

UNIVERSIDADE DE CAXIAS DO SUL

LUAN LUCAS PEREIRA DE LIMA

**APLICATIVO EDUCACIONAL PARA INSERÇÃO DA LÓGICA COMPUTACIONAL
PARA CRIANÇAS**

**Caxias do Sul
2017**

Luan Lucas Pereira de Lima

**APLICATIVO EDUCACIONAL PARA INSERÇÃO DA LÓGICA COMPUTACIONAL
PARA CRIANÇAS**

Trabalho de Conclusão de Curso para obtenção
do título de Bacharel em Ciência da
Computação pela Universidade de Caxias do
Sul.

Orientador:
Prof. Dr. Carine Geltrudes Webber

**Caxias do Sul
2017**

RESUMO

A introdução da lógica computacional para crianças proporciona o desenvolvimento de habilidades como, o pensamento computacional e o raciocínio lógico. Estas são habilidades importantes, principalmente na atualidade, sendo que a tecnologia está cada vez mais presente no cotidiano das pessoas. Elas estão diretamente ligadas ao raciocínio necessário para a modularização, organização, resolução de problemas e são essenciais para todos os profissionais no mercado de trabalho. Diversas são as dificuldades encontradas para a inserção da lógica computacional na educação infantil. Os entraves têm início em ensinar alunos na educação infantil, passam pela falta de ferramentas orientadas para crianças nesta faixa de idade, e também estão presentes na dificuldade de organizar as aulas e obter dados sobre o desenvolvimento dos alunos. A utilização de softwares isolados e a robótica educacional são capazes de auxiliar o ensino. Tais instrumentos estão inseridos em conceitos como a aprendizagem significativa, e a inserção do lúdico no ensino e contribuem de maneira fundamental para inserção da lógica computacional. A fim de contribuir para a inserção da lógica computacional no ensino, o presente trabalho tem o objetivo de desenvolver um aplicativo para a plataforma móvel *Android*. O aplicativo conta com uma comunicação remota com um robô através da tecnologia *bluetooth*. O robô foi desenvolvido utilizando a plataforma *Arduino*. Ele realiza os comandos enviados pelo aplicativo com o objetivo de realizar um cenário de utilização. Os cenários de utilização são compostos por um ponto de partida e terminam após completar todos os objetivos. Os alunos devem utilizar comandos lógicos de movimentos espaciais como: cima, esquerda e direita para cumprir os objetivos de cada cenário. Os cenários de utilização podem ser criados e modificados pelo professor, para direcionar o ensino segundo as necessidades da turma. Os dados oriundos da resolução dos cenários são armazenados em um banco de dados nativo do *Android*, para que posteriormente possa ser verificado o progresso dos alunos através de consultas realizadas no aplicativo. O projeto foi desenvolvido e experimentado em sala de aula com alunos da educação infantil. Foram avaliados os alunos com base em seu entendimento no funcionamento do projeto como um todo e no tempo de resolução dos cenários de utilização. Nesses aspectos o projeto atingiu os objetivos esperados, como a introdução de conceitos básicos de ciência da computação para crianças.

Palavras-chave: Software Educacional. Robótica Educacional. Programação para Crianças. Ensino de Programação.

ABSTRACT

An introduction to computational logic for children providing the development of skills such as computational thinking and logical reasoning. These are important nowadays as technology is increasingly present in people's daily lives. They are directly linked to the reasoning required for modularization, organization, problem solving, and are essential for all professionals not in the job market. Several are the difficulties found for the introduction of computational logic in the childhood education. The difficulties begin in teaching students in early childhood education, in addition to the lack of tools aimed at children in this age group, and are also present in the difficulty of organizing classes and obtaining data on the students' development. The use of isolated software and educational robotics aids teaching. Such instruments are embedded in concepts such as meaningful learning, and an insertion of playful non-teaching and contribution to the fundamental form for the introduction of computational logic. In order to contribute to this introduction of the computational logic for children, the present work presents an application for the Android mobile platform. The application design relies on a remote communication with a robot through the bluetooth technology. The robot was developed using an Arduino platform. It allows to comply with the commands sent to the application in order to perform a usage scenario. The usage scenarios are composed of an initial and an end point, where the work fields are some basic logical commands such as: up, down, left and right to reach the final goal. The usage scenarios can be created and modified by the professor to direct the teaching according to the needs of the class. The data coming from the scenarios resolution will be stored in a native Android database so that students' progress can be verified through in-app graphs and queries. The project was developed and tested in the classroom, with children's education students. The students were evaluated based on their understanding of the project idea, at the time of solving the usage scenarios, and in these aspects the project reached the expected objectives as the introduction of basic concepts of computer science for children.

Keywords: Educational Software. Educational Robotics. Programming for Children. Teaching Programming.

LISTA DE FIGURAS

Figura 1 – Modelo de Gowin segundo Moreira	10
Figura 2 – Modelo de Gowin utilizado em conjunto com o computador	11
Figura 3 – Comparativo Tradicional x Robótica	15
Figura 4 – Exemplo de implementação com Scratch. Programação com blocos lógicos	19
Figura 5 – LightBot Exemplo Fase do jogo	21
Figura 6 – The Foos - Demo Flash	22
Figura 7 – Codie - Robô, Blocos e Programa	24
Figura 8 – Robô Dash and Dot	25
Figura 9 – Se-Robô	26
Figura 10 – Lego MindStorm	28
Figura 11 – Bee-Bot e Cartões	29
Figura 12 – Fluxograma Aplicativo	34
Figura 13 – Diagrama da Utilização	35
Figura 14 – Modelagem da Aplicação	37
Figura 15 – Tabelas do Aplicativo	38
Figura 16 – Cenário de Ensino	39
Figura 17 – Interface de Utilização dos Alunos	40
Figura 18 – Exemplos de Cenários de Utilização	41
Figura 19 – Ilustração - Cenário caiu no Buraco	42
Figura 20 – Ilustração - Cenário Saiu para fora do cenário	43
Figura 21 – Ilustração - Cenário Quase lá	43
Figura 22 – Ilustração - Cenário Sucesso	43
Figura 23 – Comunicação Aplicativo x Robô	44
Figura 24 – Tela inicial e Cadastros	45
Figura 25 – Página Principal e Consultas	46
Figura 26 – Edição e Desenvolvimento - Cenários	47
Figura 27 – Joystick e Consultas	48
Figura 28 – Experimentos Relizados	49
Figura 29 – Primeiro e Segundo Cenários Utilizados	50
Figura 30 – Cenário com Espelhamento	51
Figura 31 – Tempos dos Alunos nos Cenários	52

LISTA DE TABELAS

Tabela 1 – Critérios de Avaliação	32
Tabela 2 – Pontuações para os Softwares Pesquisados	33
Tabela 3 – Plano de Aula 1	53
Tabela 4 – Plano de Aula 2	54
Tabela 5 – Plano de Aula 3	54

SUMÁRIO

1	INTRODUÇÃO	7
1.1	OBJETIVOS	8
1.1.1	Objetivo Geral	8
1.1.2	Objetivos Específicos	8
1.2	ORGANIZAÇÃO DO DOCUMENTO	8
2	ENSINO DE PROGRAMAÇÃO NA EDUCAÇÃO INFANTIL	10
2.1	TÓPICOS RELACIONADOS AO ENSINO DA PROGRAMAÇÃO	12
2.1.1	Benefícios e Dificuldades do Ensino de Programação	12
2.1.2	Tecnologia na Educação	14
2.1.3	Importância do Lúdico na Educação	16
2.1.4	Tecnologia da Informação	17
2.2	TRABALHOS RELACIONADOS E PRODUTOS EXISTENTES	18
2.2.1	Software Educacional	18
2.2.2	Programação com Robótica	23
2.3	CONSIDERAÇÕES FINAIS	29
3	DESENVOLVIMENTO DO SOFTWARE	34
3.1	ARQUITETURA	35
3.2	PERSISTÊNCIA DOS DADOS	37
3.3	CENÁRIOS DE UTILIZAÇÃO DOS ALUNOS	39
3.3.1	Resolução dos Cenários	42
3.4	COMUNICAÇÃO COM O ROBÔ	44
3.5	PROTÓTIPOS DE TELA	44
4	APLICAÇÃO	49
4.1	EXPERIMENTOS REALIZADOS	49
4.2	MATERIAIS E MÉTODOS	53
5	CONSIDERAÇÕES FINAIS	55
5.1	SÍNTESE	55
5.2	CONTRIBUIÇÕES ESPERADAS DO TRABALHO	56
5.3	TRABALHOS FUTUROS	56
	REFERÊNCIAS	57

1 INTRODUÇÃO

Atualmente presenciamos a era da informação e em muitos países já é realidade a inserção da ciência da computação nas grades de ensino básico das escolas (CHEN et al., 2017). Esse fenômeno é impulsionado pela indústria tecnológica, que está cada vez mais em busca de profissionais com habilidades nesta área.

Ensinar noções de Ciência da Computação, nos primeiros anos da escola, leva a criança a desenvolver habilidades como o raciocínio lógico e o pensamento computacional (GOMES; A. R. TEDESCO; MELO, 2016). O contato com o computador pode fazer com que o aluno, no futuro poça seguir na área da Tecnologia da Informação(TI), suprindo assim as necessidades do mercado que está em amplo crescimento e sofre com a falta de profissionais capacitados (SOARES, 2016).

O raciocínio lógico e o pensamento computacional são habilidade importantes e estão ligadas a racionalidade do pensamento, a resolução de problemas, e a utilização de abstrações de técnicas de ciência da computação no dia a dia (WING, 2006).

A dificuldade da inserção da lógica computacional, aliada também à dificuldade de ensinar crianças da educação infantil, faz com que seja preciso buscar alternativas eficientes para o aprendizado do conteúdo. Alternativas como a utilização da robótica educacional podem contribuir e auxiliar o professor no ensino. Com isso, o presente trabalho se insere no conceito da aprendizagem significativa, que enfatiza a importância da utilização do computador como intermediador do ensino, entre aluno e professor. A aprendizagem significativa também define a necessidade de se ter conhecimentos prévios para um aprendizado posterior (MOREIRA, 2011). A utilização da tecnologia na educação ainda pode instigar o aluno a construir seu conhecimento através de observações próprias contribuindo ainda mais para o aprendizado do aluno (MAISONNETTE, 2002).

Há alguns trabalhos precedentes a este, seguindo uma ideia semelhante de pesquisa e/ou projeto (SILVA et al., 2016; AGUIAR et al., 2015; FERREIRA et al., 2016; RICHTER et al., 2016). Nos trabalhos é possível constatar que objetivos como a implementação, a participação das crianças e até inserção da lógica computacional foram obtidos. Há também dispositivos que já estão no mercado que satisfazem a ideia da pesquisa, como o (CODIE, 2015), o (ROBO-DASH, 2015), e o trabalho de (FERREIRA et al., 2016) que traz o Se-Robô.

Dando continuidade as pesquisas e trabalhos nesta área, este estudo propõe a implementação de um aplicativo para plataforma Android. O aplicativo possui uma interface amigável, usável e de fácil compreensão para as crianças. O software se comunica com um dispositivo robótico e envia comandos básicos de movimentos espaciais, a fim de realizar os cenários de utilização. Cenários de utilização são os desafios propostos, cujo objetivo é sair do ponto inicial, cumprir todos os objetivos e assim terminar o cenário.

