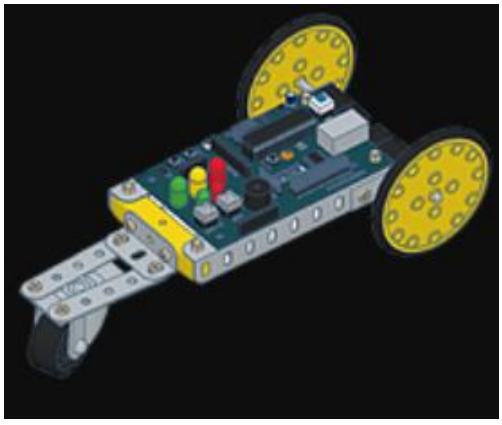


Figura 9: Modelix com Microcontrolador
Fonte: Modelix (2019)

No modo de programação utilizando o microcontrolador, o aluno irá programar o que quer que seu projeto execute. O software possibilita o aluno detectar possíveis

erros de programação, podendo assim, avaliar as melhorias a serem feitas no seu projeto. Após o desenvolvimento ele verá o que programou no projeto físico, ou seja, em seu robô.

Já na forma de simulação de cenários que simulam ambientes reais como, por exemplo, a casa automática ou um trem de passageiros que será controlado por uma rotina de programação feita pelo aluno, dispensando o uso de componentes físicos.

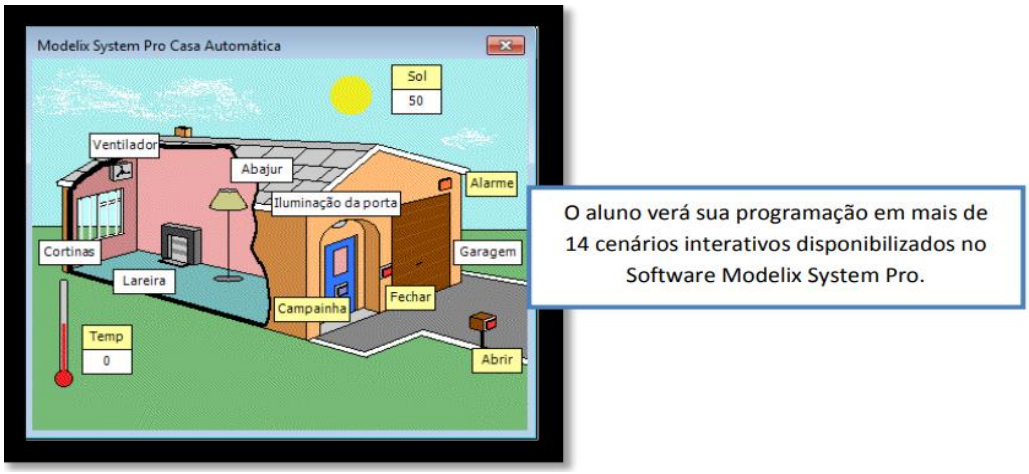


Figura 10: Exemplo cenário interativo Modelix
Fonte: Modelix (2019)

3.3.3 Arduino

O Arduíno é uma plataforma *Open Source* (Código Aberto) bastante utilizada na robótica educacional de baixo custo. Ele surgiu na Itália em 2005 com intuito de viabilizar desenvolvimento de projetos que utilizassem conceitos de circuitos eletrônicos e programação de forma mais simples. (Monk, 2014):

> “é uma pequena placa de microcontrolador que contém uma conector USB que permite ligá-la a um computador, além de possuir diversos pinos que permitem a conexão com circuitos eletrônicos externos, como motores, relés, sensores luminosos, diodos laser, alto falantes, microfones, etc.”.

Como o projeto Arduino é livre, existem inúmeros modelos de placas disponíveis no mercado, a figura 11 apresenta uma tabela comparativa de alguns modelos existentes.

- **Arduino UNO**

Possui processador ATMEGA328, 14 portas digitais, sendo que 6 delas podem ser usadas como saídas PWM, e 6 portas analógicas. A alimentação (selecionada automaticamente), pode vir da conexão USB ou do conector para alimentação externa (recomendável 7 à 12 Vdc). A placa, na versão com soquete, permite a troca do chip microcontrolador ATMEGA328 facilmente em caso de dano ao microcontrolador ou se o mesmo for utilizado em projetos dedicados. Existe também a placa Arduino Uno versão SMD, com o microcontrolador soldado na placa (ARDUINO, 2019).

- **Arduino Mega2560**

O Arduino Mega 2560 é uma placa microcontroladora baseada no ATmega2560. Possui 54 pinos de entrada / saída digitais (dos quais 15 podem ser usados como saídas PWM), 16 entradas analógicas, 4 UARTs (portas seriais de hardware), um oscilador de cristal de 16 MHz, uma conexão USB, conector de alimentação, um conector ICSP, e um botão de reset. Ele contém tudo o que é necessário para suportar o microcontrolador. Basta conectá-lo a um computador com um cabo USB ou ligá-lo a um adaptador de CA-CC ou bateria para iniciar. A placa Mega 2560 é compatível com a maioria dos escudos projetados para o Uno e as antigas placas Duemilanove ou Diecimila, , é a placa recomendada para impressoras 3D e projetos de robótica (ARDUINO, 2019).

- **Arduino Leonardo**

O Arduino Leonardo é uma placa microcontroladora baseada no ATmega32u4 (datasheet). Possui 20 pinos de entrada / saída digitais (dos quais 7 podem ser usados como saídas PWM e 12 como entradas analógicas), um oscilador de cristal de 16 MHz, uma conexão micro USB, conector de alimentação, um conector ICSP e um botão de reset. Ele contém tudo o que é necessário para suportar o microcontrolador; Basta conectá-lo a um computador com um cabo USB ou ligá-lo a um adaptador de CA-CC ou bateria para iniciar (ARDUINO, 2019).

- **Arduino Due**

O Arduino Due é uma placa microcontroladora baseada no CPU Atmel SAM3X8E ARM Cortex-M3. É a primeira placa Arduino baseada em um

microcontrolador ARM de 32 bits. Possui 54 pinos de entrada / saída digitais (dos quais 12 podem ser usados como saídas PWM), 12 entradas analógicas, 4 UARTs (portas seriais de hardware), um clock de 84 MHz, uma conexão compatível com USB OTG, 2 DAC (digital para analógico) , 2 TWI, uma tomada de energia, um cabeçalho SPI, um cabeçalho JTAG, um botão de reset e um botão de apagar (ARDUINO, 2019).

- **Arduino Mega ADK**

O Arduino MEGA ADK é uma placa microcontroladora baseada no ATmega2560. Ele tem uma interface de host USB para se conectar com telefones baseados em Android, com base no IC MAX3421e. Possui 54 pinos de entrada / saída digitais (dos quais 15 podem ser usados como saídas PWM), 16 entradas analógicas, 4 UARTs (portas seriais de hardware), um oscilador de cristal de 16 MHz, uma conexão USB, conector de alimentação, um conector ICSP, e um botão de reset (ARDUINO, 2019).

- **Arduino NANO**

O Arduino Nano é uma placa pequena, completa e fácil de usar, baseada no ATmega328P (Arduino Nano 3.x). Tem mais ou menos a mesma funcionalidade do Arduino Duemilanove, mas em um pacote diferente. Falta apenas uma tomada de energia DC e funciona com um cabo USB Mini-B em vez de um cabo padrão (ARDUINO, 2019).

- **Arduino Pro Mini**

O Arduino Pro Mini é uma placa microcontroladora baseada no ATmega328. Ele tem 14 pinos de entrada / saída digital (dos quais 6 podem ser usados como saídas PWM), 6 entradas analógicas, um ressonador integrado, um botão de reset e furos para montagem de cabeçalhos de pinos. Um conector de seis pinos pode ser conectado a um cabo FTDI ou a uma placa de fuga Sparkfun para fornecer energia e comunicação USB à placa. O Arduino Pro Mini é destinado a instalação semi permanente em objetos ou exposições. A placa vem sem cabeçalhos pré-montados, permitindo o uso de vários tipos de conectores ou solda direta de fios. O layout dos pinos é compatível com o Arduino Mini (ARDUINO, 2019).

- **Arduino Esplora**

Baseado no Arduíno Leonardo, utiliza o microcontrolador Atmega32U4, com clock de 16 Mhz e 32 KB de memória (4K usados pelo bootloader). Possui 2 conectores de 3 pinos, para conexão de módulos adicionais. A conexão com o microcomputador utiliza cabo micro-usb. (ARDUINO,2018).

o Arduino Esplora é uma placa diferente de todas as outras da família Arduíno, principalmente por possuir diversos sensores na sua construção. Nessa placa vem embutido um buzzer, um joystick, um potenciômetro deslizante, um sensor de temperatura, um acelerômetro, um led RGB, um sensor de luz (LDR), 4 push-buttons e um microfone. Além de tudo isso, ainda possui um soquete para tela LCD. É uma placa indicada para quem deseja aprender Arduíno sem se preocupar muito com eletrônica, uma vez que os componentes já estão embutidos na placa. Assim o usuário pode se concentrar na parte de programação (COELHO , 2015).

	Arduino Uno	Arduino Mega2560	Arduino Leonardo	Arduino Due	Arduino ADK	Arduino Nano	Arduino Pro Mini	Arduino Esplora
								
Microcontrolador	ATmega328	ATmega2560	ATmega32u4	AT91SAM3X8E	ATmega2560	ATmega168 (versão 2.x) ou ATmega328 (versão 3.x)	ATmega168	ATmega32u4
Portas digitais	14	54	20	54	54	14	14	-
Portas PWM	6	15	7	12	15	6	6	-
Portas analógicas	6	16	12	12	16	8	8	-
Memória	32 K (0,5 K usado pelo bootloader)	256 K (8 K usados pelo bootloader)	32 K (4 K usados pelo bootloader)	512 K disponível para aplicações	256 K (8 K usados pelo bootloader)	16 K (ATmega168) ou 32K (ATmega328), 2 K usados pelo bootloader	16 K (2k usados pelo bootloader)	32 K (4 K usados pelo bootloader)
Clock	16 Mhz	16 Mhz	16 Mhz	84 Mhz	16 Mhz	16 Mhz	8 Mhz (modelo 3.3v) ou 16 Mhz (modelo 5v)	16 Mhz
Conexão	USB	USB	Micro USB	Micro USB	USB	USB Mini-B	Serial / Módulo USB externo	Micro USB
Conector para alimentação externa	Sim	Sim	Sim	Sim	Sim	Não	Não	Não
Tensão de operação	5v	5v	5v	3.3v	5v	5v	3.3v ou 5v, dependendo do modelo	5v
Corrente máxima portas E/S	40 mA	40 mA	40 mA	130 mA	40 mA	40 mA	40 mA	-
Alimentação	7 - 12 Vdc	7 - 12 Vdc	7 - 12 Vdc	7 - 12 Vdc	7 - 12 Vdc	7 - 12 Vdc	3.35 - 12 V (modelo 3.3v), ou 5 - 12 V (modelo 5v)	5v

Figura 11: Tabela comparativa dos modelos de placas Arduino.
Fonte: Coelho (2015)

De acordo com (ALVES *et al*, 2012) o Arduino também envolve um ambiente de desenvolvimento integrado ao hardware (*IDE – Integrated Developement Environment*) para geração dos programas (denominados de sketches) que são enviados para a placa eletrônica. O IDE do Arduino foi desenvolvido em linguagem

JAVA baseado no projeto *Processing*[1], na biblioteca AVR-gcc (para microcontroladores da família AVR) e em outros softwares livres. A linguagem de programação do Arduino é baseada no projeto *Wiring*[2] e pode rodar nas plataformas Windows e Linux.

A figura 12 apresenta o ambiente de programação do Arduino, o código representa um simples programa onde está sendo feito um led ser ligado por um segundo e depois desligado por um segundo, fazendo isso repetidamente, gerando um efeito de pisca.

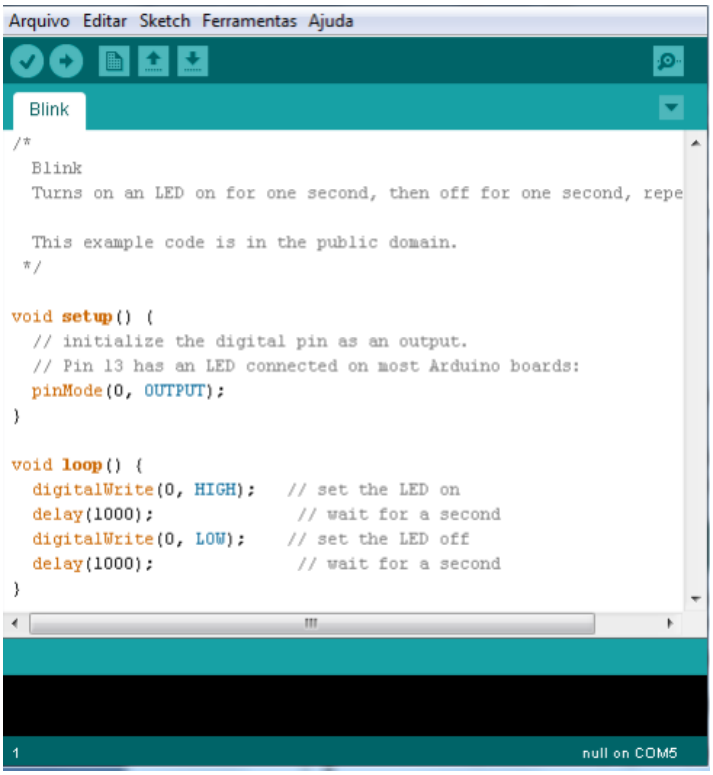


Figura 12: Ambiente de programação do Arduino
Fonte: Alves *et al* (2012)

3.3.4 VEX

O kit VEX é comercializado no Brasil desde 2007, pelo grupo INDEX tecnologia em robótica . Ele é composto por peças metálicas como vigas, porcas, parafusos, rodas, motores, sensores, arruelas. O software utilizado para construção da programação é o EasyC Vex Robotics (CABRAL, 2010). Os alunos podem aprender

[1] https://processing.org/
[2] http://wiring.org.co/

a programar com blocos, fazer a transição para o texto e passar para o C ++ com o avanço de suas habilidades (VEX, 2019). A figura 13 é representada por um dos modelos que a VEX comercializa, e a figura 14 mostra a interface de desenvolvimento da marca.

Figura 13: Modelo HEXBUG Vex IQ Robotics
Fonte: Amazon (2019)

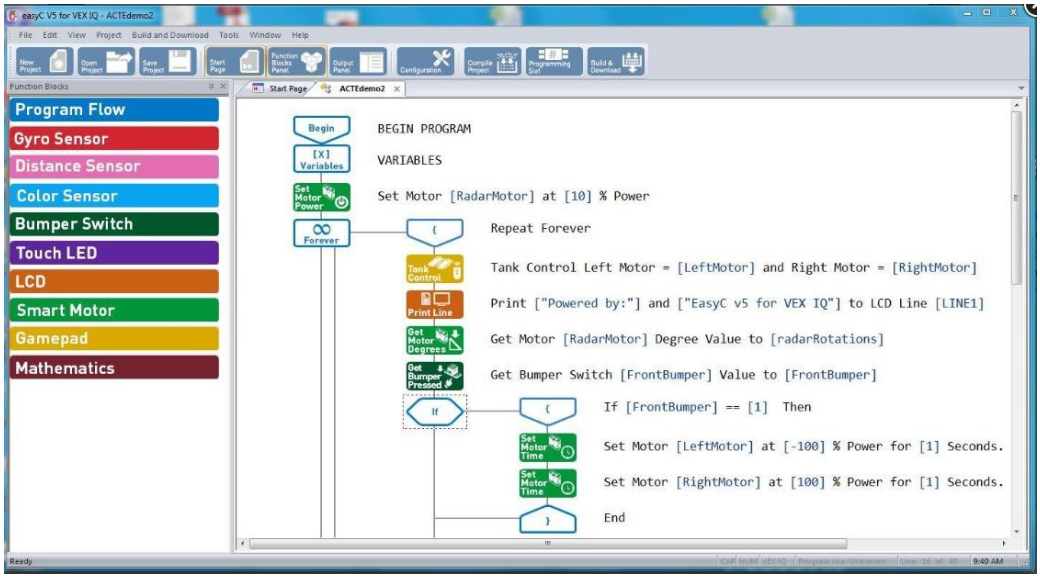


Figura 14: IDE EasyC Vex Robotics
Fonte: VEX IQ (2019)

3.3.5 Cyberbox

O Cyberbox é um kit educacional de robótica desenvolvido pela Besafe, empresa localizada em Curitiba, para uso de alunos do Ensino Fundamental, Médio e Superior. Ao kit da Besafe vêm inclusos: uma interface, fonte de alimentação dupla, cabo de comunicação, um motor DC, 10 metros de fios, 10 lâmpadas incandescentes, uma chave de fenda para manipulação dos contatos, 10 metros de fio para conexões e um cd-rom com manuais e software de controle (BESAFE, 2003). A interface Cyberbox

representada na figura 15 conecta-se à parte serial do computador. Ele também vem com uma proposta de utilizar sucata e outros materiais visando custo benefício.

O Cyberbox utiliza o software de autoria Everest para a sua programação e os baseados na linguagem Logo, como o SuperLogo, Micromundos e Imagine.

Figura 15: Cyberbox
Fonte: Chaves (2019)

3.3.6 GoGoBoard

A placa GoGo (figura 18) é um dispositivo de hardware de código aberto de baixo custo para robótica educacional, experimentos científicos e sensoriamento ambiental. As crianças podem usá-la para construir robôs, medir e registrar dados ambientais, conduzir investigações científicas, criar controladores de jogos, construir instalações artísticas interativas e muito mais, sua versão 5 lançada em 2015 integra-se com o *Raspberry PI* (Computador de placa única com conectividade LAN sem fio e Bluetooth) (GOGOBOARD, 2019).