Por fim o software foi aplicado em uma turma de educação infantil em dois experimentos. As crianças foram avaliadas do ponto de vista da utilização do dispositivo, do engajamento

delas com o experimento, e também pelo desenvolvimento delas nos cenários de utilização.

1.1 OBJETIVOS

1.1.1 Objetivo Geral

Criar um aplicativo que promova situações simples de interação e desafio para desenvolver o raciocínio lógico de crianças na educação infantil. O aplicativo deve se comunicar com um dispositivo robótico que executará as suas instruções.

1.1.2 Objetivos Específicos

Para atingir o objetivo geral serão seguidos os seguintes objetivos específicos:

- Pesquisar os benefícios e dificuldades da introdução do ensino da lógica computacional para crianças.
- Estudar sobre a área de softwares isolados e em conjunto com robótica, para o ensino na educação infantil.
- Fazer uma análise em jogos e software de robótica educacional a fim de definir as funcionalidades do projeto.
- Especificar um software educacional para programação de um robô envolvendo problemas simples. Identificar um *layout* para a comunicação com o hardware do robô.
- Implementar um software educacional contendo interface e linguagem de programação simplificada e gráfica.
- Testar o software em uma turma de educação infantil, coletar e analisar os resultados obtidos.

1.2 ORGANIZAÇÃO DO DOCUMENTO

O trabalho está organizado em cinco capítulos. O primeiro se divide entre a introdução, e os objetivos que pretendem ser alcançados com a pesquisa. O segundo descreve sobre o ensino de programação na educação infantil, trazendo os benefícios e as dificuldades do ensino da matéria, as contribuições da utilização da tecnologia na educação, a importância do lúdico na educação, a situação do mercado da tecnologia da informação e por fim os trabalhos relacionados e produtos existentes. O terceiro capítulo define o software que foi implementado. Ele apresenta a arquitetura, o armazenamento dos dados, os cenários de utilização, a comunicação do aplicativo e os protótipos de tela. O quarto capítulo é a aplicação do software implementado.

O quinto e último capítulo é composto pela síntese final do estudo e as contribuições esperados do projeto.

2 ENSINO DE PROGRAMAÇÃO NA EDUCAÇÃO INFANTIL

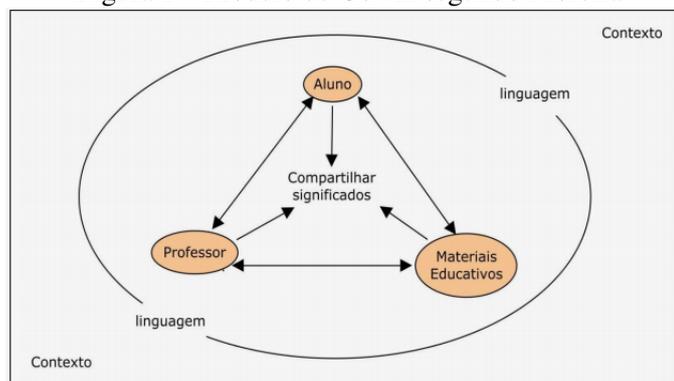
O presente estudo foi desenvolvido tendo em vista o ensino de programação na educação infantil. Portanto os temas que estão neste capítulo irão permear esse assunto. Temas como os benefícios e as dificuldade de ensino da programação e fatores que levam a isso. A utilização da tecnologia na educação. A importância do lúdico para as crianças. O mercado de trabalho da tecnologia da informação e por fim serão analisados os trabalhos e produtos que tem por objetivo inserir conceitos de ciência da computação na educação das crianças.

O ensino de programação para crianças, assim como o ensino de matemática ou quem sabe o de português, é um processo gradual. O ensino é o meio pelo qual o professor apresenta o conteúdo, para auxiliar no aprendizado do aluno. Por outro lado, para o aluno aprender de fato, somente o empenho do professor não é o suficiente. Cabe também ao aluno a responsabilidade de aprender o conteúdo que é ensinado. Este trabalho, tem como base a aprendizagem significativa. Segundo (MOREIRA, 2011):

A aprendizagem significativa é aquela em que ideias expressas simbolicamente interagem de maneira substantiva e não-arbitrária com aquilo que o aprendiz já sabe. Substantiva quer dizer não-literal, não ao pé-da-letra, e não-arbitrária significa que a interação não é com qualquer ideia prévia, mas sim com algum conhecimento especificamente relevante já existente na estrutura cognitiva do sujeito que aprende.

Em poucas palavras, a aprendizagem significativa diz que o conhecimento prévio que o aluno já possui se relaciona com novos conhecimentos que o aluno está a adquirir. E este conhecimento prévio que é mencionado é a chave para o novo conhecimento. Uma vez que novos conhecimentos são adquiridos por meio de conhecimentos prévios, o aprendizado se torna um processo iterativo.

Figura 1 – Modelo de Gowin segundo Moreira



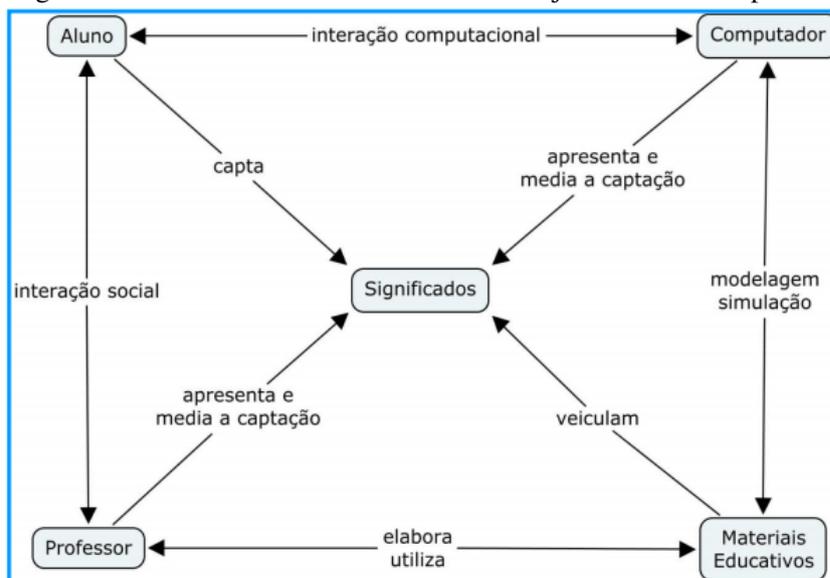
Fonte: (MOREIRA, 2011)

A aprendizagem significativa se relaciona com o modelo de ensino de Gowin. Ilustrado na figura 1, o modelo de Gowin mostra a existência de uma relação triádica, composta por aluno, materiais educativos e o professor. Essa relação precisa ser harmônica para que o aprendizado seja satisfatório. Portanto, o professor precisa ter o domínio sobre o material didático, para assim, conseguir transmitir aos alunos o conhecimento necessário para sua utilização. O material didático precisa ser útil e eficiente ao professor e ser intuitivo e agregador para os alunos. Os alunos precisam interagir e principalmente compreender tanto os materiais didáticos, quanto o professor (MOREIRA, 2011).

O resultado da interação entre as partes é o que o autor chama de significados. Significados são obtidos quando o significado captado pelo aluno sobre o conteúdo proposto, utilizando os materiais educativos, é igual ao significado que o professor deseja que o aluno obtenha utilizando aquele material.

Seguindo o modelo de Gowin, é possível adaptá-lo para o ensino da programação. Não de forma triádica como mostrado anteriormente, mas com 4 elementos. A figura 2 ilustra o funcionamento do modelo, adaptado com 4 elementos. Neste novo modelo, desenvolvido por Moreira, os 3 elementos iniciais: aluno, materiais educativos e professor se mantêm. O quarto elemento a fazer parte do modelo é o computador. O computador entenda-se como mecanismos computacionais, como tablets, smartphones, notebooks, robôs, etc. O modelo funciona da seguinte maneira: o aluno capta os significados oriundos da interação com o computador e com o professor, os materiais educativos que são elaborados pelos professores são modelados no computador veiculando significados, o professor apresenta e aplica o entendimento do conteúdo e o computador serve como mecanismo de apresentação e interação (MOREIRA, 2011).

Figura 2 – Modelo de Gowin utilizado em conjunto com o computador



Fonte: (MOREIRA, 2011)

Com base nos estudos de Moreira, é válido afirmar que o computador, ou outros mecanismos

eletrônicos como o robô, tem um papel fundamental no campo da aprendizagem significativa atuando como mediadores, auxiliando o professor, modelando e simulando situações de ensino.

2.1 TÓPICOS RELACIONADOS AO ENSINO DA PROGRAMAÇÃO

Esta seção está dividida em quatro subseções que estão organizadas nos benefícios e dificuldades do ensino de programação, a inserção da tecnologia na educação, a importância do lúdico para crianças e por fim o mercado da tecnologia da informação.

2.1.1 Benefícios e Dificuldades do Ensino de Programação

A lógica computacional segundo Forbelone é:

O uso correto das leis do pensamento, da 'ordem da razão' e de processos de raciocínio e simbolização formais na programação de computadores, objetivando a racionalidade e o desenvolvimento de técnicas que cooperem para a produção de soluções logicamente válidas e coerentes, que resolvam com qualidade os problemas que se deseja programar (FORBELLONE; EBERSPÄCHER, 2005, p.2).

Com base no que Forbelone define, é possível concluir que a lógica computacional é o pensamento aplicado à racionalidade, com o objetivo de desenvolver soluções. Essa constatação mostra a importância da lógica computacional para as pessoas. A habilidade está diretamente relacionada com o pensar de maneira produtiva.

Segundo Chaves Lima and Sousa (2015, p.3) "o raciocínio lógico é um método que contribui para que se possa realizar tarefas que estão presentes no dia a dia", e de fato podemos não perceber, mas uma série de tarefas que realizamos no cotidiano é baseada em ações lógicas ou inconscientemente pensadas e raciocinadas. Ações como, organizar uma mala para viagem são na verdade um problema do caixeiro-viajante, decidir entre ir de carro ou de ônibus pode ser encarado como algoritmo de decisão, ou refazer os passos a fim de encontrar algo perdido, o famoso backtracking, são tarefas que realizamos utilizando o raciocínio lógico.

O raciocínio lógico é um dos benefícios da inserção da computação para os alunos. Outra habilidade que pode-se desenvolver é o pensamento computacional. Não é de hoje que se escuta o termo pensamento computacional. Esse termo que é oriundo do inglês (*computational thinking*), e é descrito como uma habilidade de utilizar técnicas da Ciência da Computação, para realizar tarefas do dia a dia (BOMBASAR et al., 2015). Wing diz que:

Pensamento computacional, é uma habilidade fundamental para todos, não somente para cientistas da computação. A leitura, escrita e aritmética, deveríamos incluir pensamento computacional na habilidade analítica de todas as

crianças.[...] Pensamento computacional envolve a resolução de problemas, projeção de sistemas, e compreensão do comportamento humano, através da extração de conceitos fundamentais da ciência da computação. O pensamento computacional inclui uma série de ferramentas mentais que refletem a vastidão do campo da ciência da computação (WING, 2006, p.1).

O pensamento computacional é uma habilidade que utiliza muito da abstração do pensamento para realização de tarefas complexas ou não. É projetar e implementar um projeto ou trabalho. É modelar e representar um problema de forma a conseguir tratá-lo de maneira eficiente. É conseguir descrever as funcionalidades de uma máquina ou procedimento de maneira declarativa de forma clara. É ter autonomia e confiança para modificar uma receita ou método que já existe de maneira a melhorar a eficiência com base em certo objetivo. É utilizar modularização, a fim de dividir tarefas e organizar os problemas. Abrange também, o pensamento de prevenção e correção de erros. É planejar, aprender, pesquisar e resolver (WING, 2006).

O pensamento computacional e o raciocínio lógico são habilidades essenciais para todas as pessoas, e podem contribuir para a formação profissional principalmente na era da tecnologia em que vivemos (BLIKSTEIN, 2008). Os benefícios ligados ao pensamento computacional e o raciocínio lógico precisam estar inseridos não somente em alunos de nível superior mas deveriam ser instigados o quanto antes na escola, a fim de estimular tais habilidades nas crianças (SICA, 2011). Ainda é preciso enfatizar a importância de trabalhar tais habilidades diariamente, com o intuito de aumentar a eficácia do aprendizado (CHAVES LIMA; SOUSA, 2015).

Com base nos benefícios que podem ser adquiridos através da inserção da lógica computacional, buscaram-se alternativas para o ensino, não somente da lógica computacional, mas também de conceitos da ciência da computação para as crianças. Porém, diversas são as dificuldades encontradas para a inserção desses conceitos em sala de aula. Ressaltando três dos problemas enfrentados, estão a falta de capacitação dos professores para o ensino, a falta de planejamento de conteúdo e de ferramentas necessárias, e por fim, a falta de estruturação da escola e da grade curricular (SILVA; SOUZA; MORAIS, 2016).

Ainda com base nas dificuldades encontradas para inserção da lógica computacional, está a análise de RAABE; SILVA (2005, p.2328), onde foram levantadas dificuldades oriundas de três grupos: didáticas, cognitivas e afetivas. Didáticas, são problemas oriundos do excesso de alunos por sala de aula, diferença de ritmo de aprendizagem entre os alunos e a ausência de bons materiais. Problemas cognitivos tem relação com a falta de perfil para o aprendizado da matéria, conteúdo sem proximidade com o dia a dia do aluno, falta de estímulos exteriores do aluno. E por fim dificuldades afetivas, que tem relação com o momento pessoal dos alunos. Existem diversas dificuldades encontradas para o ensino da computação, em todos os níveis escolares. Quando o assunto é educação infantil, encontram-se ainda mais problemas, que são enfrentados não somente pelo ensino da computação, mas para o ensino de qualquer outro conteúdo.

Com o objetivo de auxiliar os professor na inserção de conceitos de ciência da computação, a associação norte americana de professores (Computer Science Teachers Association(CSTA))

fundou um currículo para a definição do ensino do raciocínio lógico e pensamento computacional para crianças. Periodicamente é lançado um livro que rege a educação da matéria. Desde os primeiros volumes do livro, foram definidos vários processos a fim de ensinar os conteúdos para as crianças. Os processos normalmente possuem como característica comum a formulação de problemas e possuem ferramentas para auxiliar a resolução das mesmas, através de passos ordenados até a solução, um processamento algorítmico. Com base no problema é preciso identificar, analisar e implementar soluções possíveis a fim de resolver o problema proposto (CSTA, 2016).

Levando em consideração os benefícios e as dificuldades encontradas para a inserção da ciência da computação na educação infantil, precisa-se buscar alternativas atrativas, principalmente quando se quer ensinar uma criança. Alternativas como a inserção da tecnologia na educação podem ser fundamentais para se obterem bons resultados.

2.1.2 Tecnologia na Educação

Além da importância da tecnologia para a aprendizagem significativa, ela pode ter um papel fundamental para inserção efetiva de conceitos da ciência da computação em sala de aula. Portanto, precisa-se buscar alternativas para a inserção de tecnologias em sala de aula. Alguns dos benefícios agregados são: a capacidade de melhorar a organização do pensamento, aumentar o interesse do aluno em sala de aula, e trazer elementos fundamentais para a construção do conhecimento (MATTEI, 2003). A introdução de meios tecnológicos pode também auxiliar no desenvolvimento de habilidades como a comunicação e a expressão, trazendo consigo inúmeros benefícios para o aluno (VALENTE, 1999). A respeito da tecnologia na educação Zilli afirma que "é uma ferramenta que permite ao professor demonstrar na prática muitos dos conceitos teóricos, às vezes de difícil compreensão"(ROCIO ZILLI, 2004, p.39).

Maisonnette ainda complementa que com a utilização da robótica educacional:

O aluno é instigado, a todo momento, a observar, abstrair e inventar, criando seus modelos a partir de materiais diversos do seu cotidiano tais como peças de brinquedos e eletrodomésticos danificados, peças de lego, circuitos eletrônicos, etc. O aluno passa, então, a construir seu conhecimento através de suas próprias observações, e como nos ensina Papert, aquilo que é aprendido pelo esforço próprio da criança tem muito mais significado para ela e se adapta melhor às suas estruturas mentais, uma vez que o processo de aprendizagem requer, para as informações novas, uma estrutura anterior que permita que estas possam ser melhor assimiladas(MAISONNETTE, 2002, p.1).

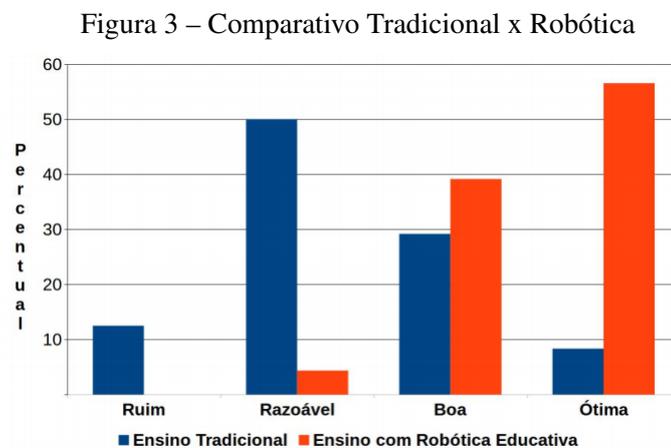
A introdução da robótica na educação infantil tem mais um papel importante que é o de inserir o material concreto. O material concreto auxilia no desenvolvimento de habilidades que requerem abstrações, como o pensamento computacional e o raciocínio lógico. Com isso as crianças imaginam situações, narram histórias sobre o robô, formulam hipóteses, ou seja,

aprendem brincando. Através da visualização da ação acontecendo, as crianças conseguem elaborar as ações pretendidas, avaliando o objetivo a ser alcançado, ou seja, estão raciocinando durante os movimentos do robô. É desenvolvida assim uma capacidade analítica de observação de fenômenos e também é desenvolvido o gosto pelo aprendizado (PERALTA, 2015).

O ensino da programação aliada ao uso de tablets, smartphones e computadores, tem seus benefícios multiplicados, aumentando ainda mais a importância de seu ensino. As tecnologias estão no mercado com o objetivo de facilitar e contribuir para o ensino, porém sua utilização em sala de aula ainda é restrita a instituições com mais recursos, sendo privilégio de poucos. Os principais obstáculos a serem ultrapassados para a inserção da tecnologia em sala de aula são a dificuldade do acesso às ferramentas como computadores, tablets, celulares e kits de robótica e a falta de capacitação dos profissionais ligados à área da educação (ABELLÓN, 2015).

A estrutura das escolas brasileira em sua maioria não possui nem o básico para a educação. A obtenção da tecnologia necessária para amplo aproveitamento dos estudantes no país ainda é um sonho. Devido a isso, é necessário buscar alternativas para a inserção da tecnologia por um custo baixo. Outro fator relevante é que professores de pedagogia e licenciatura em sua maioria possuem pouco ou nenhum referencial de utilização da tecnologia na educação em seus cursos (ABELLÓN, 2015).

Abstraindo as dificuldades e pensando pelo lado de gosto, Cambruzzi desenvolveu um trabalho de pesquisa em que tinha por objetivo comparar o ensino de lógica computacional utilizando a robótica, e os meios convencionais de ensino. O autor elaborou o gráfico da figura 3 onde é mostrado um dos resultados da pesquisa. O gráfico ilustra a avaliação do aluno sobre a metodologia adotada e mostra a preferência dos alunos pela utilização da robótica no ensino.



Fonte: (CAMBRUZZI; SOUZA, 2015)

2.1.3 Importância do Lúdico na Educação

Deve ser levada em consideração a questão de que este projeto de pesquisa é voltado para crianças, mais especificamente para crianças da educação infantil (4 a 6 anos de idade). Sabe-se que as crianças desde muito pequenas já brincam. Atualmente, as leis dão suporte ao direito da criança em estar inserida na educação infantil, e também regem os direitos de uma educação para a criança. As Diretrizes Curriculares Nacionais para a educação infantil de 2010, definem que a:

Criança é centro do planejamento curricular, sujeito histórico e de direitos que, nas interações, relações e práticas cotidianas que vivencia, constrói sua identidade pessoal e coletiva, brinca, imagina, fantasia, deseja, aprende, observa, experimenta, narra, questiona e constrói sentidos sobre a natureza e a sociedade, produzindo cultura (MINISTÉRIO DA EDUCAÇÃO, 2010, p.12).

Desde seu nascimento, a criança tem o contato com essas atividades e instrumentos que possibilitam a ela o prazer, a descoberta, a troca, a interação com o professor e colegas, ela aprende a agir, se posicionar, cria uma autoconfiança, desenvolve a linguagem, habilidades, capacidades, estimulando a curiosidade, a concentração e o pensamento. Segundo SANTOS (1999) essas atividades descritas acima, são atividades lúdicas, de extrema importância em qualquer idade ainda mais na infância, porém elas não podem apenas ser vistas como diversão. Com a ludicidade, através de jogos, a criança tem maior facilidade em sua aprendizagem, no desenvolvimento pessoal, social e cultural, de forma a facilitar os processos de socialização, comunicação e construção do conhecimento. Portanto, a atividade lúdica enquanto uma necessidade humana faz de todas as etapas da vida, sendo priorizada na infância para favorecer o desenvolvimento integral de habilidades e competências da criança.

O brincar pode proporcionar maiores possibilidades para a criança compreender o mundo físico, organizar e reorganizar seus processos psíquicos e interpretar os papéis sociais que testemunha em seu meio (VYGOTSKY, 1984). Enquanto brinca, a criança vai desenvolvendo funções psíquicas e atitudes que são condição para seu desenvolvimento futuro na escola e na vida:

[...] tudo aparece no brinquedo que se constitui, assim, no mais alto nível de desenvolvimento pré-escolar. A criança desenvolve-se através da atividade de brinquedo. Somente neste sentido o brinquedo pode ser considerado uma atividade condutora que determina o desenvolvimento da criança. (VYGOTSKY, 1984, p.135).

Por isso, é importante que se utilize a ludicidade na educação infantil, pois é a partir da pré-escola que as crianças estabelecem relações, e através de brincadeiras e brinquedos, que

desenvolvem seu raciocínio lógico de forma mais facilitada. Desta forma, estar aprendendo se torna algo prazeroso para as crianças. Segundo MUKHINA (1996) as atividades lúdicas desenvolvem as qualidades das crianças de forma intensa. Desta forma, desenvolve a atenção e memória das crianças. As brincadeiras ou os brinquedos, influenciam a atividade mental na pré-escola, sendo as atividades lúdicas elemento fundamental para que as crianças possam compreender, e substituir por exemplo certos objetos por outros e a interpretar diversos papéis, o que desenvolve sua imaginação. As condições de jogo obrigam a criança a concentrar-se nos objetos na situação lúdica, no conteúdo das ações e no argumento que interpreta.