Foi projetado no MIT e é apresentada como uma alternativa aos kits da Lego, porém de baixo custo especialmente para jovens de 10 a 18 anos e novatos em eletrônica. O princípio do design é permitir que os alunos passem o maior tempo possível sobre as idéias de sua criação desejada e menos sobre os detalhes técnicos da eletrônica de baixo nível envolvida. Os alunos podem encontrar-se rapidamente em contato com idéias poderosas em pensamento computacional e assuntos como STEM. Esse design distingue o GoGo Board de muitas alternativas, como o Arduino. Por exemplo, ao programar um LED, os alunos podem simplesmente conectar um módulo de LED e começar a criar padrões interessantes, em vez de ter que primeiro

montar um circuito de LED em uma placa de montagem. O mesmo princípio se aplica à linguagem de programação (GOGOBOARD, 2019).

Para usar o GoGo Board, o GoGo Widget (Figura 16) é necessário. É um programa que conecta a GoGo Board a um computador. Ele permite que você execute as seguintes ações:

- Exibir valores do sensor
- Controlar manualmente os motores
- Configure módulos adicionais e realize testes.
- Atualize o firmware da placa GoGo.
- Programe o GoGo Board usando uma linguagem de programação semelhante a um logo

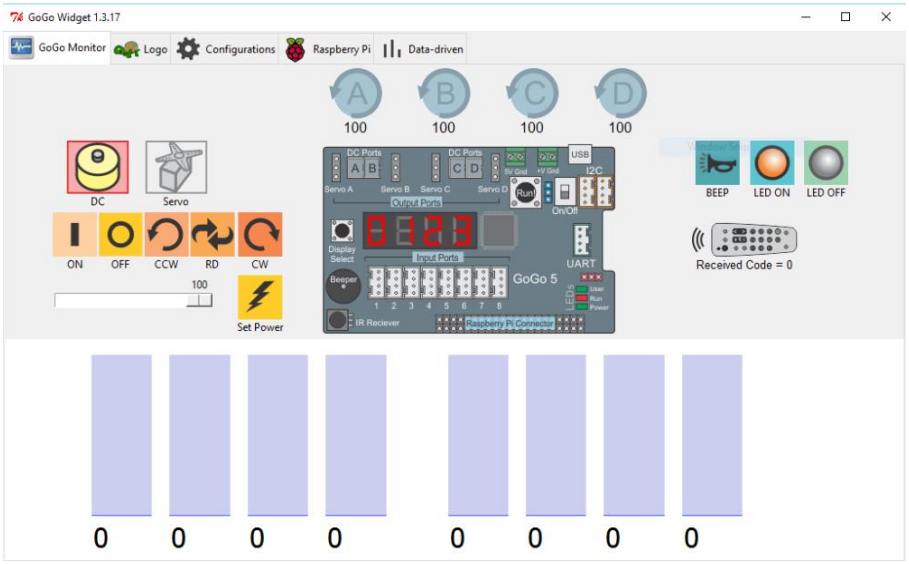


Figura 16: GoGo Widget

Fonte: GOGOBOARD (2019)

Tinker (figura 17) é um ambiente de programação *drag-and-drop* para a *GoGo Board* que é executado em um navegador. O Tinker é frequentemente usado por iniciantes que não estão familiarizados com o idioma do *GoGo Board.* Tinker traduz o programa visual em texto e envia para o *GoGo Widget*. Portanto, você precisa manter o *GoGo Widget* rodando para usar o Tinker.

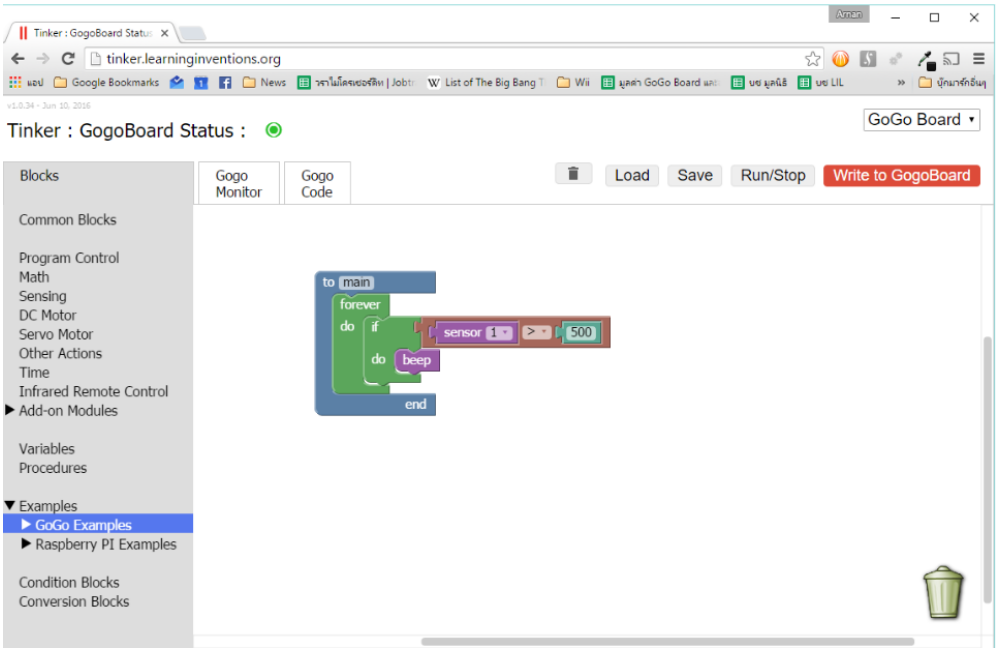


Figura 17: Tinker, ambiente de programação GoGo Board

Fonte: GOGOBOARD (2019)

Figura 18: Placa Go-Go Board
Fonte: GOGOBOARD (2019)

3.3.7 Super Robby

O kit Super Robby foi o primeiro kit de robótica educacional projetado e fabricado no Brasil. É composto de uma interface, que funciona como um tradutor entre o micro e os diversos dispositivos a ela conectados, como motores, sensores e lâmpadas. A programação do funcionamento da maquete ou protótipo pode ser feita através de uma linguagem de programação como as baseadas na linguagem Logo ou no software de autoria Everest (ARS, 2019).

O kit apresentado na figura 19 é composto por uma interface, uma fonte de alimentação, o software de simulação do funcionamento desta interface e alguns

componentes eletroeletrônicos. Possui 8 saídas de potência e 8 entradas digitais, ele pode ser utilizado com dois softwares, o Super Robby ou o MegaLogo.

Figura 19: Super Robby
Fonte: ARS (1995)

3.3.8 RoboEduc

A *RoboEduc* é uma empresa criada por pesquisadores da Universidade Federal do Rio Grande do Norte nas áreas de Educação e Computação, com foco na inovação da educação tecnológica, comprometida com o aprendizado de alunos desde o ensino infantil ao ensino superior, possibilitando que as crianças e jovens possam interagir com a realidade, desenvolvam habilidades para formular e resolver problemas, saiam da teoria para a prática usando ensinamentos obtidos em sala de aula, na interação cotidiana, nas amizades, nos princípios e valores da convivência (ROBOEDUC, 2019).

A *RoboEduc* comercializa um software educacional desenvolvido para funcionar como um mediador em atividades de robótica educativa, servindo não apenas para programação mas também para o design de robôs, foi desenvolvido pela equipe de robótica educacional do laboratório NatalNet, possuindo uma interface amigável para as crianças.

O *RoboEduc* dá suporte ao ensino de conceitos de robótica e programação de robôs, assistindo na construção, controle e programação de diversos modelos de protótipos de robôs, e permite o aprendizado de comportamentos usados para os robôs construídos (Barrios Aranibar *et al*, 2006).Outra característica importante do *software* segundo o autor, é que ele permite o movimento de um protótipo através do teclado e do mouse. O *software* foi desenvolvido em Linguagem C e XML, com total

execução dentro do RCX (neste caso autônomo) ou também executando em ambos RCX e computador (controlado).

Em sua primeira versão o RoboEduc implementa somente a funcionalidade de controle remoto, incluindo a comunicação necessária entre o sistema e o RCX, fazendo essa comunicação ser possível através da porta serial. Na Figura 20 podemos observar a versão 1.0 do software, onde é controlado através das teclas de navegação do computador.

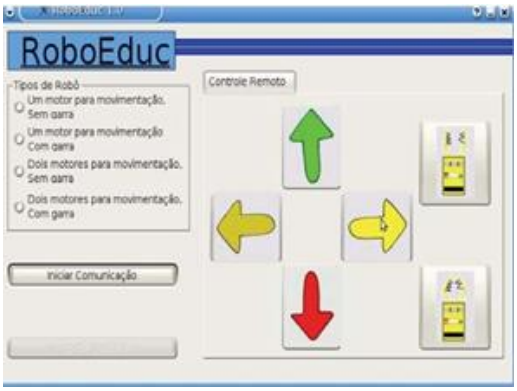
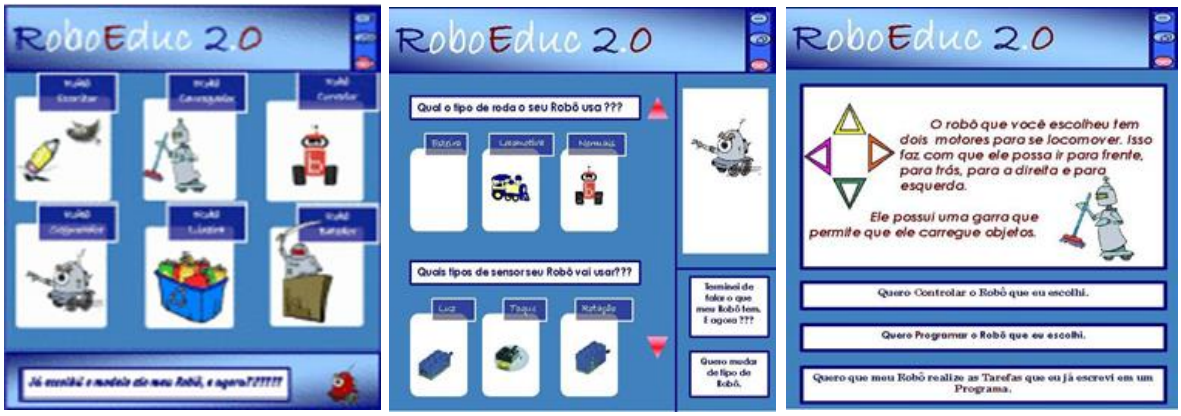


Figura 20: Tela RoboEduc 1.0
Fonte: Silva (2009)

Na figura 21 são apresentadas as telas do RoboEduc em sua segunda versão, onde foram acrescentadas novas funcionalidade como: Controlar protótipo, executar protótipo e executar programa aprendido. A comunicação passou a ser realizada pela porta usb.



a)Tela Inicial RoboEduc 2.0 **b) Tela Componentes do Robô** **c) Tela Funcionalidades**
Figura 21: Telas RoboEduc 2.0
Fonte: Adaptadas de Silva (2009)

Em sua terceira versão há uma diferença na interface em relação à versão anterior, acrescido da implementação da funcionalidade ensinar protótipo. Na figura 22 podemos ver algumas telas do RoboEduc em sua versão 2.1.

a) Tela Inicial RoboEduc 2.1 b) Tela Ensinar Protótipo
Figura 22: Telas RoboEduc 2.1
Fonte: Adaptadas de Silva (2009)

Na quarta versão (BARROS, 2008) cita que a interface do RoboEduc é totalmente reformulada. A equipe de desenvolvimento opta por usar algo mais infantil, ficando claro a opção pelas séries inicias do Ensino Fundamental. Outra novidade é a implementação do Módulo Autoria. Esse módulo é destinado a professores ou monitores (alunos ou não) de robótica educacional. A figura 23 mostra a tela inicial do RoboEduc 3.0, tela de níveis de programação e tela de exemplo de programa.

a)Tela inicial RoboEduc 3.0 b) Tela níveis de programação c)Programa Completo
Figura 23: Telas RoboEduc 3.0
Fonte: Adaptadas de Silva (2009)

Em 2018 a *RoboEduc* lançou um novo produto, denominado *RoboEduc Inbox*, o RoboEduc Inbox é o primeiro sistema de EAD (Educação a Distância) do Brasil a trabalhar com Robótica Educacional. Fornecemos tanto os componentes físicos,

quanto os Objetos Virtuais de Aprendizagem (vídeos aulas, games e muito mais) (ROBOEDUC, 2019).

Foram desenvolvidos 4 caixas temáticas (figura 24), todas as caixas atreladas a ideia de cidades inteligentes. Cada box contém videoaulas e programas de computador abrangendo 4 Oficinas. Cada oficina possui uma atividade prática (hands-on) e tem duração de até 50min. Além das 4 oficinas práticas, é lançado um desafio para o aluno resolver e postar sua solução nas redes sociais.

Juntamente com as videoaulas, o box acompanha componentes eletrônicos como microprocessadores, leds, placas de prototipagem, sensores e motores, tudo o que o aluno precisa para começar a aprender robótica e programação. Para cada box, são enviadas peças prototipadas em impressoras 3D para a construção das Oficinas e Desafios. Cada box acompanha também material para auxiliar na contextualização dos problemas trabalhados, como mapas, peças acessórias, informativos, etc.

Figura 24: RoboEduc Inbox
Fonte: ROBOEDUC (2019)

3.3.9 Robokit

O Robokit (Figura 25) é desenvolvido através de parceria do curso de Licenciatura em Computação e da empresa IMPLY - Tecnologia Eletrônica, consiste em um Kit de Robótica Educacional que possui um módulo controlador independente, ou seja, funciona sem computador, é composto por quatro LED's, dois motores de corrente contínua e um motor de passo. O kit permite a programação de sons, a repetição de comandos e a elaboração de procedimentos. Através de programação, no teclado do kit, é possível controlar os motores e os LED's (ROBOKIT, 2019).

O kit é ideal para o desenvolvimento de programação e robótica, animando protótipos desenvolvidos pelos estudantes. A programação do equipamento em questão pode ser usando o LogoWriter, Megalogo, Basic, entre outras.

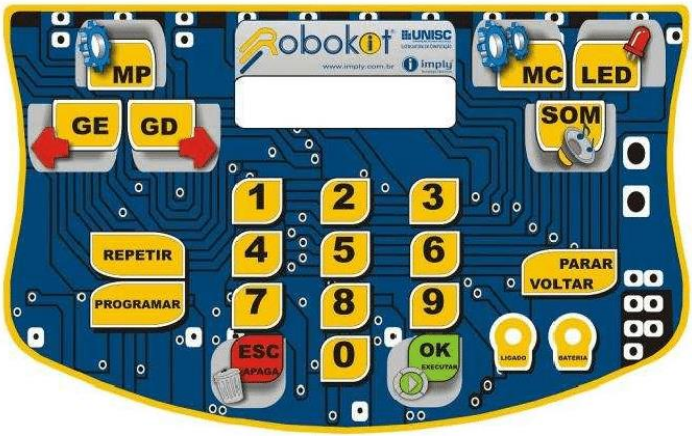


Figura 25: Robokit
Fonte: Robokit (2019)

3.4 Ambientes de programação utilizados na Robótica Educacional

Existem inúmeras ferramentas livres e comerciais que são utilizadas na programação de robôs. No entanto essas linguagens são recomendadas para usuários com conhecimentos básicos de programação e em robótica, dificultando a aprendizagem de crianças nesse meio tecnológico. Por exemplo as linguagens de programa Lego Mindstorms (LEGO, 2019), a linguagem BrickOS (BRICKOS, 2019), e o NQC (NQC, 2019), eles são baseados na linguagem de programação C, fazendo

com que sua compreensão não seja tão simples, principalmente por crianças que não possuem acesso a recursos tecnológicos.

O que apresenta um processo mais simplificado para aprendizagem é o Robolab, software vendido juntamente com os kits LEGO MINDSTORMS utilizando a linguagem de programação gráfica (visual) *Labview.*

- **Everest**

O Everest é um software que funciona como uma oficina de criação, fornecendo as ferramentas necessárias para o desenvolvimento. Ele se destaca por possibilitar o controle através de ações específicas na interface *SuperRobby* e *Cyberbox.* Com ele, é possível criar aplicações sem necessitar de conhecimentos aprofundados de programação, inserindo objetos como sons, imagens, vídeo, textos, animações, bancos de dados, etc. É uma ferramenta que pode ser utilizada em escolas ou em centros de treinamento. Por ser um programa aberto, possibilita a liberdade de criação e também de aplicação. Com ele, os alunos podem apresentar trabalhos, elaborar material de consulta para a escola, parti-lhar pesquisas entre escolas, elaborando projetos em parceria (COMPLEX, 2019).

- **Robolab**

O Robolab foi desenvolvido pela *Tufts CEEO University*, é uma linguagem de programação visual construída em *labview.* O ROBOLAB foi a linguagem de programação dominante para a robótica educacional LEGO antes do lançamento do NXT-Software (NXT-G), e serviu de inspiração não apenas para o Software NXT, mas também para o LabVIEW para LEGO MINDSTORMS. Ele oferece suporte tanto para o RCX como para o NXT (LEGO ENGINEERING, 2019).