Portanto, a utilização de softwares e robôs que introduzem consigo a ludicidade e a desenvolvem utilizando jogos com o objetivo de ensinar, contribuem ainda mais para o ensino e aprendizagem das crianças.

2.1.4 Tecnologia da Informação

A área da Tecnologia da Informação (TI) está em amplo crescimento e está em contrapartida do restante das áreas no mercado de trabalho. O problema que o TI enfrenta é a falta de profissionais capacitados, para suprir a demanda crescente de serviços. Estima-se que em 2020 o Brasil pode chegar a um déficit de até 408 mil de profissionais na área de TI (SOARES, 2016).

Segundo a organização Code.org é esperado um acréscimo de 1,4 milhões de vagas em todo o mundo no mercado de TI, e apenas 400 mil delas serão preenchidas. Devido a este fato, não é por menos que, multiplicam-se os cursos de graduação e especialização nesta área. Portanto, a grande falta de alunos para suprir estas vagas é um obstáculo a ser ultrapassado. Desta forma, a inserção da tecnologia da informação na educação infantil pode aumentar o número de profissionais da área no futuro (CODE.ORG, 2016).

(SILVA; LOPES; RODRIGUES, 2014) conta que as causas para a falta de profissionais capacitados no TI, está ligado desde a falta de interesse dos estudantes brasileiros, por carreiras ligadas às áreas exatas, até a alta evasão dos cursos da tecnologia. Ele também cita um dos problemas relacionados a falta de interesse, como o fato de os alunos apresentarem dificuldades em disciplinas que abrangem conteúdos de raciocínio lógico-abstrato, algoritmos e programação. Devido a este aspecto, inserir conceitos de ciência da computação na educação infantil pode também aumentar o número de interessados por cursos na área do TI (NUNES, 2011).

Percebendo os benefícios da estimulação em crianças com conteúdos de ciência da computação, os Estados Unidos e países da Europa têm implantado um currículo mínimo de Computação em suas escolas (FRANÇA; AMARAL, 2013).

O mercado de trabalho pode ser mais um atrativo para a inserção da computação na educação básica. Com um número maior de crianças utilizando conceitos de ciência da computação é possível aumentar o número de estudantes da área do TI em universidades e com isso suprir o mercado.

2.2 TRABALHOS RELACIONADOS E PRODUTOS EXISTENTES

O presente trabalho é precedido por trabalhos e produtos que buscam objetivos semelhantes ou possuem proposta de desenvolvimento parecidas. Esta subseção irá abordar os métodos que foram utilizados anteriormente, com o objetivo de ilustrar as possibilidades que já foram abordadas e com isso inspirar e auxiliar o desenvolvimento da proposta do software. A seção está organizada em Softwares Educacionais e Programação com Robótica.

2.2.1 Software Educacional

O software educacional irá ser o intermediador e o auxiliar do professor, para transmitir o conteúdo que será proposto em aula. A escolha de um software educacional é fundamental para um bom aprendizado dos alunos. Softwares educacionais são distintos em suas funcionalidades, interfaces, complexidade, preço entre outras. Cabe ao professor identificar qual o software será mais adequado para o conteúdo que será proposto em aula.

Esta seção tem como objetivo descrever e identificar os softwares, para o ensino de programação para crianças. Para tal, foram selecionados softwares de nível internacional e nacional. Como critério de seleção, foram escolhidas as ferramentas que possuem um produto no mercado, ou que fazem parte de um projeto de pesquisa de universidades, como monografias, artigos e projetos.

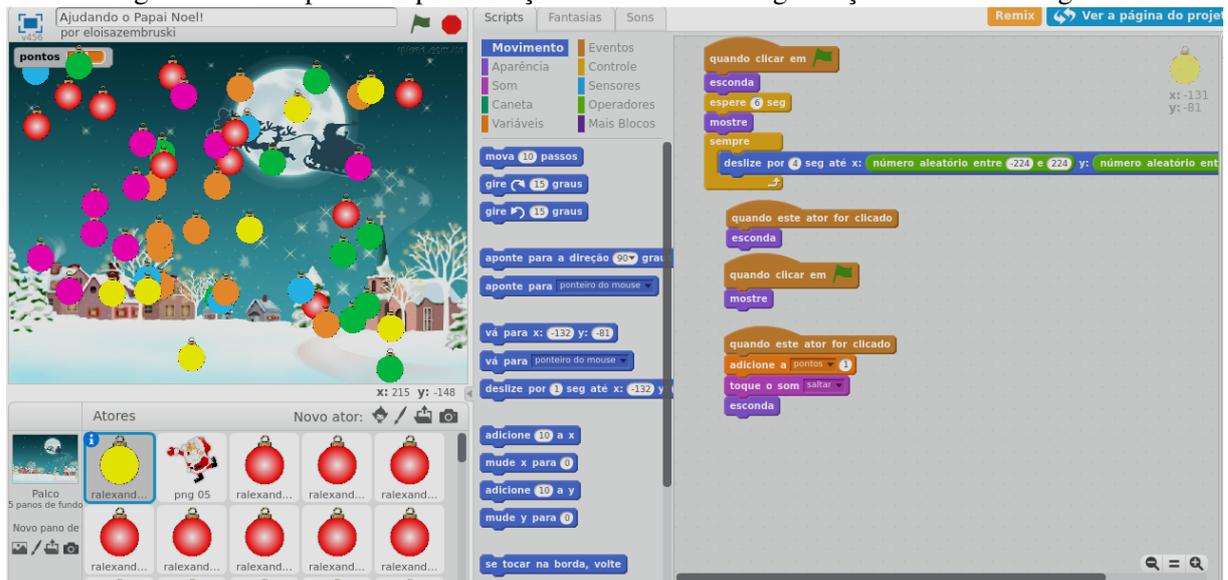
2.2.1.1 Scratch

Scratch é uma ferramenta online que possui versões offline para download. Oriunda dos Estados Unidos mais precisamente do Instituto de Tecnologia de Massachusetts (MIT). Foi criada pelo Prof. Dr. Mitchel Resnick em 2009. O software tem como objetivo ensinar conceitos de programação para crianças de 8 a 16 anos. Com a utilização de blocos lógicos, sons e imagens para o desenvolvimento de histórias interativas, animações e jogos. O *Scratch* é totalmente grátis o que é um ponto favorável a sua utilização (SCRATCH, 2007).

O software possui uma linguagem própria. É uma das principais ferramentas no mercado para o ensino de programação para iniciantes. A forma como os blocos podem ser manipulados, lhe confere uma possibilidade de aprendizagem autogerida através da prática de manipulação e teste dos projetos. A ferramenta conta com blocos que representam ações, como laços de repetições, expressões condicionais, interações com o jogo em si, comportamentos de animações, customização de cores e muito mais. É possível também utilizar pacotes externos para conectar a ferramenta com hardwares.

O *Scratch* é utilizado para o desenvolvimento de jogos, histórias e animações. O software é muito mais acessível que outras linguagens de programação, por se utilizar de uma interface gráfica que permite que programas sejam montados como blocos de montar, lembrando o

Figura 4 – Exemplo de implementação com Scratch. Programação com blocos lógicos



Fonte: (SCRATCH, 2007)

brinquedo Lego. Utiliza uma sintaxe comum a muitas linguagens de programação.

O *Scratch* é uma ferramenta de criação. Possui duas fases, a primeira é a fase de execução, onde a criação do aluno é apresentada. A segunda é a fase de criação do projeto. A interface do software para a criação é dividida em duas grandes partes. Na parte esquerda em cima da figura, fica a tela do usuário e logo abaixo encontram-se os atores. A Figura 4 ilustra esta parte da interface. Os atores são os personagens do projeto. Ali mesmo é possível fazer o upload de imagens, para serem usadas como atores do projeto.

A programação em *Scratch* é baseada em arrastar componentes para área de desenvolvimento do projeto. Na parte da direita da interface, é onde isso é realizado. A Figura 4 também ilustra a interface do lado direito. Acima é possível identificar 3 opções. Os Scripts são blocos de variados tipos para desenvolver o programa do aluno. Existem scripts de movimento, aparência, variáveis, sensores entre outros. Os scripts são baseados em atores, logo para escolher um script um ator é associado a ele. A aba fantasia é utilizada para trocar cores ou adicionar efeitos nos atores. E a aba de sons é onde é possível fazer o upload de sons.

A figura 4 em questão mostra um programa que basicamente realiza o seguinte programa: Quando for clicado no ator "bandeira". Esconder o ator, esperar 6 segundos, mostrar os outros atores do jogo, em um laço infinito movimentá-los pela tela em coordenadas aleatórias. Quando os atores forem clicados, escondê-los e somar 1 ponto a variável que conta os pontos. Os blocos que não estão ligados, fazem parte de outra sequência lógica de execução.

O *Scratch* é orientado para crianças com idade superior a 8 anos. Possui uma alta complexidade, uma interface bastante poluída, pouca abstração de conteúdo, e se faz necessário que o aluno seja alfabetizado. De outro lado é um software completo, que possui diversas opções de composições lógicas, é grátis e é possível desenvolver muita coisa com ele.

2.2.1.2 LightBot

LightBot é um aplicativo para dispositivos móveis, que conta com uma versão em flash para rodar em computadores. Desenvolvido por Danny Yaroslavski em 2014 no Canadá. O aplicativo é um jogo que conta com um personagem principal, um simpático robzinho, que precisa solucionar vários quebra-cabeças, através de comandos dados ao robô através das opções que são selecionadas em um menu (LIGHTBOT, 2008).

A ideia desse aplicativo é usar a lógica em um jogo baseado em fases. O *LightBot* ensina os conceitos básicos da programação para as crianças de maneira divertida, através de atividades lúdicas. O jogo conta com cerca de 50 fases, e também vários níveis de dificuldade. Conta também com suporte a várias linguagens incluindo o português. O jogo possui uma interface simples e de fácil compreensão, e é composto por um menu que fica na parte inferior da tela do jogo. Ele contém as opções que são possíveis para a fase atual. É preciso arrastar a opção desejada para a área de programação ao lado para montar a lógica que o robô irá realizar. Após o término do programa, é preciso apertar o botão de início e então o robzinho executa o programa que foi desenvolvido.

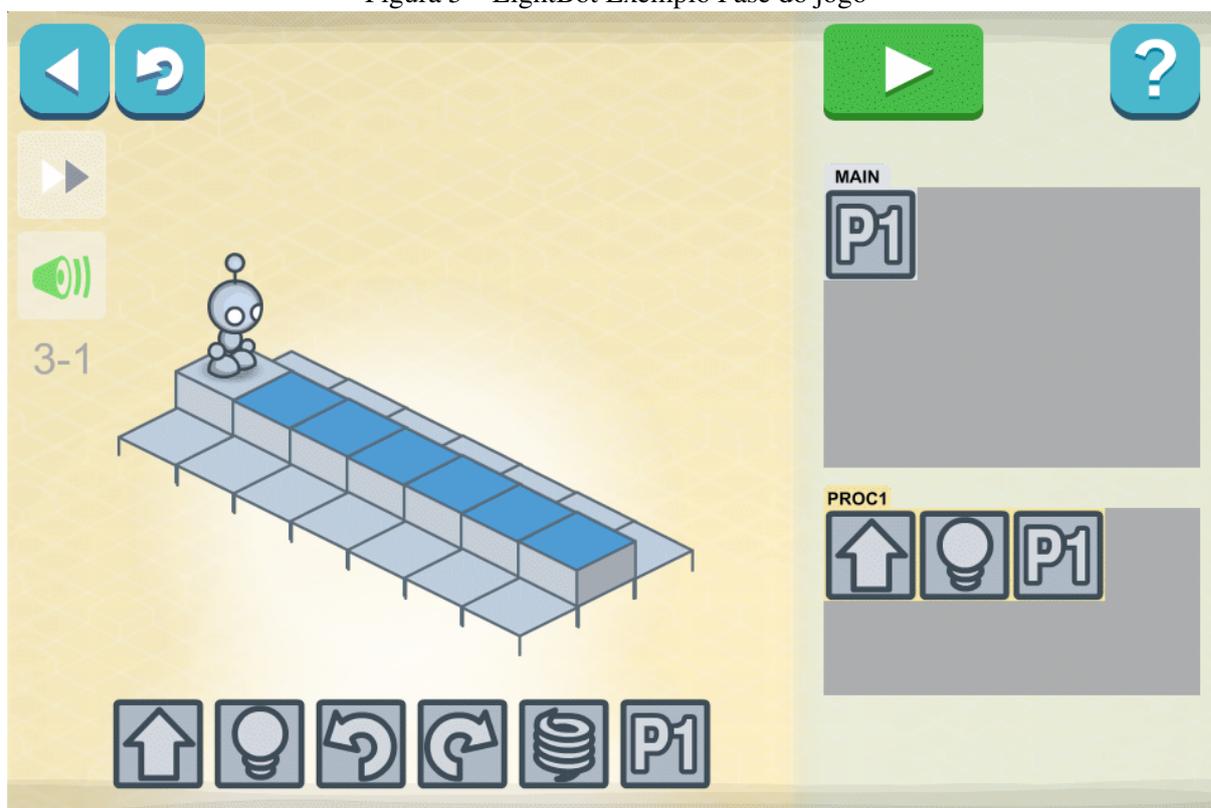
O aplicativo é interessante, pois jogos são bem vistos pelas crianças, segundo Fialho os jogos educacionais, "promovem situações de ensino e aprendizagem e aumentam a construção do conhecimento, introduzindo atividades lúdicas e prazerosas, desenvolvendo a capacidade de iniciação e ação ativa e motivadora"(FIALHO, 2008, p.2). Outro fator interessante do jogo é que precisamos montar a lógica do robô antes de iniciar o jogo, ou seja, precisamos montar o quebra-cabeça e solucioná-lo utilizando a abstração, o que ajuda ainda mais no desenvolvimento do raciocínio, mas pode aumentar em demasia a dificuldade do aplicativo.

O jogo conta com duas versões: uma júnior para crianças de 4 a 8 anos; e a outra recomendada para crianças com 9 ou mais. O que muda de uma versão para outra, é a dificuldade dos quebra-cabeças do jogo e também o preço. O preço varia de 6,99 R\$ a 11,99 R\$ para as versões júnior e normal respectivamente. Existe também uma versão de demonstração em flash do jogo que conta com cerca de 20 fases. O jogo possui traduções para várias linguagens, inclusive o português. Existem tutorias no próprio jogo, o que torna fácil a mecânica do jogo. Não é necessário que a criança saiba ler, para jogar o jogo, uma vez que os comandos do jogo são figuras que ilustram o movimento que ela realiza.

O robô pode fazer alguns comandos pré-estabelecidos de acordo com a fase em que o jogador está. Basicamente as opções são movimentos tanto para os lados, frente e traz, pulos, luzes do robô. O jogo traz outro conceito interessante da programação que é chamada de procedimentos.

A figura 5 exemplifica uma fase do jogo em flash. A esquerda vemos o robô e a fase que foi escolhida. O objetivo do robô é chegar até o fim, através do caminho em azul. A direita tem a parte intitulada "main", que é onde os comandos para o programa precisam ser arrastados. Logo abaixo do main existe a proc1 que é o procedimento que pode ser utilizado nesta fase.

Figura 5 – LightBot Exemplo Fase do jogo



Fonte: (LIGHTBOT, 2008)

Nesta fase, se faz necessário a utilização da chamada de procedimentos formando um loop. Para o procedimento 1 foi selecionado três comandos. Para frente, ligar a luz, e uma chamada recursiva para o próprio procedimento 1. E para o main foi passado o procedimento 1. Com isso após o programa ser executado, o robô irá executar até o término do percurso. Assim que o robô termina o percurso, a fase acaba.

Os comandos do aplicativo são simples e em poucos minutos o aluno já tem total conhecimento dos comandos do jogo. Após o período de aprendizado da ferramenta, a dificuldade está em resolver os quebra-cabeças das fases.

2.2.1.3 The Foos

The Foos é um aplicativo desenvolvido pela CodeSpark, que é uma escola de programação para crianças de Los Angeles Estados Unidos. Possui versões para IOS e Android, e uma versão de demonstração em flash na web. É uma plataforma completa com perfis de usuário e conta também com recursos gráficos bem produzidos. Seu público alvo são crianças de 4 a 10 anos (CODESPARK, 2008).

O jogo faz parte do currículo da escola para o ensino de programação. Possui disponível em pdf planos de aula, onde o *The Foos* é usado para o ensino de programação. Para ter acesso a

todo o conteúdo do jogo, é necessário que o professor assine o jogo. O valor é 7,99 US\$ dólares mensais.

O game tem como principal objetivo quebra-cabeças que precisam ser solucionados utilizando comandos que são selecionados abaixo do personagem, semelhante ao *LightBot*, porém conta também com diversos outros jogos dentro da plataforma. O diferencial do jogo é o fato de ser mais encorpado, possuindo perfis e customizações para meninos e meninas, um mini mundo onde as fases do game são selecionadas e vários outros detalhes visuais e animações.

The Foos introduz na criança um mundo novo. *Foos* é o nome da raça das criaturas do jogo, cada Foo pode andar, pular, jogar, comer e navegar em seu mundo, chamado “Fooville”. Alguns Foos têm habilidades especiais que os tornam únicos, por exemplo: policial, ninja, chef, astronauta, construtor entre outros. Esses *Foos* únicos fazem parte de subjogos especiais. Por exemplo, subjogo de fazer comida, onde o chef é o *Foo* utilizado. Subjogo de capturar bandidos, onde o policial é utilizado.

O jogo foi desenvolvido por uma escola de programação, portanto ele é um instrumento da escola para o ensino da programação. Além do jogo, a escola utiliza os Foos para diversas outras atividades envolvendo a programação, mas sem auxílio do jogo.

Assim como o *LightBot*, o jogo é intuitivo, portanto, o aluno não precisa saber ler para entender o jogo. A interface do jogo não é clara sobre o objetivo assim, uma criança teria dificuldades de jogar sem algum auxílio. Os comandos e o jogo são animados, o que torna o jogo atrativo. O jogo conta com *activements* que são pequenas tarefas no jogo. Possui dinheiro, que é usado para comprar novos Foos e também customizações para eles. O dinheiro do jogo é obtido ao passar pelas fases, conseguindo 1, 2 ou 3 estrelas. As estrelas são obtidas, dependendo da sequência de comandos que o aluno utilizou.

Figura 6 – The Foos - Demo Flash



A figura 6 mostra a 4ª fase do jogo para versão de demonstração. O objetivo desta fase é fazer com que o *Foo* chegue até a rosquinha. Podemos observar que existem caixas que impedem que o *Foo* chegue até lá, indo em linha reta. A solução para este problema é um movimento para frente e dois pulos.

2.2.2 Programação com Robótica

Aliado aos softwares para ensino de programação, há também um mercado ainda mais restrito, principalmente no Brasil. Os robôs para ensino de programação para crianças, ainda são pouco difundidos. Existem no mercado algumas opções mas, o seu custo elevado e o fato da produção não ser desenvolvida no país, dificulta ainda mais a obtenção dos produtos. O robô ajuda e muito o entendimento da programação para crianças, por estar inserido em conceitos como o material concreto e o lúdico na educação.

Esta seção tem como objetivo descrever e identificar os softwares ou aplicativos, aliados a robótica, que tenham como objetivo ensinar conceitos de ciência da computação para crianças. Como critério de seleção, dos produtos pesquisados, foram escolhidas as ferramentas que possuem um produto no mercado, ou que fazem parte de um projeto de pesquisa de universidades, como monografias, artigos e projetos.

2.2.2.1 Codie

O projeto *Codie* teve início em 2013 como uma startup, seus criadores Adam e Andras participaram da Microsoft Imagine Cup onde idealizaram o projeto. O *Codie* venceu a etapa nacional e então eles perceberam que o projeto tinha futuro e resolveram entrar no mercado com o produto. O valor do *Codie* é de cerca de 150US\$, mais o custo com o transporte e frete do mesmo, o que torna o produto caro para os padrões brasileiros (CODIE, 2015).

Codie é um robô de brinquedo, que ajuda a ensinar os princípios da programação de computadores, e ainda ajuda a entender como a tecnologia funciona. Funciona aliado a um aplicativo para celular, que se comunica via *bluetooth* com o robô, podendo acessar seus sensores. Desta forma, é possível desenvolver programas para seu *Codie*. É possível criar programas como um despertador, utilizando o sensor de luminosidade. Utilizando loops é possível fazer o *Codie* dançar, e muitas outras coisas.

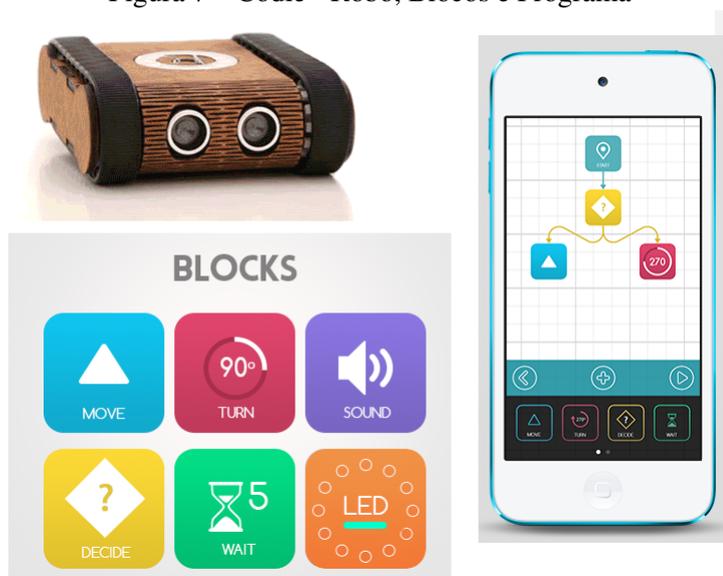
Para montagem do programa, é necessário arrastar os blocos para cima e após o programa montado é só rodar. Os blocos são compostos por movimentos, giros, sons, condicionais, espera, e os leds do robô. Logo, é possível montar programas bastante elaborados com o *Codie*. Cabe ao professor definir quais serão os algoritmos que serão passados às crianças, ou orientá-las e deixá-las usarem a imaginação.