O *Robolab* é uma versão simplificada do *LabView*, com o objetivo de tornar mais atraente a programação, com menos opções disponíveis e com uma interface para o usuário apropriada para crianças. Baseia-se em figuras, sem linhas de texto escrito, apoiando-se em uma seqüência lógica de im- agens e grava os programas sobre temas relativos aos conjuntos de construções com Lego ou categorias educacionais como matemática (EDACOM, 2019). Podemos ver um exemplo de sua tela inicial na figura 26.

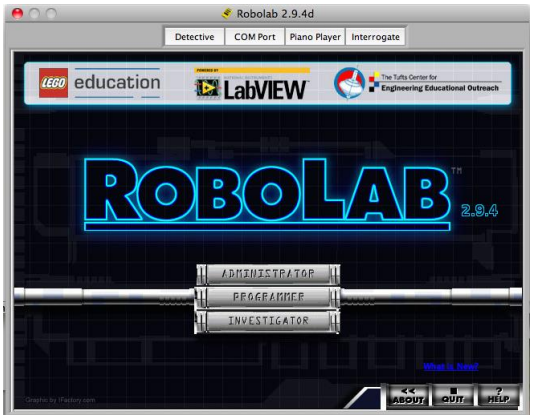


Figura 26: Tela de Início Robolab
Fonte: Lego Engineering (2019)

- **LOGO**

A linguagem LOGO desenvolvida por Seymour Papert em meados da década de 60 no MIT, passou por diversas fases. Desde sua criação, foram desenvolvidas diversas versões do Logo por diferentes empresas. Também, possibilita a sua utilização como *software de autoria*, ou seja, permite a criação de outros *softwares* a partir dele. Algumas versões de programas da família Logo: *LogoWriter*, *SuperLogo* e *StarLogo*. O *Micromundos* e o *Imagine* são versões da linguagem Logo orientados a objetos (CONSULT, 2019).

Segundo Zacharias (ZACHARIAS, 2003), as características do Logo que o tornam uma linguagem de programação de fácil assimilação são:

- Exploração de atividades espaciais, permitindo assim que a criança tenha contato imediato com o computador;
- linguagem procedimental (procedural), que possibilita a criação de novos comando ou procedimentos em Logo;
- Divide-se, basicamente, nos comandos primitivos que são próprios da linguagem e em nomes ou rótulos de procedimentos, escritos pelo usuário;
- Possui comandos para manipular palavras e listas, com os quais é possível programar a tartaruga, criar histórias, animações, jogos, etc.

- **W-EDUC**

O W-Educ[3] é um ambiente de desenvolvimento web multiplataforma configurável para aplicações em robótica educacional desenvolvido pela NatalNet, (Sá, 2016) cita que o W-Educ é uma nova ferramenta *web*, dinâmica, aberta e gratuita para a programação de robôs que contém um conjunto único de funcionalidades para dar suporte ao ensino de programação e algoritmos para robótica.

A ferramenta torna possível programar e enviar o programa a diferentes tipos de dispositivos robóticos utilizando uma linguagem universal, a linguagem R-Educ, através de comandos gráficos ou textuais, programar na linguagem base do controlador, permitir interação entre aluno e professor. Tudo isso em uma ferramenta web aberta, dinâmica, e gratuita, possibilitando que a mesma seja utilizada em escolas ou pelo público em geral (Sá, 2016).

A figura 27 mostra o ambiente com algumas linguagens já cadastradas disponíveis para programação, e na figura 28 apresenta o ambiente de programação do W-Educ. Nele, é possível selecionar se deseja programar em R-Educ ou diretamente na linguagem alvo, nesse caso NXC, por exemplo. É possível salvar e abrir programas salvos no banco de dados, compilar o programa, visualizar dicionário de funções, enviar código ao robô, baixar o código escrito na extensão atual, converter para a linguagem alvo e solicitar correção do código.

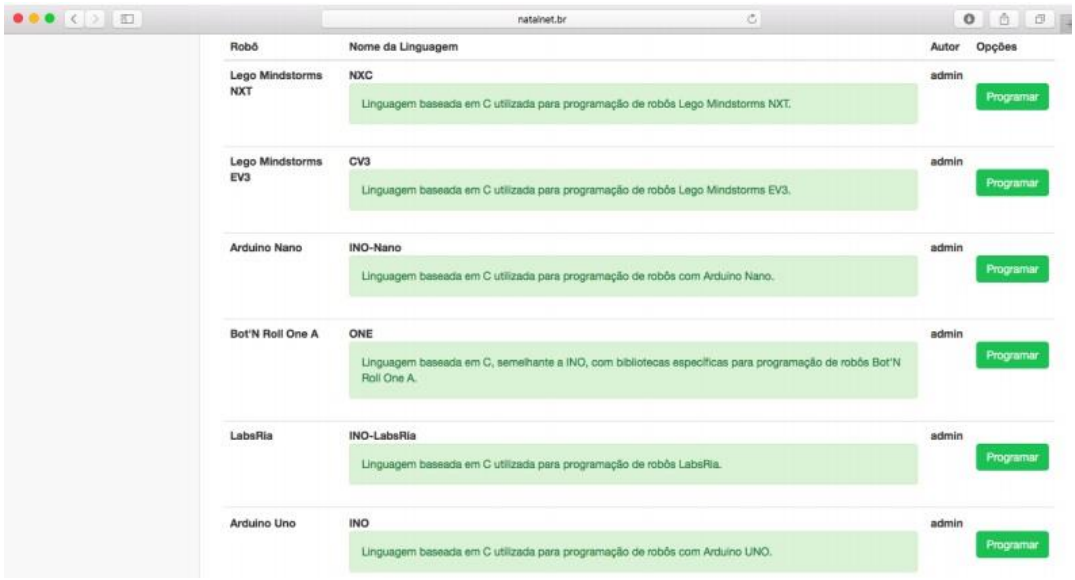


Figura 27: Linguagens cadastradas no ambiente Weduc
Fonte Sá (2016)

[3] http://www.natalnet.br/weduc

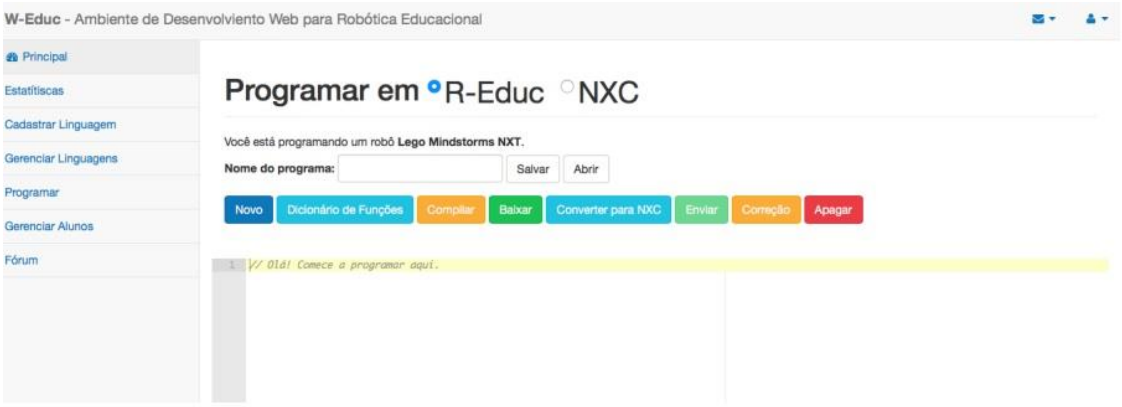


Figura 28: Ambiente de desenvolvimento weduc
Fonte: Sá (2016)

- **Megalogo**

O MegaLogo é um programa de computador utilizado para automação e controle de dispositivos robóticos. Foi desenvolvido em 1994 por A. Blaho Tomacsányi e distribuído na Inglaterra, em Portugal e no Brasil. No Brasil o programa obteve mais de duzentos mil usuários. Embora ainda seja procurado por professores de educação tecnológica, não é mais comercializado desde 1 de abril de 2003, tendo sido substituído pelo Imagine, do mesmo autor, mas com mais recursos (FABRICIO *et al*, 2012).

O autor cita que MegaLogo é um poderoso ambiente de programação e aprendizagem multimídia, que permite realizar atividades centradas na aprendizagem, na cooperação e na descoberta. É baseado na linguagem de programação Logo e desenvolvido para o ambiente Microsoft Windows, suportando todos os recursos desse ambiente operacional: som, vídeo, imagens, cores, fontes e impressão, entre outros.

A figura 29 mostra a interface do ambiente de desenvolvimento do Megalogo, vários trabalhos de robótica educacional foram desenvolvidos através do Megalogo, todas as partes que tem animação são formadas por tartarugas programadas para ter várias fases, permitindo que o usuário crie animações com facilidade.

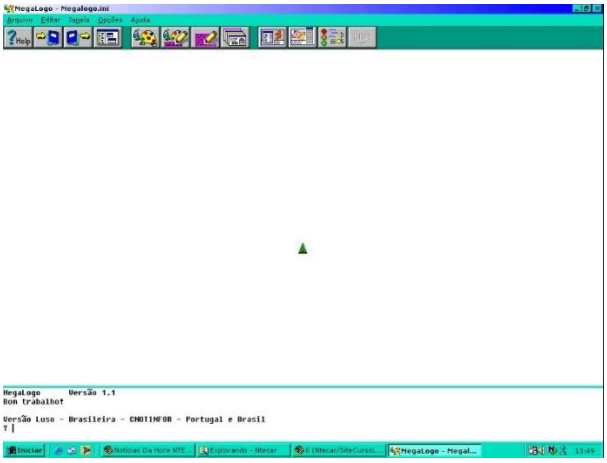


Figura 29: Ambiente de desenvolvimento Megalogo
Fonte: FABRICIO *et al* (2012)

4 USO DO PADRÃO DUBLIN CORE PARA OBJETOS DE ROBÓTICA EDUCACIONAL

Este capítulo apresenta a análise de alguns trabalhos que serviram como base para a realização desta monografia, que tem como proposta a criação de um repositório de objetos de robótica educacional baseado no padrão de descrição de metadados Dublin Core. Portanto foi necessário a pesquisa e o aprofundamento em projetos que utilizam esse tipo de padrão para o desenvolvimento e modelagem de suas aplicações, o foco principal foi explorar a metodologia utilizada para a descrição dos dados e identificar os principais aspectos e dificuldades encontradas no padrão para o desenvolvimento dos projetos propostos. Após isso é mostrada a proposta sugerida para utilização, adaptação e extensão do Dublin Core para objetos de aprendizagem que tratam sobre robótica educacional.

4.1 Trabalhos Relacionados

O trabalho de Roseto & Nogueira (2002) aplica os elementos metadados Dublin Core para a descrição de dados bibliográficos on-line da biblioteca digital de teses da USP, nele é exposto que dentre as atividades desenvolvidas para a implementação da biblioteca digital, constou a definição do conjunto de elementos metadados necessários à descrição dos recursos bibliográficos a serem introduzidos na base de dados. Essas informações são digitadas pelo próprio autor da tese, durante o processo de submissão do documento em formato eletrônico ao programa para armazenagem de dados.

Também é citado pelos autores que foi feita uma análise de alguns formatos metadados, e acabou optando-se pelo Dublin Core, que é um padrão adotado mundialmente em vários tipos de projetos para gestão de recursos digitais. Além disso, trata-se de um modelo que permite a inclusão de elementos adicionais para atender às particularidades da instituição, sendo ainda um formato padrão adotado para efetuar a interoperabilidade entre outros formatos, na figura 30 podemos observar parte do mapeamento que foi feito para a descrição dos dados desejados.

ELEMENTOS DUBLIN CORE (DC)	Status (M – O – R)[1]	MARC (DEDALUS)	Origem/ Preenchimento (A – B - F – T – Aut)[2]	Conteúdo / Notas
DC.Title.Pt	M	245 $a	A	Título do documento em Português
DC.Title.X-Subtitle.Pt (+)	M	245 $b	A	Subtítulo do documento em Português
DC.Title.En (+)	O	245 $ a	A	Título do documento em Português e Inglês
DC.Title.X-Subtitle.En (+)	O	245 $b	A	Subtítulo do documento em Português e Inglês
DC.Title.X-Notes	O	-----		Notas sobre o Título
DC.Creator.PersonalName	M	100 $a	F	Autor Principal do Documento
DC.Creator.X-PID	M	946 $b 946 $c	F	Identificação na USP (CodPess do Autor)
DC.Creator.X-No.USP (+)	M		F	Identificação na USP (No. USP do Aluno de Pós-Graduação quando houver)
DC.Creator.Address	O	EXT 01	B	Email do autor
DC.Creator.X-Institution	M	------	Default	Nome da Instituição do Autor (Default Universidade de São Paulo)
DC.Creator.X-Major	M	590 $d $e	F	Área de Concentração (programa de pós) (nome e código)
DC.Creator.X-College	M	946 $e $f	F	Unidade USP sigla e nome da Unidade
DC.Creator.X-Dept	O	599 $a $b	F	Departamento da Unidade sigla e nome da Unidade
DC.Subject.X-Keyword.Pt (+)	M	952$a $f	A A	Palavras-chave em português (Colocar o valor default "Proveniente da Base Teses Digitais da USP" no DEDALUS)
DC.Subject.X-Keyword.En (+)	M	952$a $f	A A	Palavras-chave em Inglês Colocar o valor default "Proveniente da Base Teses Digitais da USP" no DEDALUS)

Figura 30: Mapeamento de Elementos Metadados Dublin Core para Descrição e Gerenciamento da Biblioteca Digital de Teses da USP.
Fonte: Roseto & Nogueira (2002)

Em Gonçalves *et al.* (2013) foi feita uma adequação do Dublin Core para o caso da Biblioteca Digital da Assembleia Legislativa do Estado de Minas Gerais. Na implementação da biblioteca foi utilizado o software DSpace, que é um software de código aberto que tem como principal funcionalidade a implementação e estruturação de acervos digitais, a descrição bibliográfica ficou condicionada ao Código de Catalogação Anglo Americano 2 (AACR2).

O padrão de metadados escolhido foi o Dublin Core, pois segundo os autores, ele permite a descrição padronizada de publicações digitais. Contudo, para que as regras do AACR2 pudessem ser compatibilizadas com o DCMI, foram realizadas algumas adaptações no formato. Na figura 31 pode-se ver a tabela onde foi feita a adaptação do AACR2 ao DCMI.

Áreas das ISBDs	Campos do AACR2	Campos DCMI
Área de título e indicação de responsabilidade	Título principal Título equivalente Título alternativo Título uniforme Indicação de responsabilidade	dc.title dc.title.equivalent dc.title.translate dc.title.alternative dc.title.uniform dc.contributor.advisor dc.contributor.author dc.contributor.compiler dc.contributor.coordinator dc.contributor.editor dc.contributor.funder dc.contributor.illustrator dc.contributor.machine dc.contributor.organization dc.contributor.organizator dc.contributor.other dc.contributor.reviewer dc.contributor.sponsor dc.contributor.translater
Área de outros detalhes específicos do material	Tipo	dc.type
Área de edição	Edição dc.e	dc.edition
Área de publicação e distribuição	Local Editora Data	dc.publisher.place dc.publisher dc.date.issued
Área de descrição física	Extensão Dimensões Material adicional	Não usado
Área de série	Série	dc.relation.ispartofseries dc.relation.isformatof
Área de notas	Resumo Sumário Notas de Conteúdo Resumo em língua estrangeira	dc.description.abstract dc.description.summary dc.description.tableofcontents dc.description.translated
Área de número normalizado	Número de ISBN Número de ISSN Outros números normalizados	dc.identifier.isbn dc.identifier.issn dc.identifier.other

Figura 31: Adaptação do AACR2 ao DCMI
Fonte: Gonçalves *et al.* (2013)

No trabalho proposto por Pires (2012), foi escolhido o padrão Dublin Core para a Descrição de Obras Raras na Web: A Coleção da Biblioteca Brasiliana Digital, o autor cita que depois de definido o esquema de metadados (Dublin Core qualificado) para a descrição bibliográfica, houve a necessidade de adaptá-lo às necessidades da coleção, definindo quais campos e qualificadores seriam utilizados e como estes atenderiam às necessidades da coleção e as diferentes tipologias documentais, assim como atender as orientações do repositório (DSpace), quanto ao uso dos campos.

Para garantir a uniformidade e a padronização da descrição dos metadados foram adotados normas e padrões utilizados e consolidados pelas bibliotecas, o documento norteador utilizado foi o AACR2. Após a importação dos metadados para o DSpace os metadados são conferidos e descritos conforme o padrão de descrição adotado pela Biblioteca Brasiliana Guita e José Mindlin para recursos na web. Na figura 32 podemos ver uma parte dos metadados descritos na adaptação.