O robô é desenvolvido com base em *Arduino*. Utiliza uma bateria e tem autonomia de 4h de duração. O robô possui microfone e luzes de led que podem ser acionadas via comando. A

comunicação com o robô, é feita via *bluetooth*, através de um aplicativo. O aplicativo pode ser baixado tanto para Android como para IOS. O robô apenas pode ser controlado pelo aplicativo e o aplicativo somente controla o robô.

O funcionamento do aplicativo é simples. Na parte superior da interface, existe a área onde serão arrastados os blocos referentes aos comandos que desejam ser acionados. É possível agrupar alguns comandos, formando um programa. Você tem a opção de salvar a sequência de códigos para carregar ele no futuro.

Figura 7 – Codie - Robô, Blocos e Programa



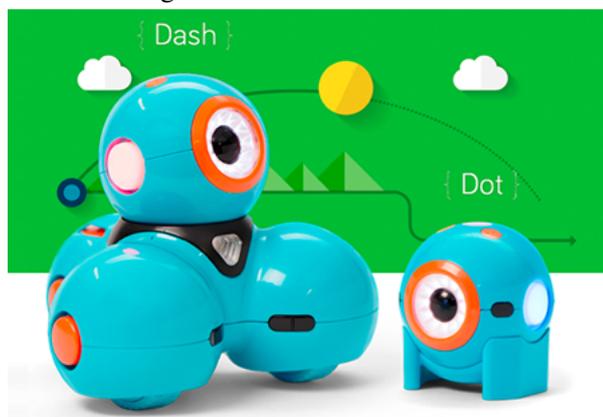
Fonte: (CODIE, 2015)

2.2.2.2 Robô Dash

O robô é desenvolvido pela empresa *Wonder Workshop*, que está localizada nos Estados Unidos. O robô é projetado para o ensino de programação. Ele custa 150US\$, sem contar o custo para envio do produto. Dash é projetado para o ensino de programação para crianças a partir dos 6 anos de idade. O robô é controlado por um aplicativo móvel, via *bluetooth*. Existem cinco aplicativos disponíveis para o controle do Robô Dash, que são: *Wonder*, *Path*, *GO*, *Blockly*, *Xylo*. Cada aplicativo tem características diferentes e cada usuário pode utilizar qualquer um deles (ROBODASH, 2015).

Wonder, é um aplicativo que possui como característica, ter alguns algoritmos agrupados em únicos comandos. Com ele é possível fazer o Dash dançar e acender luzes em dois comandos. Ele funciona arrastando os comandos da parte inferior da interface, até a parte superior. Os códigos são agrupados formando programas. É possível salvar os código e compartilhar os mesmos na nuvem do próprio aplicativo. Também é possível visualizar códigos de outros usuários.

Figura 8 – Robô Dash and Dot



Fonte: (ROBODASH, 2015)

GO é o aplicativo para personalização do Dash. Nele é possível configurar e personalizar o robô. No aplicativo podemos escolher a cor dos leds do robô, e é ali que se encontram tutorias de como controlar o Dash. O GO também é necessário para a criação do perfil online do usuário.

Path, programa o Dash para seguir um caminho traçado em lugares fictícios. Nele temos vários mapas onde o Dash é inserido, portanto, é utilizado aliado a imaginação do aluno. O aplicativo cria um cenário onde podemos tentar reproduzir ele no mundo real. Possui também acesso a dispositivos como sons. Por exemplo, podemos imaginar que o Dash está na fazenda, e imitar sons de animais.

Blockly é um aplicativo semelhante ao Scratch. Nele é possível elaborar algoritmos e programas complexos. No Blockly temos um controle total sobre o Dash. Podemos fazer condicionais, loops, e tem-se acessos a todos os sensores do robô. Para programar com o aplicativo, é preciso arrastar os componentes do menu que fica a esquerda para a área de desenvolvimento que fica a direita. É o aplicativo mais completo para o Dash. Existem também quebra-cabeças que podem ser encarados como desafios aos alunos. O Blockly é bem semelhante ao Scratch.

Xylo é um aplicativo, aliado a uma mesa de som, que fica acoplada ao Dash. Isso faz com que o robô consiga reproduzir notas musicais feitas pelo aplicativo. No aplicativo, o aluno escreve as notas e manda o Dash executar a composição.

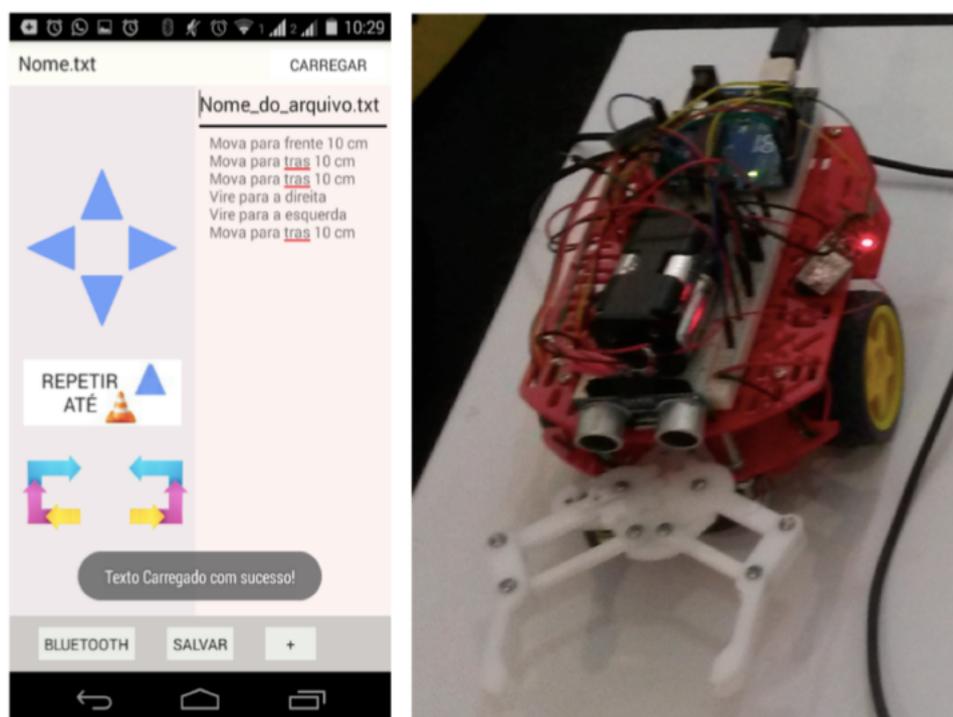
Assim como o The Foos, o Dash é oriundo de uma escola de programação, portanto, existem diversos materiais e planos de aula para orientação de educadores que pretendam utilizar ele em sala de aula. A Happy Code é uma escola brasileira que já está em várias cidades do Brasil, e traz como conteúdo, aulas de programação com a utilização do Robô Dash. O Dash não está sozinho, existe outro robô que é utilizado com ele nas aulas. O Dot é um robô semelhante ao Dash, mas conta com um número reduzido de sensores. Alguns dos aplicativos também funcionam para o Dot, mas com funcionalidades diferentes.

2.2.2.3 Se-Robô

Em alternativa aos produtos do exterior não existem aplicativos brasileiros que são vendidos como produtos assim como os mostrados anteriormente. Existem projetos que estão sendo desenvolvidos em busca de um produto que possa de fato ser utilizado no padrão escolar brasileiro. O maior fator que leva o mercado brasileiro a produzir esse tipo de produto é o preço elevado com que é importado.

"O Se-Robô se destina principalmente para crianças de 6 a 12 anos de idade e é composto por um aplicativo para dispositivos móveis com sistema Android e um robô móvel de baixo custo desenvolvido na plataforma aberta de hardware *Arduino*. Através do Se-Robô o estudante poderá programar um robô móvel de maneira intuitiva e dinâmica, vendo o resultado de seus comandos em tempo real. Os experimentos iniciais realizados, têm mostrado resultados promissores em relação ao Se-Robô"(FERREIRA et al., 2016, p.1).

Figura 9 – Se-Robô



Fonte: (FERREIRA et al., 2016)

O projeto conta com uma interface para controle do robô, que pode fazer movimentos básicos e loops. A interface é um aplicativo Android que manda mensagens gravadas em arquivos via *bluetooth*. O robô recebe as mensagens e as interpreta.

O aplicativo foi desenvolvido para Android e conta com uma interface simples com alguns comandos, dentre estes comandos se encontram: direcionais, frente, esquerda, direita, atrás;

repita até encontrar um obstáculo; desvie do obstáculo; e um controle de abrir e fechar a garra. Na parte direita da interface, o usuário poderá visualizar a sequência de comandos informados. Esta sequência, poderá ser salva como um arquivo de código e ser carregada novamente. Os comandos são executados em tempo real, para tornar a programação do robô através do aplicativo, mais dinâmica e manter a atenção do jovem usuário.

O robô foi desenvolvido na plataforma *Arduino*, por ser uma plataforma aberta e de baixo custo. Para a confecção do robô, foram utilizadas peças e dispositivos de baixo custo em toda sua montagem. O robô é composto basicamente por: um controlador *Arduino*, dois motores do tipo DC para locomoção; uma ponte H, para controle dos motores DC; uma garra de acrílico, um micro-servo, para controle da garra; um sensor do tipo ultrassônico para desvio de obstáculos e um módulo *bluetooth* para comunicação com o dispositivo móvel. Ao total, foram gastos cerca de 300R\$ na montagem do robô (FERREIRA et al., 2016).

Para a aplicação do projeto em sala de aula, existiriam diversas possibilidades. Sob a orientação do professor, os alunos poderiam desenvolver programas a fim de fazer com que o robô possa atravessar obstáculos, utilizando os recursos disponíveis pelo robô. Segundo (FERREIRA et al., 2016) após isso, os estudantes poderiam medir os movimentos do robô de acordo com a sequência de comandos visualizada no aplicativo. Esta atividade pode estimular o raciocínio lógico e a noção de espaço, forma e geometria nas aulas de matemática.

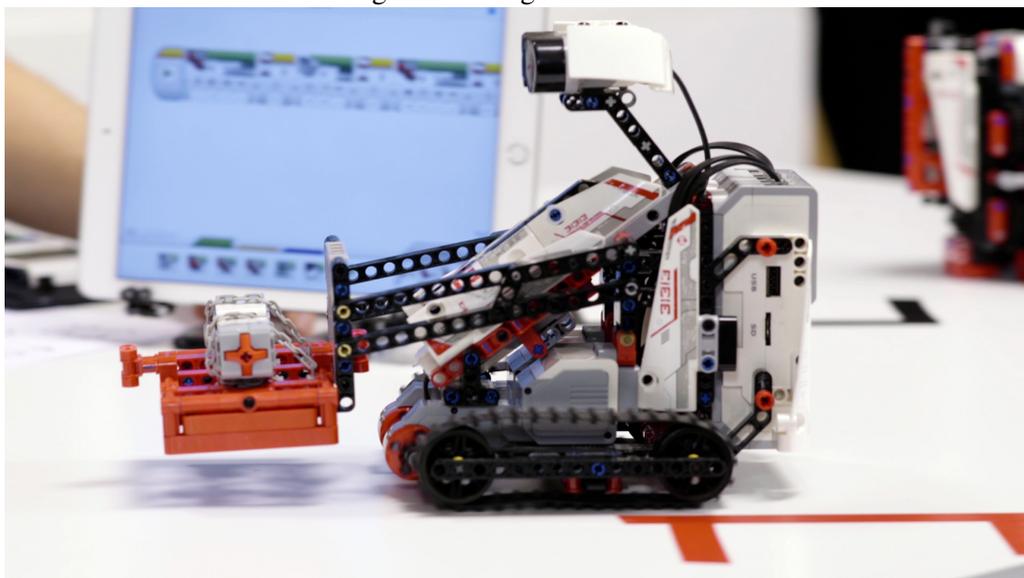
2.2.2.4 Lego Mindstorms

Silva e Aguiar utilizaram em seus trabalhos o kit Lego Mindstorms da LEGO®. O kit é composto por sensores para interação com o mundo real, um controlador que é utilizado para programação do kit, três motores, peças e componentes para montagem do robô. Silva compara a utilização do kit LEGO® com a utilização do *Arduino* como alternativa, e segundo sua análise pela falta de conhecimento em eletrônica é mais conveniente a utilização do kit LEGO® (SILVA et al., 2016). Por outro lado, devido ao preço do produto, que é de cerca de 1600R\$, sua utilização efetiva fica comprometida, para não concluir inviável. O kit é vendido pela LEGO® e sua última versão é a 31313.

O produto vem com um software para programação do projeto desenvolvido. Segundo Aguiar o software "possui uma interface projetada para ser dinâmica e atrativa, utiliza blocos de comando coloridos que se conectam uns aos outros pelo simples arrastar do mouse na tela do computador" (AGUIAR et al., 2015, p.1419). Ainda segundo Aguiar, com o kit é possível a criação de diversos projetos como robôs, naves, carros e o que a imaginação permitir.

Silva tinha como objetivo, estimular o pensamento computacional nas crianças. O experimento do projeto foi realizado com crianças de 11 a 13 anos. O autor observou a importância de introduzir os conceitos teóricos sobre robótica e programação antes de realizar os experimentos com o kit de robótica. Foi constatado também que a aparência do robô é também um ponto a ser pensado, pois precisa agradar aos alunos (SILVA et al., 2016).

Figura 10 – Lego MindStorm



Fonte: (LEGO, 2017)

Aguiar, propunha a introdução à robótica e estímulo à lógica de programação, seu trabalho foi experimentado por alunos mais velhos de 14 e 15 anos. Por esse motivo também ele partiu do ponto de introduzir a robótica, ensinando os alunos a montarem o robô e posteriormente utilizando a programação para elaborar desafios aos alunos (AGUIAR et al., 2015).

Em ambos os trabalhos foram obtidos resultados satisfatórios da utilização do produto para os objetivos propostos. Porém a experimentação do kit de robótica LEGO® foi em alunos de uma faixa etária elevada. O fabricante orienta sua utilização para alunos maiores de 10 anos.

A interface de programação do kit é confusa e sua utilização até mesmo por alunos de 10 a 15 anos já é complicada, a figura 10, ilustra a interface de programação ao fundo do robô.

2.2.2.5 Bee-Bot

O trabalho de Richter utilizou o brinquedo Bee-Bot®. O robô é produzido pela empresa Terrapin, com sede em Massachusetts Estados Unidos. Ele é uma abelhinha, que pode se mover e girar e possui indicação de uso à crianças de 4 e 5 anos. O robzinho acompanha também cartões identificados com os comandos possíveis, para realização dos programas anteriormente a programação e execução pelo robô (BEE-BOT, 2016).

O experimento do trabalho de Richter foi realizado com alunos do jardim II que são crianças de 4 e 5 anos. O seu projeto foi elaborado em três partes. Na primeira as crianças foram orientadas a descobrir sozinhas os movimentos do robô, sem a interferência dos pesquisadores. Na segunda foram mostrados os cartões de programação, que servem de auxílio a abstração dos movimentos. Por último foi realizado uma competição com as crianças com desafios propostos (RICHTER et al., 2016).

Figura 11 – Bee-Bot e Cartões



Fonte: (BEE-BOT, 2016)

Na primeira etapa foi constatado que as crianças tiveram dificuldades em desvendar os movimentos do robô sem auxílio externo. Conseguiram movimentar o robô de forma randômica. Notou-se também um certo afeto das crianças pelo robô (RICHTER et al., 2016).

Mesmo após a explicação dos pesquisadores do funcionamento do robzinho, as crianças tiveram dificuldade de entender do que se tratava. Notou-se também durante o experimento que algumas crianças simplesmente copiavam os códigos das outras, e outras faziam qualquer programação, sem de fato pensar a respeito da solução (RICHTER et al., 2016).

Observando a composição do Bee-Bot® e o trabalho de Richter é possível observar o nível de dificuldade necessário para a introdução de conteúdos como a lógica computacional para crianças tão jovens.

2.3 CONSIDERAÇÕES FINAIS

Introduzir a computação para crianças pode contribuir com o desenvolvimento de diversas habilidades. O raciocínio lógico e o pensamento computacional estão entre estas habilidades. Além de essenciais para todas as pessoas, elas são cada vez mais buscadas em profissionais no mercado de trabalho. Tais habilidades são muito importantes em um mundo tecnológico como hoje e a demanda crescente por profissionais de área de informática, pode torná-la potencialmente promissora para as gerações futuras.

O fato de se utilizar tecnologias em sala de aula, estimula habilidades diferentes nas crianças. O brincar como auxílio na educação, além de essencial, é uma alternativa para manter a atenção e estimular as crianças de maneira eficiente. Existem diversas e variadas ferramentas para a inserção de lógica de programação na escola, seja puramente o ensino através de softwares ou em conjunto com robótica. Tanto o software isoladamente, quando em conjunto com a robótica, possuem seus pontos positivos e negativos. Cabe ao professor que irá introduzir o conteúdo, a escolha que mais se adequar com a realidade da escola e também da turma. Encontrar um mecanismo para auxiliar o ensino de programação vai depender também dos recursos disponíveis e principalmente a idade dos alunos.

Softwares, em sua maioria, são baratos se comparados com a utilização combinada da robótica. Mas, em contrapartida, não oferecem às crianças os benefícios que a robótica na educação proporciona. O alto custo da utilização da robótica na educação, pode ser um entrave para sua inserção. Aliado a isso, foi observado que existe uma escassez de ferramentas voltadas a crianças mais novas de 4 a 6 anos (educação infantil). Os produtos encontrados não têm como foco principal crianças nessa faixa etária. O que também foi constatado é a falta de recursos que os softwares têm para auxiliar o professor no ensino.

Observou-se que os softwares pesquisados não oferecem recursos para adaptação do seu uso, sendo aplicáveis em situações de aprendizagem previamente definidas. Este fato limita a possibilidade de o professor produzir o seu material didático ou ainda reutilizar parcialmente um material disponível. Esta característica torna os softwares analisados restritos ao uso previamente definido, diferente do que se propõe neste trabalho.

A fim de fazer uma análise sobre as características dos softwares que foram pesquisados foi desenvolvida uma tabela com o intuito de ranquear os produtos pesquisados para o trabalho. Dentre os quesitos que serão avaliados, estão a idade orientada para o produto, o preço, a complexidade, a interface, o conteúdo, e o também a utilização de robótica como diferencial. A tabela 1 define os critérios de avaliação que foram utilizados com o objetivo de ranquear os softwares e produtos pesquisados. Os critérios são: idade, custo, complexidade, interface, conteúdo e robótica. Foi elaborado um grau de avaliação para que os softwares possam ser ranqueados.

A tabela 2 apresenta a pontuação dos softwares pesquisados. Pode-se analisar que, softwares com qualidade existem no mercado. O *Scratch*, Robô Dash, *The Foos* e o kit Lego foram as ferramentas mais completas e por isso tanto sua complexidade é maior como a interface mais carregada. O Se-Robô por ser um protótipo e não estar no mercado ainda é uma ferramenta simples, mas com um bom potencial para a inserção de lógica computacional para crianças, devido a sua facilidade e simplicidade. O Bee-Bot não utiliza software, muito menos aplicativo, apenas o robô, porém possui uma metodologia de ensino simples e objetiva. O *Codie* e o *LightBot* pessoalmente foram os que mais me agradaram para o objetivo da pesquisa. Devido a fatores como a utilização da robótica no *Codie* e a complexidade e interface do *LightBot*.

Identifico que os softwares são de elevada complexidade para crianças da educação infantil, muitos dos símbolos e contextos utilizados nos softwares pesquisados não fazem parte de seu conhecimento, tornando assim o aprendizado de programação nessas ferramentas inviável. Softwares baseados em jogos são importantes para a inserção do lúdico no ensino, porém somente a utilização dos jogos não significa aprendizado. É necessário a elaboração do jogo com enfoque no aprendizado, e ninguém melhor que o próprio professor tem capacidade de elaborá-los.

A utilização da robótica como auxiliar é indispensável, por todos os benefícios atrelados a ela. O único ponto negativo é o preço, tornando ainda mais necessário a elaboração de projetos que possam baratear a tecnologia.

O professor, esse que possui convívio diário com seus alunos, conhece suas dificuldades e suas qualidades. Ele precisa ser amparado com a tecnologia, tendo uma ferramenta que possa ser adaptada às suas necessidades e que traga um feedback do aprendizado, para que assim possamos introduzir a lógica computacional nas crianças.

Tabela 1 – Critérios de Avaliação

Critério	Conceito	Avaliação
Idade	A idade de utilização é importante pois o aluno a cada idade possui características diferentes, e um ano a mais ou a menos, faz com que o ensino seja totalmente diferente. Logo, de nada adianta o software ser eficiente, se para o aprendizado do mesmo o aluno precisa de conhecimentos que ainda não possui.	3 níveis de avaliação que serão: 4+anos, 6+anos, 8+anos
Custo	O custo também será analisado como quesito, por que para a introdução da ciência da computação em sala de aula, é preciso de recursos financeiros. Seja para materiais didáticos, ou para capacitação dos professores, como também o preço do software ou computador. O preço elevado faz com que se faça necessário buscar meios alternativos que possam dar boas respostas de igual modo.	Para o custo a avaliação será a seguinte: Gratis, Mensal, Pagamento Único
Complexidade	A complexidade para a avaliação, é entendida como o grau de dificuldade que os alunos irão demonstrar com a utilização do software em questão. A complexidade pode variar de aluno para aluno. O que também influencia a complexidade, é a quantidade de opções ou comandos que são disponíveis na ferramenta, dificultando o entendimento total da ferramenta.	Para avaliação da complexidade foi estimado 3 níveis: Pouco Complexa, Complexa, Muito Complexa
Interface	A interface é importante, por que é nela que o aluno irá trabalhar. Ela precisa ser feita de maneira coerente e sem ambiguidades que possam confundir os alunos. Uma boa interface pode também, fazer com que os alunos se sintam motivados e gostem de utilizar a aplicação.	Foi definido que a avaliação da interface será do seguinte modo: Fácil Entendimento, Normal, Difícil Entendimento
Conteúdo	O Conteúdo é visto como o total de opções que o software, possui para o ensino de programação. Ele é importante pois, um software que só faz uma coisa pode em poucas utilizações se tornar obsoleto e cansativo. É importante que o software seja completo e que possa se adaptar ou melhor ao longo do tempo.	Para o conteúdo foram elaborados dois níveis: Falta Conteúdo, Normal, Completo
Robótica	A utilização da robótica no software, também será levada em consideração a fim de ranquear os softwares. Como foi constatado, a importância do concreto para crianças da educação infantil e também o lúdico tornando o ensino mais proveitoso para as crianças	A robótica será avaliada pela utilização ou não da mesma.