Elemento Dublin Core	Qualificador	Etiqueta / legenda para o site	Notas
DC.Contributor	author	Autor	Entidade responsável pelo conteúdo do item
DC.Contributor	other	Colaborador	Pessoas ou entidades que contribuíram para o conteúdo do item. Ex.: tradutores, ilustradores, gravadores, prefaciadores, impressor, etc
DC.Title	none	Título	Nome dado ao documento. Forma pelo qual o documento é formalmente conhecido
DC.Title	alternative	Título alternativo	Forma alternativa ao título. Neste campo incluem-se as traduções ou abreviaturas do título
DC.Date	none	Data de publicação	Data de publicação do item
DC.Description	none	Descrição/Notas	Descrição de particularidades do item
DC.Description	isversionof	Versão	Referencia a edição
DC.Description	provenance	Procedência	Descrição da custódia do item. Utilizar este campo para descrever a que biblioteca pertence o item
DC.Description	abstract	Resumo da obra	Sinopse do conteúdo do recurso
DC.Description	tableofcontents	Conteúdo	Descrição do conteúdo do item: sumário, lista de ilustrações
DC.Publisher	none	Local de Publicação	Entidade responsável por tornar o documento acessível em sua forma. Incluem pessoas, organizações ou serviços. (Colocar Local de publicação e o nome da editora)
DC.Format	medium	Descrição física	Descrição dos aspectos físicos do livro (n. de páginas, volume, ilustrações etc)
DC.Language	none	Idioma	Utilizar este campo para introduzir informação sobre a língua diferente do formato ISO
DC.Relation	none	Relação	Relação do item. (Nestes campos de metadados devem ser atribuídos as relações de uma obra, deve-se utilizar para se estabelecer os *links* no acervo ou fora dele)
DC.Relation	ispartofseries	Série	Nome e número da série/coleção
DC.Relation	ispartof	Parte de	Utilizar quando o item for parte de uma obra (Ex. capítulos de livros, para imagens, mapas que fazem parte de um item que já faz parte do acervo, inserir o handle)

Figura 32: Metadados utilizados pela Brasiliana Digital na Plataforma Corisco
Fonte: Pires (2012)

Apesar destes trabalhos estarem voltados para para uso e adaptação do padrão Dublin Core, nenhum deles trata sobre Robótica Educacional. Assim como repositórios de objetos de aprendizagem pesquisados não tinham como foco a Robótica Educacional. Dessa forma fez-se necessário elaborar uma proposta de adaptação do padrão Dublin Core para representar objetos de robótica, assim como implementar um repositório próprio para esses objetos.

4.2 Adaptação da proposta ao padrão Dublin Core para Robótica Educacional

No capítulo 2 foram citadas as principais características do padrão DC, sua importância e simplicidade, e o motivo escolhido para sua aplicação. Foi feita uma análise de alguns formatos metadados, e acabou optando-se pelo Dublin Core, pois ele é um padrão adotado mundialmente em vários tipos de projetos para gestão de recursos digitais. Além disso, trata-se de um modelo que permite a inclusão de elementos adicionais para atender diversas particularidades, sendo ainda um formato padrão adotado para efetuar a interoperabilidade entre outros formatos.

Para realizar a adaptação dos metadados dos objetos de RE para o padrão DC foram usados como base os trabalhos relacionados no que são expostos no capítulo 4 desta monografia, como também foi feita uma pesquisa bibliográfica a respeito dos Kits de Hardware utilizados na RE, e os softwares e linguagens mais procurados, tipos de montagem, programação ou outra tarefa específica.

O apêndice A traz uma tabela onde é feito o mapeamento de cada campo disponível para a descrição de metadados do Dublin Core para os Objetos de Robótica Educacional, vale ressaltar que no padrão DC não há obrigatoriedade de campos e que eles podem aparecer duplicados.

A seguir temos um breve resumo explicativo de como foi desenvolvido cada campo da tabela mostrando suas principais características, atendendo aspectos de especificação dos metadados de acordo com o que foi visto na literatura.

1- Elemento **Título** DC: Este elemento representa o nome que será dado ao objeto ou recurso, sendo composto por 2 campos.

- *dc.title:* Campo Original que serve para descrever o título do objeto ou do projeto, (ex: Carro de controle remoto, robô aranha, etc).

- *dc.subtitle:* Elemento para caracterizar um complemento ao título.

2- Elemento **Criador** DC: Elemento utilizado para descrever o autor do objeto, podendo ser uma pessoa, uma organização ou um serviço, sendo composto por 5 campos.

- *dc.creator:* Campo Original onde o autor será representado.
- *dc.creator.authority:* Responsável por prescrever os objetos de aprendizagem e os resultados.
- *dc.creator.search_information:* Responsável por buscar por recursos de acordo com os metadados fornecidos.
- *dc.creator.apprentice:* Elemento que caracteriza aquele que irá utilizar o Objeto.
- *dc.creator.organizer:* Criador responsável por projetar atividades de aprendizagem e revisar as licenças e direitos autorais de uso.

3- Elemento **Assusto** DC: Elemento responsável por expressar o assunto com tópicos, palavras chave, ou códigos que representam o conteúdo informativo do objeto, sendo composto apenas por um campo.

- *dc.subject.theme:* Descrever o tema do objeto ou um conteúdo informativo e sobre do que ele se trata.

4- Elemento **Descrição** DC: Responsável por representar uma breve descrição do conteúdo do objeto podendo ser composto por textos, resumos ou tabelas, o elemento é composto por 7 campos.

- *dc.description.keyword:* Palavra-chave relacionada ao objeto ex: Lego, Arduino, sensores.
- *dc.description.general:* Breve descrição geral sobre de que se trata o objeto.
- *dc.description.building:* Descrição de como o objeto é construído.

- *dc.description.type_programming:* Tipo de programação nativa do objeto eu está sendo representado, ex: Programação em texto, blocos, etc.

- *dc.description.program_objective:* Descrever se o objetivo do objetivo é construir um movimento específico ou alguma funcionalidade.

- *dc.description.educational_objetictive:* Descrição do objetivo educacional do Objeto, ex: Interatividade, Autonomia, Cooperação, Cognição, e Afetividade.

- *dc.description.requirements:* Requisitos necessários para construção do objeto, por ex: Informar aos usuários se será necessário algum conhecimento prévio específico, ou a utilização de alguma ferramenta ou plataforma específica.

É importante que os dados descritivos estejam bem definidos para atender aos requisitos técnicos dos OAs, com ênfase no requisito de disponibilidade, pois quanto melhor descrito for o objeto mais disponível ele estará para os usuários.

5- Elemento **Editor** DC: O editor é representado por uma entidade responsável por deixar o objeto acessível para outros usuários, e é composto por apenas 1 campo.

- *dc.publisher.editor:* Responsável pela edição do objeto, pode ser uma pessoa, organização ou serviço.

6- Elemento **Contribuinte** DC: Representa qualquer pessoa, organização ou serviço que é responsável por colaborar com qualquer conteúdo durante o desenvolvimento do objeto, sendo composto por 2 campos.

- *dc.contributor:* Campo original que representa os membros responsáveis pelo desenvolvimento.

- *dc.contributor.partner:* Instituição parceira na qual o objeto foi desenvolvido.

7- Elemento **Data** DC: O elemento Data é responsável por informar a data que determinados eventos acontecem com o objeto, sendo composta por 2 campos.

- *dc.date.creation:* Data de criação do objeto
- *dc.date.availability:* Data de disponibilização do objeto no repositório.

8- Elemento **Tipo** DC: O elemento tipo se caracteriza por 10 campos responsáveis por identificar detalhes, descrevendo as categorias gerais e funções específicas de cada objeto.

- *dc.type.kit:* Tipo de kit que está sendo utilizado, por ex: Lego, Arduino, Vex, Modelix, roboeduc, etc.
- *dc.type.version:* Tipo de versão, ex: 1.0, 2.0, 3.0.
- *dc.type.model:* Modelo do objeto, ex: NXT programs, Lego Boost, EV3, Arduino Mega, nano, etc.
- *dc.type.language:* Tipo de linguagem utilizada na construção do programa, por ex: Java, C++, C, Programação em blocos, linguagem visual, etc.
- *dc.type.activity:* O tipo de atividade que pode ser caracterizado como montagem, construção, programação ou outra tarefa.
- *dc.type.age*: Idade recomendada para que o objeto seja utilizado ou compreendido. Ex: 7 anos, 8 ou 10.
- *dc.type.level:* Tipo de nível de ensino onde exemplifica se o objeto é recomendado para algum nível específico, como ensino médio, fundamental, superior, ou infantil.
- *dc.type.programming_environment*: Qual ambiente de programação é utilizado pelo objeto, ex: IDE arduíno, ambiente NXT, Robolab, everest, etc.
- *dc.type.user:* Qual usuário o objeto é destinado, ex: Professor, aluno ou usuário comum.
- *dc.type.software:* Tipo de software utilizado para realização dos movimentos, ex: NXT, robolab, IDE.

9- Elemento **Formato** DC: Elemento responsável por descrever qual o formato que o objeto está sendo disponibilizado para os usuários, sendo mantido o campo original.

- *dc.format:* Formato que o objeto é disponibilizado, ex: Audio, vídeo, página web, imagem, aplicação, etc.

10- Elemento **Identificação** DC: Representar uma referência que é responsável por identificar o objeto, sendo adicionado um campo extra ao campo original.

- *dc.identifier.area:* Identificar a área de ensino que o objeto poderá ser utilizado, ex: Química, Matemática, Física, etc.

11- Elemento **Fonte** DC: A fonte pode representar uma referência ao objeto atual ou onde ele se encontra. O campo original foi modificando sendo adicionado um subcampo.

- *dc.source.location:* Onde o objeto está localizado, onde foi publicado e disponibilizado, podendo ser fornecido um link ou página.

12- Elemento **Idioma** DC: Este elemento mostra a linguagem em que o objeto foi construído e que será disponibilizado, sendo mantido o campo original.

- *dc.language:* Idioma que o objeto foi construído e disponibilizado.

13- Elemento **Relação** DC: Esse campo representa a descrição de algum tipo de relação do objeto atual com outro objeto, sendo mantido o campo original.

- dc.relation: Relação com outro objeto, a relação pode ser por ex: Tipo de versão, referência, formato do objeto, tipo de kit, linguagem, idade recomendada etc.

14- Elemento **Direitos** DC: Informar as informações de direitos de propriedade intelectual sobre o objeto, sendo mantido o campo original.

- dc.rights: Direitos e informações de propriedade do objeto.

5 RepositORE

Este capítulo tem como principal objetivo apresentar o desenvolvimento da aplicação, a forma que ela foi analisada e feita a modelagem dos seus diagramas para poder garantir um desenvolvimento preciso sem perder o foco das principais funcionalidades. São expostas as principais tecnologias que foram utilizadas durante o desenvolvimento e por fim os resultados obtidos.

O sistema RepositORE, Repositório de Objetos de aprendizagem para Robótica Educacional é uma ferramenta que realiza o cadastro e busca de objetos de aprendizagem voltados para robótica educacional. Ele facilita a busca por materiais voltados para o tema de robótica, pois na web existem diversos materiais que são disponibilizados de forma não convencional, dificultando a busca por objetos específicos.

Dessa forma o sistema visa melhorar essa busca e o cadastro desses objetos, e utilizando esse sistema o usuário poderá utilizar os serviços de forma livre, ou realizar o cadastro dos seus próprios objetos, podendo também sugerir edições para objetos já cadastrados, fornecendo mais informações que possam facilitar a busca.

No âmbito educacional é crescente a busca por novas tecnologias para compartilhamento de conteúdos de forma simples e didática. A partir dessa demanda, é observado um aumento na quantidade de materiais dos mais diversos tipos e temas que são disponibilizados na internet.

Nesse contexto temos, por exemplo, imagens, vídeos, cursos e jogos, cabe a cada indivíduo que busca por esse conhecimento escolher a melhor ferramenta que irá auxiliar no seu processo de aprendizado. Dessa forma o sistema descrito contribui de forma significativa para a realização de buscas por determinado material específico na área de Robótica Educacional, fazendo com que os resultados buscados sejam disponibilizados com uma maior precisão.

5.1 Modelagem do Sistema

O sistema foi projetado para ter dois tipos de papéis: o Usuário e o Administrador. O primeiro é relativo aos usuários do RepositORE que estão

interessados em buscar ou compartilhar informações sobre os OA's em Robótica. O segundo assume o papel de gestor e moderador do sistema. Na figura 33 pode-se observar o diagrama de casos de uso propostos para esse sistema de acordo com as funcionalidades que foram descritas a seguir:

- **Papel Usuário:**

- Realizar Cadastro no Sistema: Esta funcionalidade permite que o usuário crie uma conta no repositório para poder cadastrar os objetos. O usuário precisa fazer o seu cadastro informando nome, login e-mail e a senha desejada, ou poderá fazer o seu cadastro utilizando a conta do facebook ou do Google.

-Pesquisar Objeto: Esta funcionalidade é essencial para o sistema, pois ela permite que o usuário possa pesquisar os objetos que estão disponíveis no repositório. Após a pesquisa os objetos são retornados de acordo com pesquisa realizada pelo usuário.

-Ver Metadados: Esta funcionalidade permite que o usuário visualize os metadados do objeto que ele visitou. Para isso ele precisa pesquisar o objeto e visualizar metadados, essa é uma função imprescindível para o sistema pois os metadados irão facilitar a busca dos objetos na *web.*

-Cadastrar Objeto no Sistema: Esta função do sistema permite que o usuário faça o cadastro do objeto que ele deseja disponibilizar no repositório. Ele irá preencher os campos necessários para que o objeto seja disponibilizado no sistema.Para realizar o cadastro do objeto o usuário deve realizar o seu cadastro no sistema, ou se já tiver cadastro basta estar logado.

-Editar Objeto Cadastrado: Permite que o usuário possa editar o objeto que ele mesmo cadastrou, caso alguma informação inserida esteja errada ou ele queira modificar alguma informação,ou até mesmo excluir do sistema o objeto cadastrado. Para que essa modificação seja realizada o usuário precisa ter algum objeto cadastrado no sistema para poder editá-lo.

-Deletar Objeto Cadastrado: Esta funcionalidade permite que o usuário possa deletar do sistema o objeto que ele cadastrou. Dessa forma o objeto é deletado do repositório e não ficará mais disponível para os usuários.

-Sugerir edição para objeto cadastrado: Esta funcionalidade propõe que o usuário possa sugerir a edição em algum objeto que ele não cadastrou, ou seja, o objeto foi cadastrado por outra pessoa, e ele irá sugerir uma edição. Ao sugerir a edição, o usuário criador do objeto cadastrado irá verificar a edição sugerida e verificar se irá aceitar ou não a modificação. O usuário administrador também pode verificar as sugestões de edição.

-Avaliar Objeto: Esta função permite que o usuário possa fazer uma avaliação do objeto que ele visitou ou, informar se a informação adquirida foi útil e se atendeu as suas expectativas. Ele deverá avaliar o objeto que ele visitou, deixando sua opinião em forma de comentário ou fazendo uma avaliação de 0 a 10.

- **Papel Administrador**

-O administrador do sistema poderá tanto realizar todas as funcionalidades dos usuários comuns, como também analisar as edições para objetos que forem cadastrados no sistema.

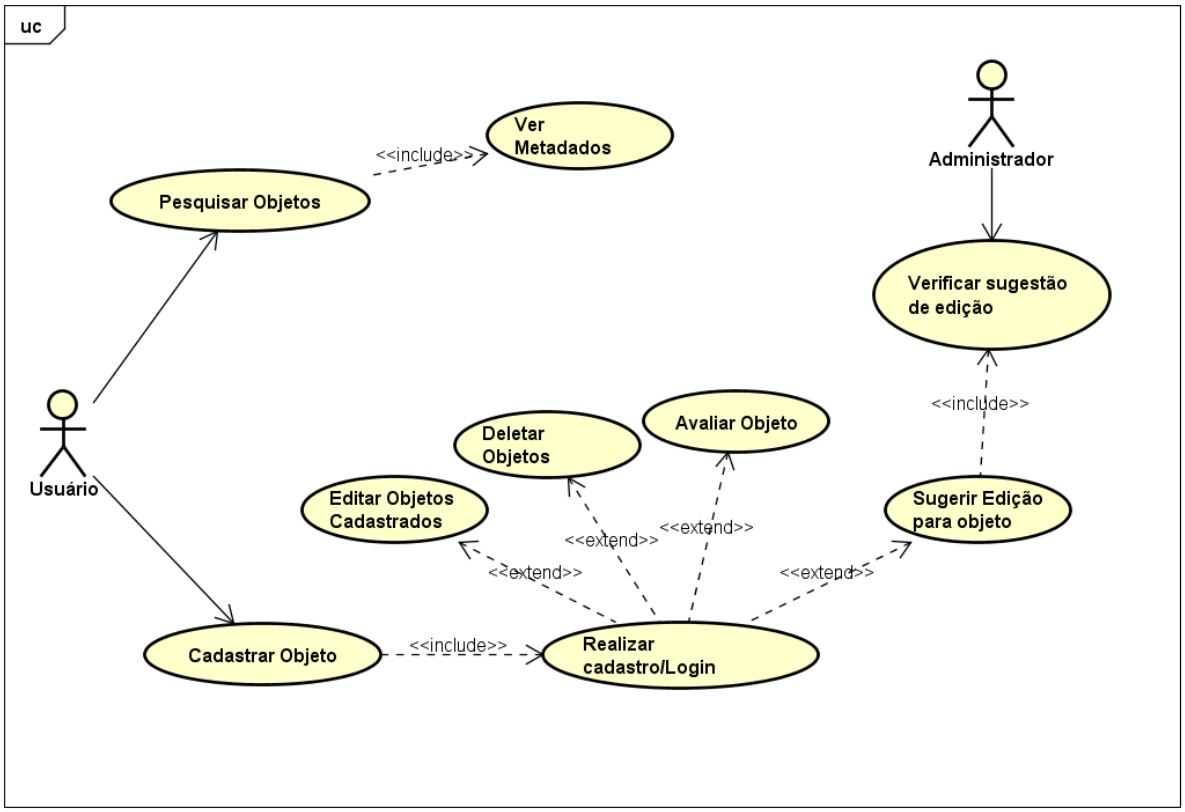


Figura 33: Diagrama de Caso de Uso do sistema
Fonte: Autoria Própria

A arquitetura do sistema proposto descrita na figura 34, apresenta como é feita a comunicação entre o usuário e a interface visual do sistema. Ao entrar no sistema o usuário entra em contato com a interface *front-end*, onde são feitas as requisições necessários e enviadas ao servidor por meio da internet, o servidor é responsável pelo processamento dos dados do sistema e pelo retorno das informações que são requisitadas pelo usuário, ele retorna essas informações a partir do contato com o banco de dados que tem a função de armazenar as informações gerais do sistema e dos usuários, assim como devolver as informações armazenadas ao servidor.

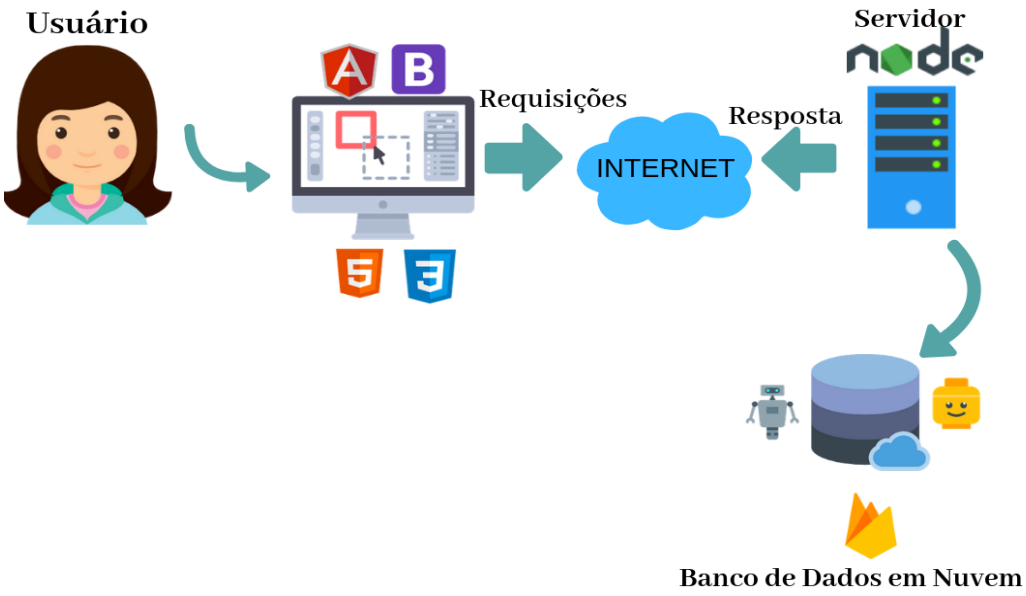


Figura 34 : Arquitetura da aplicação
Fonte: Autoria Própria

5.2 Tecnologias utilizadas

Nesta seção é feita a descrição das principais características das tecnologias utilizadas para implementação do RepositORE.

- **Firebase**

O Firebase é uma plataforma de desenvolvimento de aplicativos para dispositivos móveis e *Web* que fornece aos desenvolvedores uma gama de ferramentas e serviços para ajudá-los a desenvolver aplicativos de alta qualidade, aumentar sua base de usuários e obter mais lucros. Ele surgiu a partir da crição de uma startup chamada Envolve. Em abril de 2012, o Firebase foi criado como uma empresa que oferecia o *Backend-as-a-Service* com funcionalidade em tempo real. Depois de ser adquirida pelo Google em 2014, o Firebase evoluiu rapidamente para o gigante multifuncional de uma plataforma para dispositivos móveis e Web que é hoje (GEEKYANTS, 2017).

Ele fornece aos seus usuários as seguintes funcionalidades:

- Banco de dados em tempo real;
- Autenticação de usuários;
- Laboratório de Testes;
- Crashlytics;
- Funções de Nuvem;
- Firestore;
- Armazenamento na nuvem;
- Monitoramento de desempenho;
- Relatório de Falhas;
- Hospedagem.

O Firebase *Realtime Database* é um banco de dados NoSQL hospedado na nuvem que permite armazenar e sincronizar seus usuários em tempo real. O Realtime Database é um grande objeto JSON que os desenvolvedores podem gerenciar em tempo real. Com apenas uma única API, o banco de dados do Firebase fornece ao seu aplicativo o valor atual dos dados e qualquer atualização desses dados.

A sincronização em tempo real facilita o acesso dos usuários a seus dados em qualquer dispositivo, seja ele na Web ou em dispositivos móveis. O Realtime Database também ajuda seus usuários a colaborarem uns com os outros.

De acordo com as funcionalidades e características benéficas citadas, foi escolhido o Firebase como a ferramenta de banco de dados deste trabalho devido ser uma tecnologia atual, apresenta inúmeros benefícios para a aplicação como foi citado anteriormente, e possui uma simples sincronização com os frameworks utilizados nesta aplicação.

- **Angular JS**

A Angular, apoiada pelo Google, é uma plataforma de engenharia de software de código aberto usada para construir interfaces de usuário (front-end). Sua história começou a 2009, quando Misko Hevery e Adam Abrons, engenheiros do Google, desenvolveram o framework atualmente conhecido como AngularJS e o lançaram oficialmente em 2010 (ALTEXSOFT, 2018).

O AngularJS simplifica o desenvolvimento de aplicativos, apresentando um nível mais alto de abstração para o desenvolvedor. Como qualquer abstração, ela tem um custo de flexibilidade. Em outras palavras, nem todo aplicativo é adequado para o AngularJS. O AngularJS foi construído com o aplicativo CRUD *(Create, Read, Update, Delete)* em mente. Felizmente, os aplicativos CRUD representam a maioria dos aplicativos da web (ANGULARJS, 2019).

Atualmente o Angular encontra-se disponível na sua versão 7, que foi utilizada neste trabalho, e é escrito utilizando a linguagem *typescript,* que é basicamente um superconjunto para *JavaScript*.

Segundo os conceitos descritos pela ALTEXSOFT (2018) ele é totalmente compilado para *JavaScript*, mas ajuda a identificar e eliminar erros comuns ao digitar o código. Embora pequenos projetos JavaScript não exijam tal aprimoramento, os aplicativos de escala corporativa desafiam os desenvolvedores a tornar seu código mais limpo e a verificar sua qualidade com mais frequência. De acordo com a pesquisa 2018 StackOverflow, 36,9% dos engenheiros de software agora aplicam o AngularJS e para criar interfaces de usuário.

Podemos citar alguns benefícios de utilizar o Angular no desenvolvimento uma aplicação:

- Arquitetura baseada em componentes que fornece uma qualidade de código mais alta;
- Reutilização;
- Legibilidade;
- Manutenção;
- Ligação de dados bidirecional;
- Injeção de Dependência;
- Comunidade ativa.

O Angular assim como diversos frameworks possui seus concorrentes, de acordo com um estudo feito por Peyrott (2016), ele mostra na figura 35 o desempenho do angular em relação aos seus concorrentes.

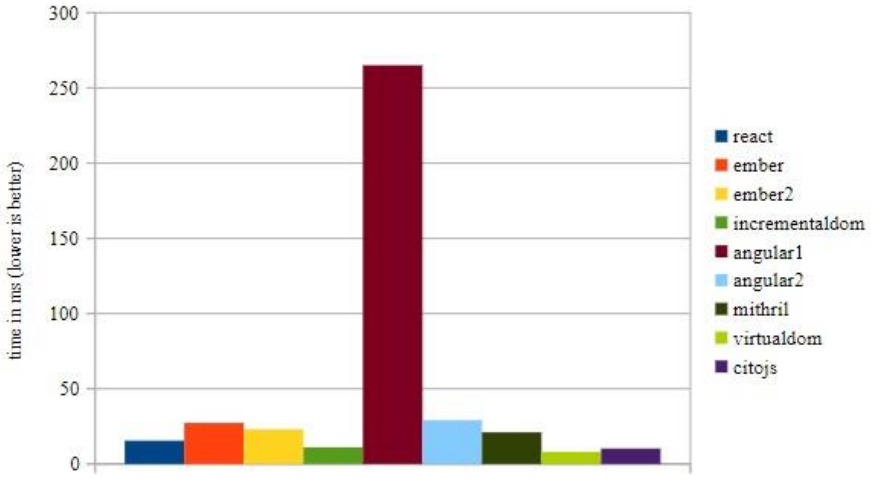


Figura 35: Desempenho do angular relacionado aos seus concorrentes
Fonte: Peyrott (2016)

O Angular trabalha com a arquitetura MVC (*Model View Controller*), possuindo uma arquitetura pré estabelecida (figura 36).

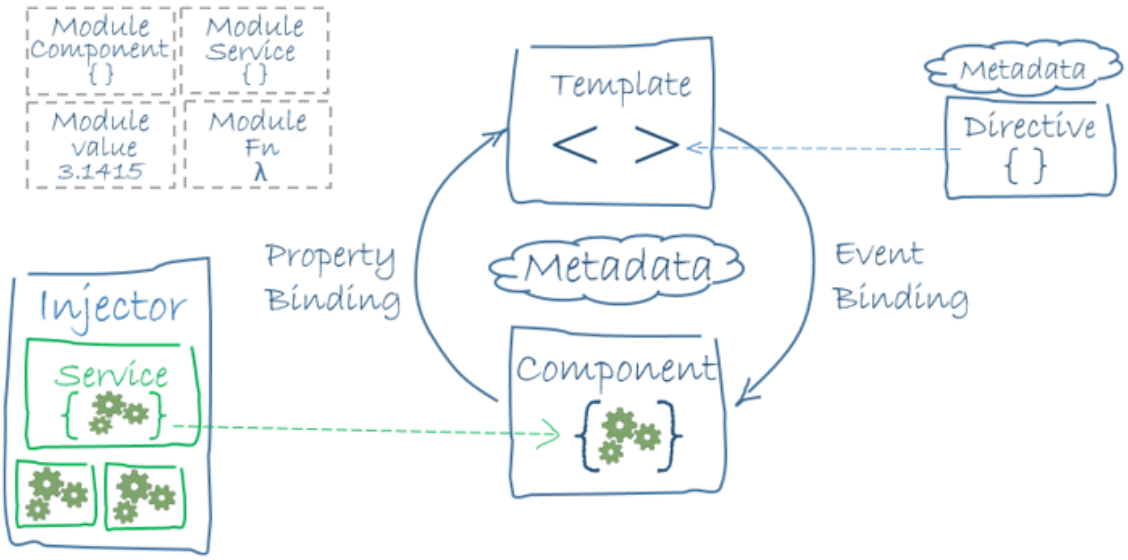


Figura 36: Arquitetura do Angular
Fonte: Pereira (2018)

Neste modelo, a camada de *Component* e *Template* têm a responsabilidade de definir a *view*. Um *decorator* em *Component* define o *metadata,* onde associa-se qual *Template* faz parte do *Component* e outras configurações. Também há as diretivas, na camada *Directive,* que tem a função de modificar a *view*, baseado nos dados e lógica utilizados.

- **Bootstrap**

Originalmente criado por um designer e um desenvolvedor no Twitter, o Bootstrap se tornou um dos *frameworks front-end* mais populares e projetos de código aberto no mundo. O Bootstrap foi criado no Twitter em meados de 2010 por Mark Otto e Jacob Thornton. Antes de ser uma estrutura de código aberto, o Bootstrap era conhecido como *Twitter Blueprint*. Após alguns meses de desenvolvimento, o Twitter realizou sua primeira *Hack Week* e o projeto explodiu à medida que os desenvolvedores de todos os níveis evoluíram sem nenhuma orientação externa. Ele serviu como guia de estilo para desenvolvimento de ferramentas internas na empresa por mais de um ano antes de seu lançamento público, e continua a fazê-lo hoje (BOOTSTRAP, 2019).

O Bootstrap é uma estrutura de front-end gratuita e de código aberto para o desenvolvimento de websites e aplicativos da web . Ele contém modelos de design baseados em HTML e CSS para tipografia , formulários , botões, navegação e outros

componentes de interface, além de extensões JavaScript opcionais . Ao contrário de muitos frameworks web, ele se preocupa apenas com o desenvolvimento front-end (BOOTSTRAP, 2019).

A versão 4 que foi utilizada neste trabalho suporta as versões mais recentes do Google Chrome , Firefox , Internet Explorer , Opera e Safari (exceto no Windows). Ele também suporta de volta, mas não incluindo, o IE8 e o mais recente ESR (*Firefox Extended Support Release*). O Bootstrap é modular e consiste em uma série de folhas de estilo *Less* (ou Sass na versão 4 e em diante) que implementam os vários componentes do kit de ferramentas. Essas folhas de estilo são geralmente compiladas em um pacote e incluídas em páginas da Web, mas componentes individuais podem ser incluídos ou removidos. O Bootstrap fornece várias variáveis de configuração que controlam cor e preenchimento de vários componentes, por exemplo

Ele fornece um conjunto de folhas de estilo que fornecem definições básicas de estilo para todos os principais componentes HTML. Estes fornecem uma aparência moderna e uniforme para formatar texto, tabelas e elementos de formulário. O Bootstrap vem com vários componentes JavaScript na forma de plugins jQuery . Eles fornecem elementos adicionais da interface do usuário, como caixas de diálogo, dicas de ferramentas e carrosséis. Eles também estendem a funcionalidade de alguns elementos de interface existentes, incluindo, por exemplo, uma função de preenchimento automático para campos de entrada.

Desde o 2.0, o Bootstrap suporta design web responsivo. Isso significa que o layout das páginas da Web se ajusta dinamicamente, levando em conta as características do dispositivo usado (desktop, tablet, celular). Começando com a versão 3.0, o Bootstrap adotou uma filosofia de design móvel , enfatizando o design responsivo por padrão. A versão 4.0 adicionou suporte a Sass e *flexbox.*

Todas as informações previamente descritas foram retiradas da documentação do Bootstrap (BOOTSTRAP, 2019), essa ferramenta foi escolhida para auxiliar no processo de responsividade da página, pois ela suporta um design web responsivo e dinâmico, além de uma melhor usabilidade e uma comunidade ativa onde são desenvolvidos diversos templates, que são disponibilizados de forma gratuita, fazendo com que o usuário possua uma melhor experiência em navegação.

- **GitHub**

Outra ferramenta muito importante que foi utilizada durante o desenvolvimento dessa aplicação foi o GitHub, que é uma plataforma de colaboração e controle de versão baseada na *Web* para desenvolvedores de *software*. A Microsoft, a maior colaboradora individual do GitHub, iniciou uma aquisição do GitHub por US $ 7,5 bilhões em junho de 2018. O GitHub, entregue por meio de um modelo de negócios de software como serviço (SaaS), foi iniciado em 2008 e foi fundado em Git, um sistema de gerenciamento de código aberto criado por Linus Torvalds para criar software mais rápido (Rouse, 2018).

O Git é usado para armazenar o código-fonte de um projeto e rastrear o histórico completo de todas as alterações. O autor afirma que ele permite que os desenvolvedores colaborem em um projeto de maneira mais eficaz, fornecendo ferramentas para gerenciar alterações possivelmente conflitantes de vários desenvolvedores. O GitHub permite que os desenvolvedores alterem, adaptem e melhorem o software de seus repositórios públicos gratuitamente, mas cobra por repositórios privados, oferecendo vários planos pagos. Cada repositório público ou privado contém todos os arquivos de um projeto, bem como o histórico de revisão de cada arquivo. Os repositórios podem ter vários colaboradores e podem ser públicos ou privados.

O GitHub facilita a codificação social, fornecendo uma interface da Web para o repositório de código Git e ferramentas de gerenciamento para colaboração. O GitHub pode ser considerado um site de rede social sério para desenvolvedores de software. Os membros podem seguir um ao outro, avaliar o trabalho uns dos outros, receber atualizações para projetos específicos e se comunicar publicamente ou em particular.

A utilização do GitHub foi de extrema relevância, pois a todo momento estamos suscetíveis a imprevistos onde podemos perder o nosso Código Fonte, ou realizar alterações indesejadas, e tendo acesso ao controle de versionamento do código esses imprevistos podem ser amenizados. A figura 37 mostra o controle de versionamento da aplicação no Github .

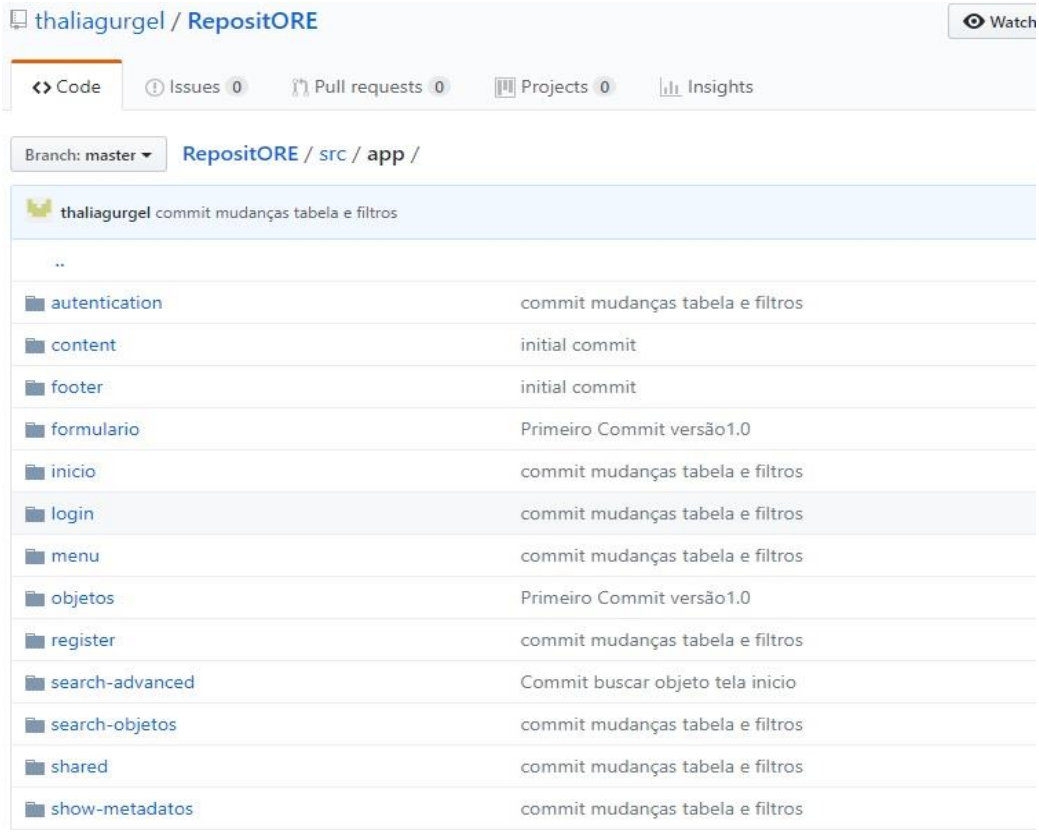


Figura 37: Controle de versionamento da aplicação utilizando Github

Fonte: Autoria Própria

- **VSCODE**

Para a codificação do Repositório, foi utilizado o vscode (*Visual Studio Code)*. O Visual Studio Code é um editor de código-fonte leve, mas poderoso, que é executado em sua área de trabalho e está disponível para Windows, macOS e Linux. Ele vem com suporte embutido para JavaScript, TypeScript e Node.js e possui um rico ecossistema de extensões para outras linguagens (como C ++, C #, Java, Python, PHP, Go) e tempos de execução (como .NET e Unity) (VSCODE, 2019).

O Visual Studio Code foi anunciado, com uma versão de previsão lançada, em 29 de abril de 2015 pela Microsoft na conferência Build de 2015. Em 18 de novembro de 2015, o Visual Studio Code foi lançado sob a licença MIT e o seu código-fonte foi postado no GitHub. Suporte para extensões também foi anunciada (LARDINOIS, 2015).

Em sua essência, o Visual Studio Code possui um editor de código fonte rápido e adequado para o uso no dia-a-dia. Com suporte para centenas de idiomas, o VS

Code ajuda a ser instantaneamente produtivo com realce de sintaxe, correspondência de colchetes, recuo automático, seleção de caixa, *snippets* e muito mais. O VS Code também se integra a ferramentas de criação e *script* para executar tarefas comuns, tornando os fluxos de trabalho diários mais rápidos. O VS Code tem suporte para o Git, para que se possa trabalhar com o controle de código-fonte sem deixar o editor, incluindo a visualização de diferenças de alterações pendentes (VSCODE, 2019).

A integração do Vscode com o Git facilitou muito o processo de codificação, pois a medida que o código ia sendo produzido, através de simples comandos na própria ferramenta, o código já ia sendo atualizado no Git, portanto é importante desfrutar das funcionalidades que uma ferramenta pode oferecer, pois isso facilita muito o processo de desenvolvimento. A figura 38 mostra um módulo da aplicação sendo desenvolvido no ambiente.

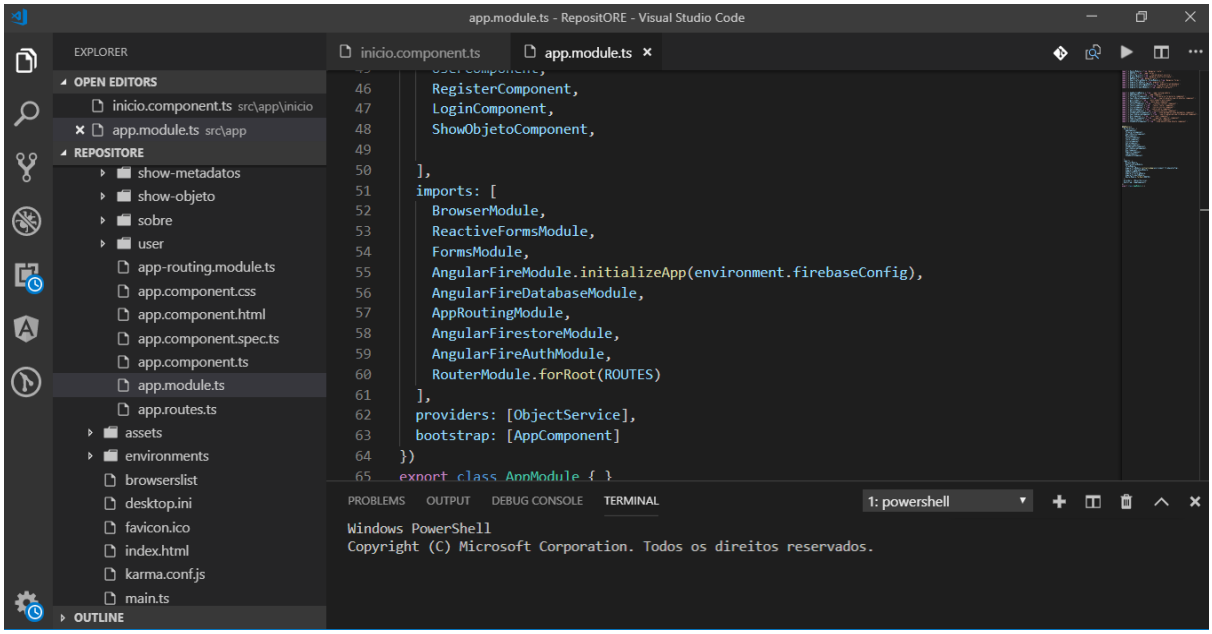


Figura 38: Ambiente de desenvolvimento Visual Studio Code

Fonte: Autoria Própria

- **NodeJs**

O Node é um ambiente de desenvolvimento em JavaScript, criado por Ryan Dahl em 2009, em sua maior parte implementada em C e C++. É uma plataforma para construir aplicações *web* escaláveis de alta performance usando JavaScript. Ele foi construído em cima da *engine* V8 que interpreta JavaScript, criado pela Google e usado em seu navegador, o Chrome (SANTOS, 2016).

O autor cita que ele usa uma arquitetura voltada a eventos, o que se integra muito bem com JavaScript (*callbacks*). Usando um *loop* de eventos, o Node interpreta, em uma única *thread*, as requisições de forma assíncrona em vez de sequenciais, e não permitindo bloqueios. Isso o torna rápido, adequado para lidar com um número alto de requisições. Apesar da ideia original ser essa, Node não é só um servidor. Com ele pode-se montar servidores http e https, assim como servidores de DNS, TCP, *Media Server* e etc.

Santos (2016) também afirma que o *loop* de eventos do Node recebe várias requisições, como por exemplo, buscar alguma informação no banco, ler um arquivo no servidor, etc. Depois de executar uma das ações, em vez de esperar a resposta (como esperar o banco responder), ele passa a processar a próxima requisição. Quando a resposta de uma delas é retornada, é disparado um evento e o *callback* correspondente da requisição, ou seja, a função que deve ser executada como resultado da resposta, é posta na fila para ser executada assim que possível.

Por causa de sua natureza, Node.js é indicado para realizar diversos tipos de projetos, como:

- APIs;
- Aplicações web *real-time* como servidores de chat ou aplicações colaborativas entre múltiplos usuários;
- Jogos *multiplayer*;
- Aplicações que demandam alta escalabilidade;
- Servidores de *streaming* de dados.

Pode-se citar algumas empresas de renome que utilizaram o node e reportaram as vantagens de terem utilizado.

- **Walmart:** Em 2013, resolveu passar todo tráfego mobile pelo Node.js na black-friday (o período mais movimentado do ano) e seus servidores não passaram de 1% de utilização, mesmo em um deploy com 200 milhões de usuários online (GLOZIC, 2014).

- **Paypal:** Desenvolveu uma mesma aplicação em Java e em Node.js e a segunda foi desenvolvida na metade do tempo com menos pessoas, com 33% menos linhas de código e 40% menos arquivos, além disso dobrou o número de requests-por-segundo em relação a aplicação Java e diminuiu em 35% o tempo de resposta (Harrell, 2013).

- **Groupon:** Diminuiu em 50% o tempo de carregamento da página (Geitgey, 2013).

- Outras empresas que o utilizam: Netflix, LinkedIn, New York Times,Flickr, Mozilla, Yahoo.

Além das tecnologias supracitadas, também podemos destacar o uso da biblioteca de funções do JavaScript (*Jquery*), a linguagem do Angular o *Typescript*, e funcionalidades do HTML e CSS, que são idiomas para descrever a apresentação de páginas da Web, incluindo cores, layout e fontes.

5.3 Resultados

Visando criar um produto com maior extensibilidade, reusabilidade e flexibilidade, o RepositORE foi construído utilizando linguagens de marcação de texto (HyperText Markup Language (HTML), alguns conceitos de Javascript, Cascading Style Sheets (CSS) voltadas ao desenvolvimento web, e também utilizando o banco de dados em tempo real Firebase, fazendo com que o sistema tenha uma resposta rápida e ágil para as requisições dos usuários.

A interface com o usuário é de vital importância para o sucesso do sistema. Principalmente por ser um sistema que não será utilizado diariamente, o usuário não possui tempo disponível para aprender como utilizar o sistema. Portanto o sistema possui uma proposta de interface amigável e intuitiva, com recursos chamativos para o público em geral, mas principalmente para o público estudantil que deseja adquirir novos conhecimentos na área de robótica.

Sua estrutura de *layout* foi realizada utilizando os frameworks Angular 7 e Bootstrap 4. Ele possui uma interface sólida, dinâmica e intuitiva, facilitando a navegação entre as páginas da aplicação. Durante toda a fase de desenvolvimento foram utilizadas as boas práticas de programação, com Typescript, e os frameworks

descritos. A figura 39 apresentada a seguir mostra a tela de início da aplicação, onde é possível acessar as páginas pela barra de navegação superior, e buscar os objetos.

Figura 39 : Tela de Início
Fonte: Autoria Própria

No campo de busca apresentado na figura 40, são mostrados os objetos cadastrados no repositório, o usuário pode digitar o objeto que deseja buscar, e partir dos filtros implementados no sistema ele irá fazer a varredura dos campos de acordo com os metadados que são inseridos na hora do cadastro.

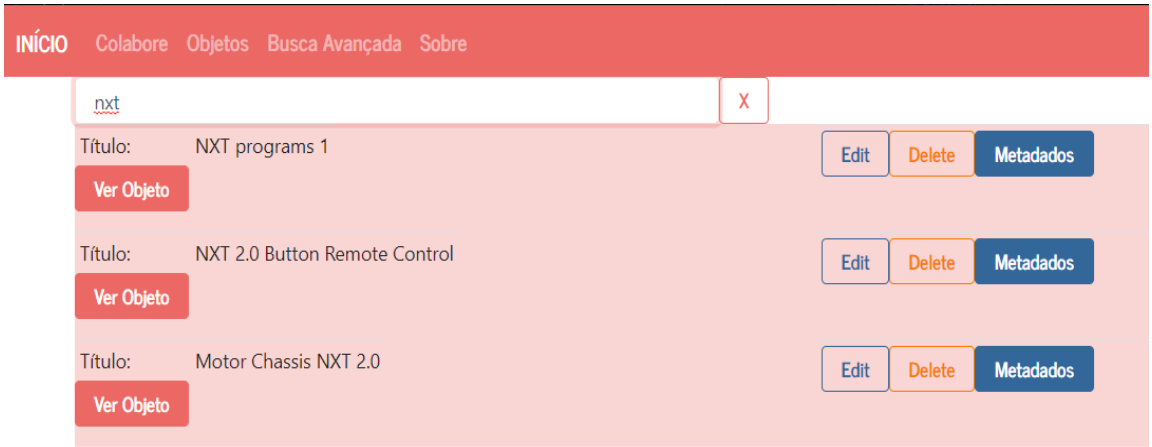


Figura 40 : Tela Pesquisar Objetos
Fonte: Autoria Própria.

Após realizar a pesquisa do objeto, ao clicar no botão ver objeto, será aberto um modal onde estão contidas todas as informações referentes ao objeto, no campo Fonte, temos o link no qual o objeto está armazenado, ao clicar nele o usuário será redirecionado para a página na qual o objeto se encontra. A figura 41 apresenta um modal de representação de descrição de um objeto.

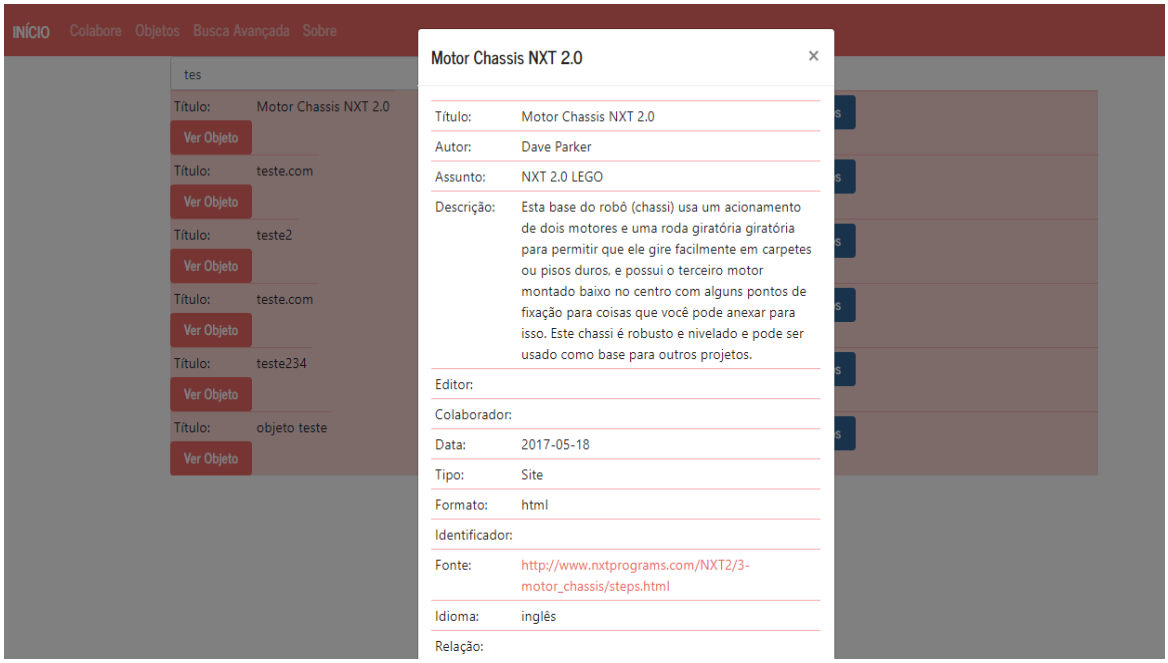


Figura 41: Visualização de Descrição do Objeto
Fonte: Autoria Própria

Ao clicar no botão de metadados, localizado no canto direito da figura 41, o usuário é redirecionado para a página onde é possível visualizar os metadados Dublin Core daquele determinado objeto, veja um exemplo na figura 42 apresentada a seguir.

Figura 42: Funcionalidade Ver metadados
Fonte: Autoria Própria

A seguir na figura 43, a tela de login do sistema, onde caso o usuário queira depositar algum objeto de robótica neste repositório, ele poderá efetuar seu login inserindo e-mail e senha, ou poderá optar pelo login utilizando as redes sociais, neste caso o Google ou o facebook, ao realizar login o usuário é redirecionado para a tela de colaboração.

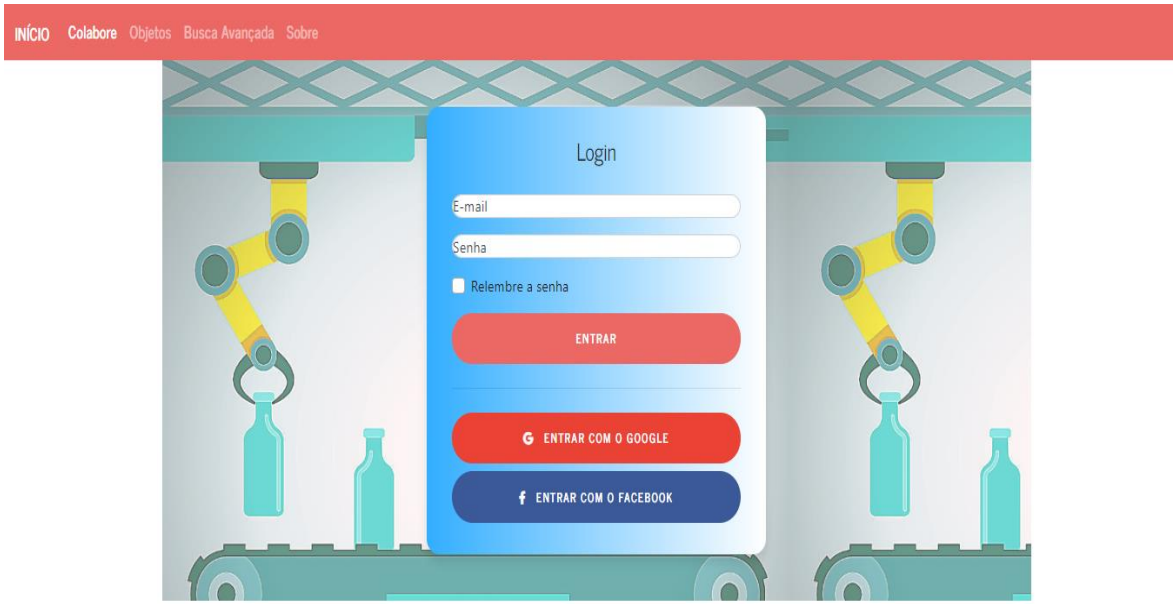


Figura 43: Tela de Login
Fonte: Autoria Própria

Na tela de colaboração apresentada na figura 44, o usuário pode preencher os campos de acordo com o que achar adequado para o seu objeto e suas funcionalidades. Não há campos obrigatórios, a não ser os de título e de descrição. Ao cadastrar o objeto no sistema ele é enviado ao banco de dados e ficará disponível para pesquisa e filtragem.

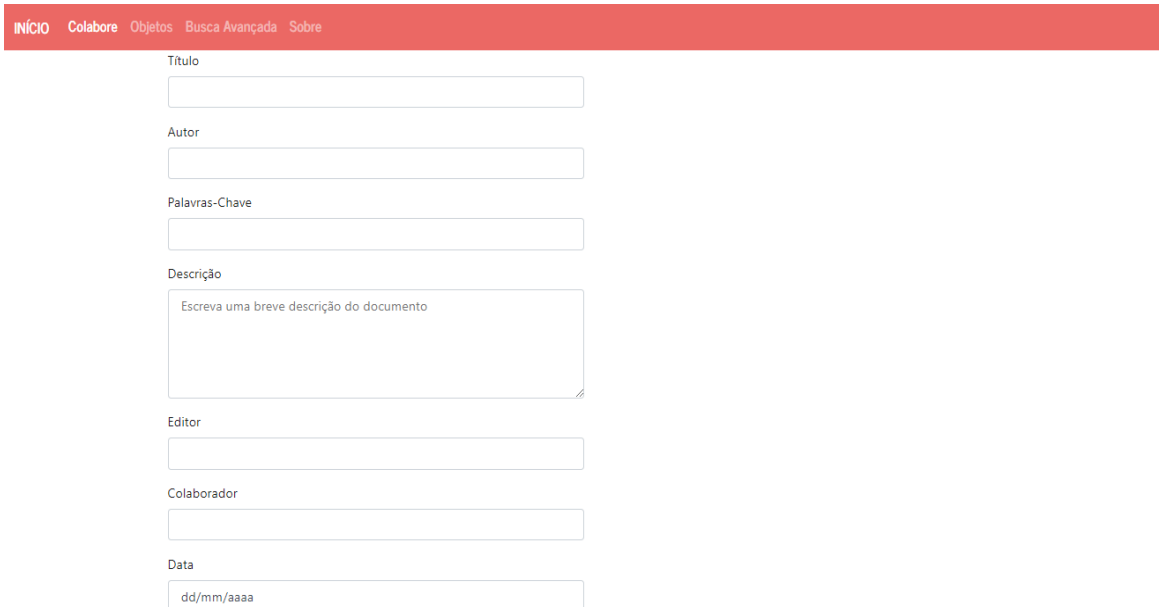


Figura 44: Tela de Colaboração/Cadastro
Fonte: Autoria Própria

Na representação a seguir como mostra a figura 45, a interface do banco de dados do sistema, o banco de dados utilizado nesta aplicação foi o *Firebase- Real time Database* (Banco de dados em tempo real) NoSql. Este é um banco de dados que fica armazenado em nuvem, é mantido pela Google, e apresenta várias funcionalidades que podem facilitar a vida do desenvolvedor. Esse banco de dados pode ser visto na figura 46, onde os objetos que são cadastrados na interface do sistema são armazenados no banco e cada um deles possui um identificador de objeto, esses identificadores são como chaves primárias, onde cada objeto possui a sua própria identificação, e ela não poderá ser duplicada no sistema.

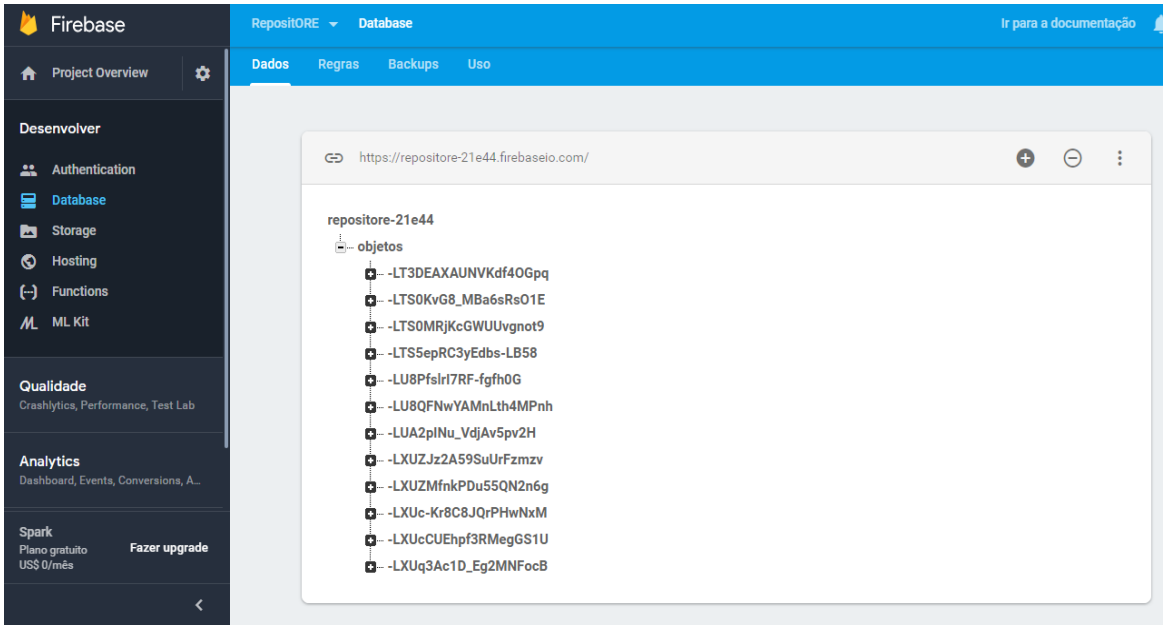


Figura 45: Interface do Banco de Dados com objetos cadastrados
Fonte: Autoria Própria

Na figura 46, temos a representação de um objeto cadastrado no sistema e como ele é armazenado no banco de dados. Os campos Dublin Core inseridos de forma indireta no sistema ficam armazenados juntamente com seus campos descritivos. Os campos que estão em branco não foram preenchidos no ato do cadastro pois não se torna obrigatório o preenchimento de todos os campos.

Os objetos que são cadastrados no banco ficam armazenados em formato de objeto JSON. De acordo com (Crockford, 2000) em computação, JSON é um acrônimo de *JavaScript Object Notation*, é um formato compacto, de padrão aberto independente, de troca de dados simples e rápida (*parsing*) entre sistemas, que utiliza texto legível a humanos, no formato atributo-valor (natureza auto-descritiva). Isto é, um modelo de transmissão de informações no formato texto, muito usado em *web services* que usa transferência de estado representacional (REST) e aplicações AJAX, substituindo o uso do XML. Padrão foi especificado em 2000 e, definido em 2013 nos dois padrões concorrentes, RFC 7159 e ECMA-404.

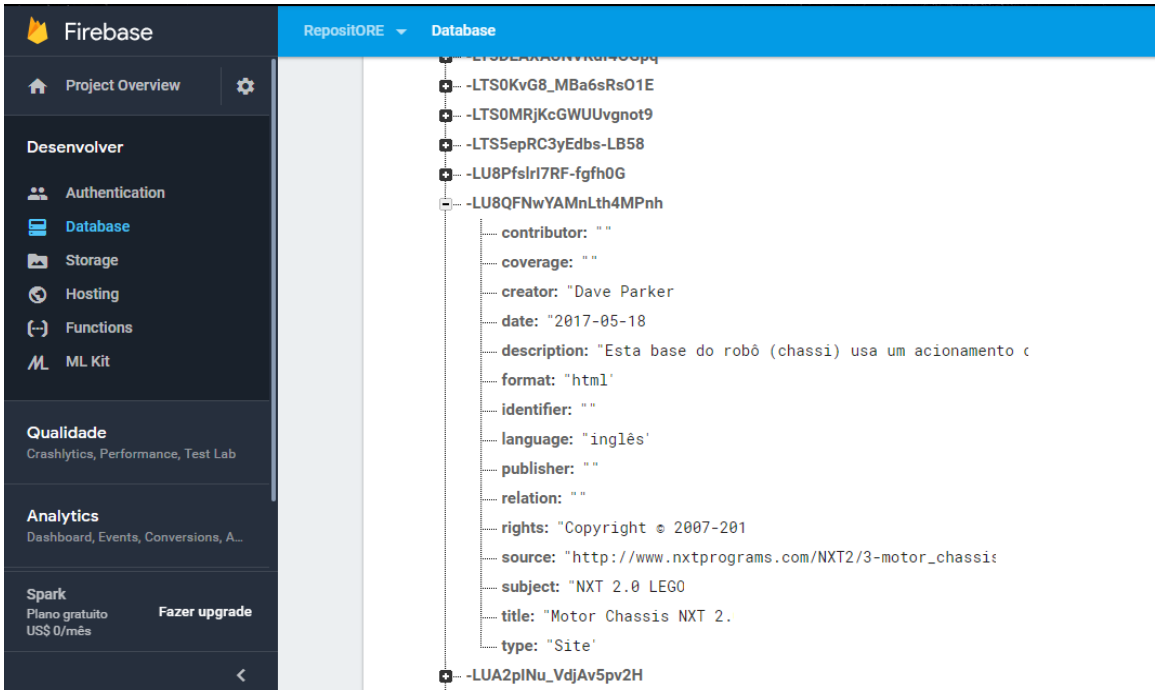


Figura 46: Objetos descritos no Banco de Dados em formato Json
Fonte: Autoria Própria

6 CONCLUSÃO

O uso de tecnologias no meio educacional abriu uma grande quantidade de possibilidades para que a Robótica Educacional pudesse ser melhor trabalhada nos mais diversos meios de aprendizado. Visto que a Robótica Educacional potencializa conhecimentos, e melhora o desenvolvimento físico e mental de estudantes que trabalham com a resolução dos problemas propostos pelo ensino da robótica.

O desenvolvimento deste trabalho se orientou pela necessidade de que os conhecimentos sobre robótica educacional fossem organizados e bem descritos em um repositório de objetos de aprendizagem, apresentando metadados descritivos com o objetivo de facilitar a busca por esses objetos.

Neste trabalho foi proposto o projeto de desenvolvimento de um ambiente voltado para a *web*, com o objeto de dar suporte a atividades voltadas para Robótica Educacional, também foi proposto um mapeamento de descrição de metadados descritivos para o armazenamento dos objetos no ambiente, a partir do mapeamento que foi baseado no padrão Dublin Core *Metadata.* Os dados são inseridos no repositório e outros usuários podem ter acesso a essa informação de maneira mais simplificada.

A partir da modelagem dos diagramas do sistema, o sistema foi implementado utilizando tecnologias de código aberto disponíveis para desenvolvimento focando em um *front-end* amigável e um *back-end* usando formatos simples para obtenção de melhor desempenho

6.1 Trabalhos Futuros

Como proposta de trabalhos futuros para esta monografia partir dessa versão inicial, pôde-se notar que várias melhorias são necessárias, como a inserção de novas funcionalidades, como uma tela inicial com objetos mais relevantes e objetos mais acessados, implementação da funcionalidade de sugestão de objeto a partir de determinado conhecimento. Caso o usuário necessite de algum conhecimento prévio para entender aquele problema, essas sugestões também poderiam ser apresentadas.

Também é intenção futura realizar um teste de usabilidade para aperfeiçoar a interface do sistema, tornando-a cada vez mais dinâmica e intuitiva, apresentando um melhor desempenho e navegabilidade. É imprescindível que após o desenvolvimento completo do sistema obtenha-se uma validação com usuários reais do sistema, possibilitando melhorias e sugestões para o crescimento e manutenabilidade da aplicação.

REFERÊNCIAS

ADVANCED DISTRIBUTED LEARNING. **Sharable Content Object Reference Model-SCORM** (2004) 2nd Edition - Overview.

ALVES, R. M. ; SILVA, A. L. C. ; PINTO, M. C. ; SAMPAIO, F. F. ; ELIA, M. F. **Uso do hardware livre Arduino em ambientes de ensino-aprendizagem.** In: CONGRESSO BRASILEIRO DE INFORMÁTICA DA EDUCAÇÃO, 2012, Rio de Janeiro; JAIE - JORNADA DE ATUALIZAÇÃO EM INFORMÁTICA E EDUCAÇÃO, 2012, Rio de Janeiro. Anais... Rio de Janeiro: CEIE, 2012.

ALTEXSOFT. 2018. Disponível em: https://www.altexsoft.com/blog/engineering/the-good-and-the-bad-of-angular-development/. Acesso em: 24 jan 2019.

AMAZON. 2019. Disponível em: https://www.amazon.com/HEXBUG-VEX-Robotics-Construction-Kit/dp/B00ON4F7SE. Acesso em 18 jan, 2019.

ANGULARJS. 2019. Disponível em: https://docs.angularjs.org/guide/introduction. Acesso em: 24 jan 2019

ARS CONSULT. Apostila de Introdução a Robótica. Recife, 1995.

ARS Consult. Disponível em: www.arsconsult.com.br. Acesso em: 18 jan. 2019.

ARDUINO. 2019. Disponível em:https://www.arduino.cc/. Acesso em 26 jan 2019

BARRIOS ARANIBAR, Dennis, GURGEL, Viviane, SANTOS, Marcela, ARAUJO. Gianna R., ROZA, Valber C., NASCIMENTO, Rafaella A., SILVA, Alzira F. da, SILVA, Akynara & NASCIMENTO, Luis M. G. 2006, **Roboeduc: A software for teaching robotics to technological excluded children using lego prototypes**, *em* '3rd IEEE Latin American Robotics Symposium', Santiago,Chile.

BARROS, Renata Pitta. **Roboeduc- uma ferramenta para programação de robôs lego,** Relatório técnico, Universidade Federal do Rio Grande do Norte. 2008.

Behar, P. A. e Torrezan, C. A. W. (2009) **"Parâmetros para a construção de materiais educacionais digitais do ponto de vista do design pedagógico**", In: Behar, P. A. (Cols.). Modelos Pedagógicos em Educação a Distância. v.1. Porto Alegre: Artmed., p.33-65.

BESAFE. **A casa do Cyberbox**. Disponível em: www.cyberbox.com.br. Acesso em: 15 jan, 2019.

BOOTSTRAP. 2019. Disponívem em: https://getbootstrap.com/docs/3.3/about/. Acesso em: 24 jan 2019

BRAGA, Juliana Cristina et al. **Desafios para o desenvolvimento de objetos de aprendizagem reutilizáveis e de qualidade**. In: DESAFIE! 2012, Curitiba. Anais Curitiba/PR:CEIE/SBC, 2012. p. 90-99.

BRAGA, J. (2014) "**Objetos de Aprendizagem. Volume 1 - Introdução e Fundamentos"**, p.65, p.29.

BRICKOS. '**Brickos operating system and c/c++ development environment for the lego mindstorms rcx controller**', 2019. Disponível em: http://brickos.sourceforge.net/. Acesso: 18 jan 2019.

CABRAL, Cristiane Pelisolli. Robótica Educacional e Resolução de Problemas: Uma abordagem microgenética da construção do conhecimento. 2010.

CASTRO, Viviane Gurgel de. **RoboEduc: Especificação de um Software Educacional para o ensino de Robótica às Crianças como Uma Ferramenta de Inclusão Digital.** Natal. 2008.

CENDÓN, B. V. **Ferramentas de busca na Web. Ciência da Informação**, Brasília, v. 30, n. 1, p. 39-49, 2001.

CHAVES, Leonardo. Disponível em: https://leonardochavesd.wordpress.com/2010/05/25/robotica/. Acesso em: 18 jan. 2019

COELHO, Alexandre Ferreira. 2015. Disponível em: https://luthortronics.com.br/qual-arduino-comprar-conheca-os-tipos-de-arduino/. Acesso em: 26 jan 2019.

Congresso Brasileiro de Biblioteconomia, Documento e Ciência da Informação – Florianópolis, SC, Brasil, 2013.

COMPLEX. 2019. Disponível em: http://www.complex.com.br. Acesso em: 18 jan 2019.

CROCKFORD, Douglas. 2000. Disponível em: https://tools.ietf.org/html/rfc4627. Acesso em 30/01/ 2019

D'ABREU, J. V. V. **Robótica pedagógica: Percurso e perspectivas.** 2014. Workshop de Robótica Educacional.

DANIELA, Pires. **USO DO DUBLIN CORE NA DESCRIÇÃO DE OBRAS RARAS NA WEB: A COLEÇÃO DA BIBLIOTECA BRASILIANA DIGITAL**, 2012.

DALZIEL, James et al. **Reflections on the colis (collaborative online learning and information systems) demonstrator project and the" learning object lifecycle**".

In: ASCILITE. 2002. p. 159-166. Disponível em: http://www.ascilite.org.au/conferences/auckland02/proceedings/papers/207.pdf

DCMI. **Dublin Core Metadata Initiative**. 2019. Disponível em: http://dublincore.org/documents/dcmes-qualifiers/. Acesso em: 26 jan 2019.

DCMI. **Dublin Core Metadata Initiative**. 2001. Disponível em: http://www.dublincore.org/specifications/dublin-core/usageguide/2001-04-12/. Acesso em 21/05/2019.

DCMI. **Dublin Core Metadata Initiative**. 1999. Disponível em: http://www.dublincore.org/specifications/dublin-core/dces/1999-07-02/. Acesso em 21/05/2019.

DICIONÁRIO AURÉLIO FERREIRA, AB de H. **Dicionário aurélio eletrônico**. Ed. Nova Fronteira, 1993. Disponível em: http://www.dicionariodoaurelio.com/ Acesso em 10/01/2019.

DICIONÁRIO INTERATIVO DA EDUCAÇÃO BRASILEIRA. Agência Educa Brasil. Disponível em: http://www.educabrasil.com.br/cat/dic/. Acesso em: 15 jan. 2019.

EDACOM, Tecnologia. Disponível em: www.edacom.com.br. Acesso em: 16 jan, 2019.

EDACOM, Tecnologia., **Robolab.** 2019. Disponível em: http://www.edacom.com.br. Acesso em: 19 jan 2019.

ERCEGOVAC, Zorama. **Introduction. Journal of the American Society for Information Science**, v.50, n.13. 1165-1168, 1999.

FABRICIO, FABIANO, LUCIA**. Linguagens de Programação Logo para a Educação**. 2012. Disponível em : http://linguagemlogounisc2012.blogspot.com/p/megalogo.html. Acesso em: 27 jan 2019.

GALAFASSI, F. P., Gluz, J. C. e Galafassi, C. (2013) **"Critical Analysis of Recent Researches on Learning Objects and Learning Environments Technologies**". Brazilian Journal of Computers in Education, [S.l.], v. 21, n. 03, p. 100, dec. 2013. ISSN 2317-6121. Disponível em: <http://www.br-ie.org/pub/index.php/rbie/article/view/2351/2457>.

GEEKYANTS. **A brief post about what Firebase is all about, and it's new NoSQL Database—Cloud Firestore**. 2017. Disponível em: https://hackernoon.com/introduction-to-firebase-218a23186cd7. Acesso em: 23 jan 2019

GEITGEY, Adam. **I-Tier: Dismantling the Monolith**.2013. Disponível em: https://engineering.groupon.com/2013/misc/i-tier-dismantling-the-monoliths/. Acesso em: 23 jan 2019.

GILL, T. **Los metadatos y la World Wide Web**. **In: BACA, M. (Ed.). Introducción a los metadatos vías a la informacíon digital**. Traducido al español por Marisol Jacas-Santoll. Los Angeles, CA: J. Paul Getty Trust, 1998. p. 10-20.

GILLILAND-SWETLAND, A. J. **La definición de los metadatos. In: BACA, M. (Ed.). Introducción a los metadatos vías a la informacíon digita**l. Traducido al español por Marisol Jacas-Santoll. Los Angeles, CA: J. Paul Getty Trust, 1998. p. 1-9.

GLOZIC, Dejan. NodeDay 2014. 2014. Disponível em: https://dejanglozic.com/2014/03/04/nodeday-2014/. Acesso em: 23 jan 2019.

GODOFREDO, Siumar, ROMANÓ, Rosana e ZILLI, Silvana. **Robótica Pedagógica – uma aplicação de inteligência artificial**. 2001. Artigo apresentado na disciplina de Engenharia do Conhecimento. Programa de Pós-graduação em Engenharia de Produção, UFSC, Florianópolis.

GODOY, Norma. **Curso de Robótica Pedagógica**. Apresentação em Power Point. Curitiba: Empresa Ars Consult, 1997.

GOGOBOARD. Disponível em: https://gogoboard.org/. Acesso em: 18 jan. 2019.

GONÇALVES, Paulo de Castro. *et al.* **Adequação do Dublin Core ao AACR2: o caso da Biblioteca Digital da Assembleia Legislativa do Estado de Minas Gerais.** XXV

GONÇALVES, T. B. (2011) "**Estudo exploratório sobre padrões de objetos de aprendizagem para ambientes colaborativos de aprendizado eletrônico**".

GRÁCIO, J. C. A. **Metadados para a descrição de recursos da Internet: o padrão Dublin Core, aplicações e a questão da interoperabilidade**.2002.

GRÁCIO, J. C. A. (2012) "**Preservação digital na gestão da informação: um modelo processual para as instituições de ensino superior".** São Paulo, SP: Cultura Acadêmica. p.214. Disponível em: http://www.culturaacademica.com.br/_img/arquivos/Preservacao_digital_na_gestao_da_informacao-WEB_v2.pdf

HARRELL, Jeff. **Node.js at Paypal**. 2013. Disponível em: https://www.paypal-engineering.com/2013/11/22/node-js-at-paypal/. Acesso em: 23 jan 2019.

HEERY, Rachel; ANDERSON, Sheila. **Digital repositories review**. 2005. Joint Information Systems Committee: UKOLN.

HILLMANN, D. **Using Dublin Core**. Nov. 2005a. Disponível em: http://dublincore.org/documents/usageguide/

IEEE. (2002) "**IEEE Standard for Learning Object Metadata**". New York. Disponível em: https://standards.ieee.org/reading/ieee/std/learning/1484.12.1-2002.pdf.

IETF 2019. Disponível em: https://www.ietf.org/rfc/rfc2413.txt. Acesso em: 21/05/2019

KLOC, A. E; KOSCIANSKI, A e PILATTI, L. A. (2009). **Robótica: Uma ferramenta pedagógica no campo da computação**. In: I Simpósio Nacional de Ensino de Ciência e Tecnologia.

KOPER, R. (2002) "**Educational Modelling Language: Adding instructional design to existing specifications**". Open University of the Netherlands.

KRUTCHEN, P. (2003) "**The rational unified process: an introduction**". 3 ed. Addison Wesley.

LARDINOIS, Frederic. Microsoft Launches Visual Studio Code, **A Free Cross-Platform Code Editor For OS X, Linux And Windows**. 2015. Disponível em: https://techcrunch.com/2015/04/29/microsoft-shocks-the-world-with-visual-studio-code-a-free-code-editor-for-os-x-linux-and-windows/. Acesso em 24 jan 2019

LEGO. 2019. Disponível em: https://www.lego.com/pt-br/aboutus/lego-group/the_lego_history. Acesso em : 16 jan.2019.

LEGO. **Lego mindstorms**. 2019. Disponível em: http://mindstorms.lego.com. Último acesso em 18 jan 2019.

LEGO. **Lego Mindstorms History**. Disponível em: https://www.lego.com/pt-br/mindstorms/history?ignorereferer=true. Acesso em: 26 jan 2019.

LEGO BOOST. 2019. Disponível em: https://www.lego.com/pt-br/themes/boost/products. Acesso em: 17 jan. 2019

LEGO ENGINEERING. 2019. Disponível em: http://www.legoengineering.com/platform/robolab/. Acesso em: 19 jan 2019

MAISONNETTE, Roger. **A utilização dos recursos informatizados a partir de uma relação inventiva com a máquina: a robótica educativa**, 2002. In: Proinfo – Programa Nacional de Informática na Educação – Paraná. Disponível em: www.proinfo.gov.br. Acesso em: 15 jan. 2019.

MCGREAL, Rory**. A typology of learning object repositories**. In: Handbook on information technologies for education and training. Springer Berlin Heidelberg,

2008. p. 5-28. MEC. **Banco Internacional de objetos educacionais.**2019. Disponível em: http://objetoseducacionais2.mec.gov.br/. Acesso em: 26 jan 2019.

MODELIX, Robotics. 2019 Disponível em: https://www.modelix.com.br/software-de-programacao-robotica. Acesso em: 17 jan. 2019

MONK, Simon. **30 PROJETOS COM ARDUINO**, tradução: Anatólio Laschuk 2ª edição, Editora Bookman, Porto Alegre 2014.

MOREIRA, Anderson Harayashiki. **Utilizando o software de programação do EV3 com o NXT**, 2015. Disponível em: http://aprendarobotica.com.br/utilizando-o-software-de-programacao-do-ev3-com-o-nxt/. Acesso em: 17 jan 2019.

NASCIMENTO, A; MARIETTO, M. G. B; SUYAMA. R; BOTELHO, W. T Capitulo 9 **Modelagem e Simulação Computacional: Conceitos Fundamentais**. In: Maria das Graças Bruno Marietto; Mário Minami; Pieter Willem Westera. (Orgs.). BASES COMPUTACIONAIS DA CIÊNCIA. 1.ed.Santo André: Universidade Federal do ABC, 2013, v.1, p. 1-241.

NQC. **Not quite c language with a c-like syntax for the lego mindstorms rcx controller.** 2019. Disponível em: http://bricxcc.sourceforge.net/nqc/. Acesso em 18 jan 2019.

PAPERT, Seymour**. A Máquina das Crianças: repensando a escola na era da informática**. Artes Médicas. Porto Alegre. 1994.

PAPERT, Seymour. **LOGO: Computadores e Educação**. São Paulo: Brasiliense, 1986.

PEREIRA, Manacés**. Angular vs Reactjs - A web sempre ganhando**. 2018. Disponível em: https://tableless.com.br/angular-vs-reactjs-a-web-sempre-ganhando/. Acesso em: 24 jan 2019.

PEYROTT, Sebastian. **More Benchmarks: Virtual DOM vs Angular 1 & 2 vs Others.** 2016. Disponível em: https://auth0.com/blog/more-benchmarks-virtual-dom-vs-angular-12-vs-mithril-js-vs-the-rest/. Acesso em: 24 jan 2018.

PIMENTEL, E. BRAGA, J. C**.; Fundamentos da computação.** In: MARIETTO, Maria das Graças Bruno; MINAMI, Mário; WESTERA, Pieter Willem. (Orgs.). Bases com putacionais da ciência. 1.ed. Santo André: Universidade Federal do ABC, 2013, v.1, p.1-241.

Portal Educação 2013 Disponível em: https://www.portaleducacao.com.br/conteudo/artigos/direito/o-que-e-videoaula/41495 Acesso em: 10/01/2019

Porto Editora, 2003-2019. [Acesso em 2019-01-10 13:35:28]. Disponível na Internet: https://www.infopedia.pt/dicionarios/lingua-portuguesa/hipertexto

Porto Editora, 2003-2019. [Acesso em. 2019-01-10 13:21:03]. Disponível na Internet: https://www.infopedia.pt/dicionarios/lingua-portuguesa/animação

RESMINI, R; ROCHA, K. C; CORDEIRO, M. S. **Robótica no Ensino Médio**. 2018.

RIVED. **Rede Interativa Virtual de Educação**. 2019. Disponível em: Rede http://rived.mec.gov.br/. Acesso em: 26 jan 2019.

RNP. 2019. Disponível em: https://www.rnp.br/destaques/voce-conhece-diferenca-entre-recursos-educacionais-abertos-e-objetos-aprendizagem, Acesso em: 21 mai 2019.

ROBOEDUC. 2019. Disponível em: http://www.roboeduc.com.br/. Acesso em: 26 jan 2019.

ROBOKIT. 2019. Disponível em: http://www.imply.com/pt/?s=robokit. Acesso em: 26 jan 20191.

ROSETTO, Marcia; NOGUEIRA, Adriana Hipólito. **Aplicação de elementos metadados DUBLIN CORE para descrição de dados bibliográficos on-line da biblioteca digital de teses da USP**. Anais.. Recife: SNBU, 2002.

ROUSE, Margaret. **Github. 2018**. Disponível em: https://searchitoperations.techtarget.com/definition/GitHub. Acessado em: 24 jan 2019

SÁ, Sarah Thomaz de Lima. **W-Educ: Um Ambiente Web, Completo e Dinâmico para Robótica Educacional.** Natal. 2016.

SANTOS, Guilherme. **Node.js—O que é, por que usar e primeiros passos**. 2016. Disponível em: https://medium.com/thdesenvolvedores/node-js-o-que-%C3%A9-por-que-usar-e-primeiros-passos-1118f771b889. Acesso em: 23 jan 2019

SARAIVA, I. B. e Netto, C. M. (2010) "**Monitor: um conjunto de objetos de aprendizagem para apoio ao ensino de programação de computadores",** In: XXX Congresso da Sociedade Brasileira de Computação. No XVIII Workshop sobre Educação em Computação.

SILVA, Alzira Ferreira da. **Roboeduc: Uma Metodologia de Aprendizado com Robótica Educacional.** Natal. 2009.

STACKOVERFLOW. 2018. Disponível em: https://insights.stackoverflow.com/survey/2018/. Acesso em 30 jan 2018

SUTTON, Stuart A. **Conceptual design and development of a metadata framework for educational resources on the Internet**. Journal of the American Society Information Science, v.50, n.13, p.1182-1192, 1999.

UFSC**. Repositório de objetos de aprendizagem da EAD- UFSC**. 2019. Disponível em: https://uab.ufsc.br/. Acesso em: 26 jan 2019.

VEIGA, Fernando. Robótica - **Movimento Mindstorms**. Disponível em: https://imasters.com.br/tecnologia/robotica-movimento-mindstorms. Acesso em: 17 jan.2019

VEX. **VEX ROBOTICS.** 2019. Disponível em: https://www.vexrobotics.com/. Acesso em: 26 jan 2019.

VEX, IQ. 2019. Disponível em: https://www.vexrobotics.com/vexiq. Acesso em 18 jan. 2019

VSCODE. **Visual Studio Code.** 2019. Disponível em:https://code.visualstudio.com/docs. Acesso em: 24 jan 2019

WILEY, David. **Learning objects need instructional design theory**. The ASTD e-Learning handbook, p. 115-126, 2002.

ZACHARIAS, Vera L. C. 2003. **A linguagem logo.**

ZILLI, Silvana. **Apostila de Robótica Educacional. Expoente Informática**. Curitiba: Gráfica Expoente, 2002.

APÊNDICE

APÊNDICE A - Mapeamento dos dados do Repositório de objetos de RE ao padrão DC

Elementos DC	Campo DC	Dado do Objeto	Descrição
Titulo	dc.title	Título	Título do Objeto ou do projeto (ex: Carro de controle remoto, Robô aranha, etc)
	dc.title.subtitle	Subtítulo	Subtítulo do Objeto
Criador	dc.creator	Autor	O autor do objeto de aprendizagem, ou responsável por submeter ele para publicação.
	dc.creator.authority	Autoridade	Responsável por prescrever os objetos de aprendizagem e os resultados
	dc.creator.search_information	Buscador de informação	Responsável por buscar por recursos de acordo com metadados fornecidos.
	dc.creator.apprentice	Aprendiz	Aquele que irá utilizar o OA.
	dc.creator.organizer	Organizador	Responsável por projetar atividades de aprendizagem e revisar as licenças e direitos autorais de uso.
Assunto	dc.subject.theme	Assunto ou tema	Descrever o tema do objeto ou um conteúdo informativo sobre de que ele se trata.
Descrição	dc.description.keyword	Palavra-chave	Palavra-chave relacionada ao objeto (ex: lego, arduino, sensor)
	dc.description.general	Descrição Geral	Breve descrição geral sobre do que se trata o objeto do objeto
	dc.description.building	Construção	Breve descrição de como o objeto é construído.

	dc.description.type_programming	Tipo de Programação	Tipo de programação do objeto, ex: Texto, blocos.
	dc.description.program_objective	Objetivo Do programa	Se o objetivo do programa é fazer um movimento específico, ou uma funcionalidade específica.
	dc.description.educational_objective	Objetivo Educacional	Objetivo educacional do objeto, por ex: Interatividade, Autonomia, Cooperação, Cognição, Afetividade.
	dc.description.requirements	Requisitos Necessários	Requisitos necessários para construção do objeto, ex: Conhecimentos específicos, ou ferramentas específicas.
Editor	dc.publisher.editor	Editor	Responsável pela edição do objeto, pode ser uma pessoa, organização ou serviço.
Contribuinte	dc.contributor	Contribuinte	Membro(s) responsável pelo desenvolvimento do objeto.
	dc.contributor.partner	Instituição parceira	Instituição na qual foi desenvolvida o objeto.
Data	dc.date.creation	Data de Criação	Data referente a criação do objeto
	dc.date.availability	Data de Disponibilização	Data referente a disponibilização do objeto no repositório
Tipo	dc.type.kit	Tipo do kit utilizado	O tipo do objeto pode ser Lego, arduino, modelix, VEX, etc.
	dc.type.version	Versão	Ex: Versão 1.0, 2.0, 3.0
	dc.type.model	Modelo	Modelo do objeto,ex: NXT, Boost, EV3, Arduino nano, etc.
	dc.type.language	Linguagem	Linguagem utilizada para construção do programa, ex: Java, C, C++, Blocos, linguagem visual.

	dc.type. activity	Tipo de atividade	O tipo da atividade pode ser montagem, programação, ou outra tarefa específica.
	dc.type. age	Idade recomendada	Idade recomendada para utilização do objeto ex: 7, 8, 10 anos.
	dc.type .level	Nível de Ensino	Se o objeto é recomendado para algum nível de ensino específico ex: ensino fundamental, ensino médio, educação infantil etc.
	dc.type. programming_ environment	Ambiente de Programação	Qual ambiente de programação é utilizado pelo objeto, ex: Ide Arduino, NXT, Everest, Robolab, etc.
	dc.type. user	Usuário Final	Objeto recomendado para professor, aluno ou qualquer tipo de usuário.
	dc.type. software	Software	Software utilizado para realização dos movimentos, ex: IDE, Robolab, NXT.
Formato	dc.format	Formato do objeto	Formato em que o objeto está sendo disponibilizado, ex: aplicação, áudio, imagem, vídeo, página web, etc.
Identificação	dc.identifier. area	Área de Ensino	Área de ensino que o objeto pode ser trabalhado, ex: física, matemática etc.
Fonte	dc.source. location	Localização do objeto	Local onde o objeto foi publicado, link do site ou página de onde o objeto está sendo disponibilizado.
Idioma	dc.language	Idioma	Idioma em que o objeto é disponibilizado
Relação	dc.relation	Referência a outro objeto	Relação que o objeto tem com outro, podendo ser colocado como uma referência.
Direitos	dc.rights	Direitos Autorais	Inserir os direitos de propriedade intelectual sobre o objeto.

.