Fonte: Elaborada pelo autor.

Tabela 2 – Pontuações para os Softwares Pesquisados

Software	Idade	Custo	Complexidade	Interface	Conteúdo	Robótica
Scratch	8+anos	Grátis	Muito Complexa	Difícil Entendimento	Completo	sim
LightBot	4+anos	Pg Único	Pouco Complexa	Fácil Entendimento	Normal	não
The Foos	6+anos	Mensal	Complexa	Normal	Completo	não
Codie	6+anos	Pg Único	Complexa	Normal	Normal	sim
Robô Dash	6+anos	Pg Único	Muito Complexa	Difícil Entendimento	Completo	sim
Se-Robô	4+anos	Pg Único	Pouco Complexa	Fácil Entendimento	Falta Conteúdo	sim
Lego	8+anos	Pg Único	Muito Complexa	Difícil Entendimento	Completo	sim
Bee-Bot	4+anos	Pg Único	Pouco Complexa	Fácil Entendimento	Falta Conteúdo	sim

Fonte: Elaborada pelo autor.

3 DESENVOLVIMENTO DO SOFTWARE

O presente capítulo descreve o desenvolvimento do software projetado neste trabalho. O objetivo do software é introdução de conceitos de ciência da computação para crianças utilizando atividades lúdicas.

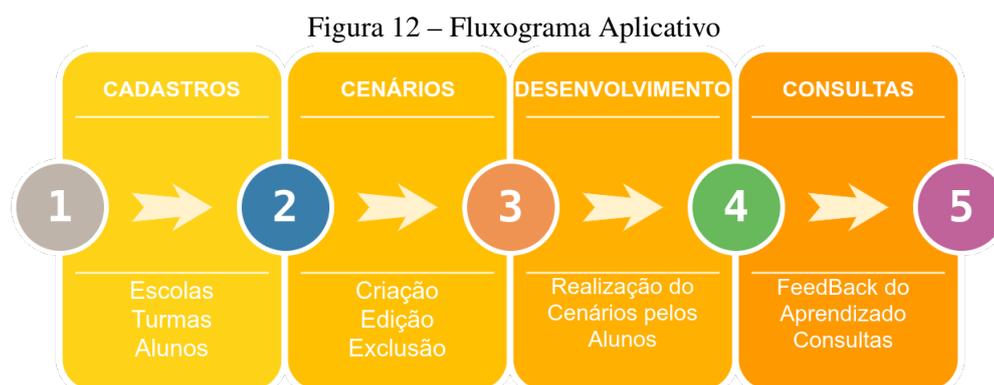
Optou-se pelo desenvolvimento de um aplicativo para o sistema operacional móvel Android. A escolha em se desenvolver para o sistema Android, se deve principalmente por ser uma plataforma móvel e gratuita. O aplicativo foi implementado em linguagem de programação Java.

O aplicativo tem por característica a comunicação *bluetooth* com um robô desenvolvido em *Arduino*. O robô foi desenvolvido por Guilherme Duso, colega do curso de Engenharia de Automação. Portanto neste trabalho apenas serão abordados os assuntos pertinentes ao aplicativo.

As funções do aplicativo podem ser divididas em quatro componentes, assim detalhados:

- **Cadastros:** compreende o cadastro de escolas, turmas e alunos.
- **Cenários:** engloba o cadastro e customização dos cenários de ensino que serão trabalhados pelos alunos.
- **Desenvolvimento:** compreende o uso dos cenários de ensino pelos alunos.
- **Consultas:** consultas dos dados gerados pelo desenvolvimento dos cenários.

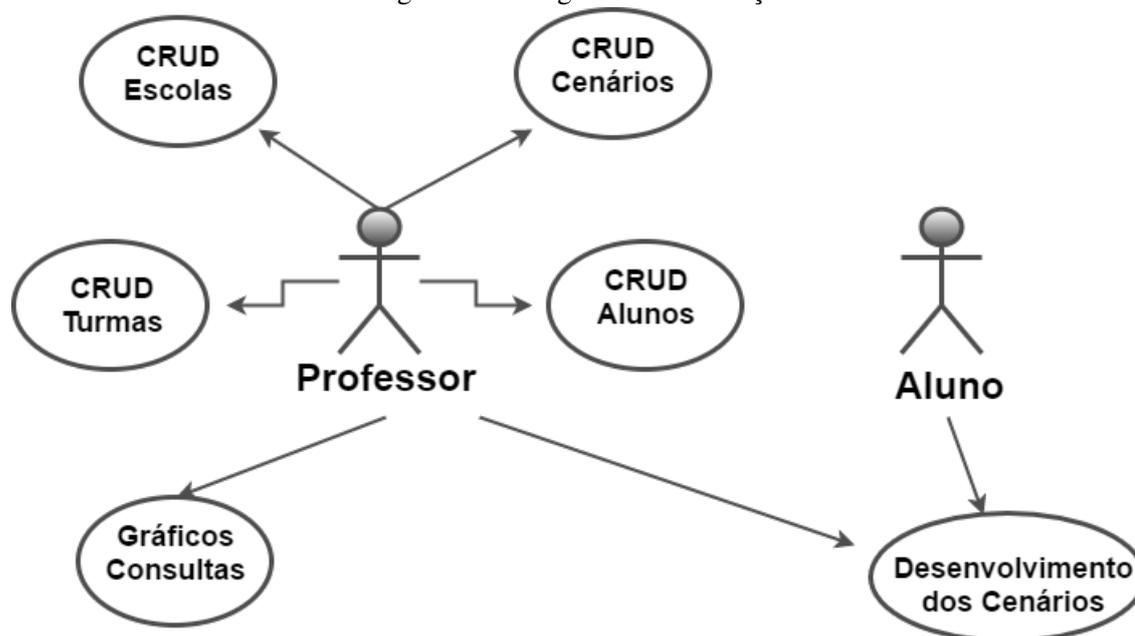
A Figura 12 ilustra com um fluxograma o aplicativo. A numeração indica a ordem em que as atividades devem ser realizadas.



O aplicativo foi projetado, para a utilização por dois atores o professor e o aluno. O professor terá acesso total ao aplicativo, enquanto os alunos somente terão disponíveis, a resolução dos cenários. A Figura 13 ilustra o acesso de cada ator ao sistema. Entende-se como CRUD (acrônimo de Create, Read, Update e Delete na língua Inglesa) os módulos de cadastro, exclusão, acesso e atualização de uma tabela.

Dessa forma, a seguir são descritas características do aplicativo que será desenvolvido. Também são apresentadas a arquitetura, a persistência dos dados, os cenários de ensino, a comunicação, os protótipos de tela, a aplicação e os testes e experimentos.

Figura 13 – Diagrama da Utilização



3.1 ARQUITETURA

O aplicativo foi projetado para ser utilizado em *smarthphones* e *tablets*. O intuito é exibir a execução dos cenários no aplicativo, além de transmitir os comandos que o aluno definiu para a execução no robô. Uma interface específica gerencia a manipulação de cenários. Como é utilizado um robô para a visualização da ação dos comandos, a sala de aula deve conter um tabuleiro real. Com isso se faz necessária a customização de cenários de ensino, a fim de se adequar ao espaço físico disponível ao professor. Para garantir que o projeto ofereça os recursos necessários é preciso garantir uma boa modularidade de código entre os componentes que realizam a interface do aplicativo, desde o nível arquitetural.

No desenvolvimento do projeto foi levado em consideração o fato de as interfaces de dispositivos com suporte ao Android serem de variados tamanhos. Por causa disso foram utilizados Fragmentos (Fragments). Com a utilização deste padrão é possível criar interfaces mais ricas e interativas. Fragmentos são utilizados a fim de organizar melhor a interface do aplicativo, modularizando também as funcionalidades que estão disponíveis para o usuário.

Cada Fragmento delimita-se ao seu próprio escopo, sendo capaz de tratar seus próprios eventos e atividades, executar processos de maneira independentes, como também gerenciar seus próprios componentes gráficos. Com essa característica se obtém mais flexibilidade e extensibilidade para o aplicativo, uma vez que a interface tende a sofrer alterações na forma de exibição do conteúdo. Com a utilização de Fragmentos é possível adicionar conteúdos e removê-los sem alterar a funcionalidade dos outros componentes. O padrão Fragmentos, permite ao desenvolvedor o *design* de *layouts* flexíveis e responsivos, o que significa que a mesma aplicação pode se adaptar a diferentes tamanhos de tela de dispositivos. Isso evita retrabalho de

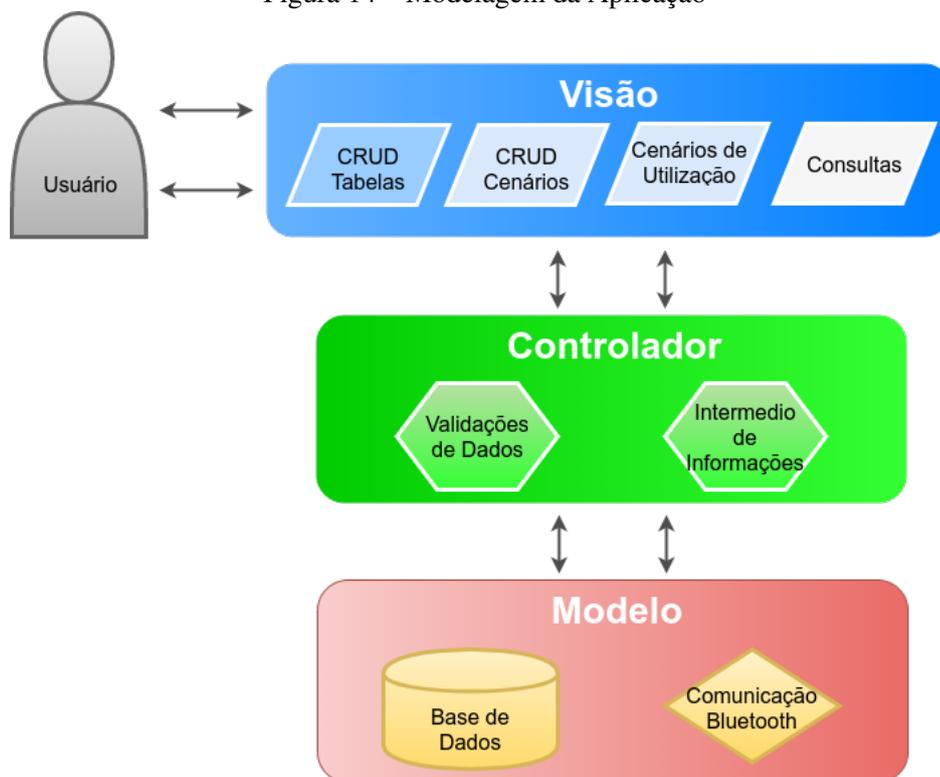
codificação para os diferentes tamanhos de tela existentes.

Para a arquitetura do projeto, optou-se pela utilização do padrão MVC (Model, View, Controller, que pode ser traduzido como Modelo, Visão e Controlador). Com a utilização desse padrão, a aplicação divide-se nas camadas: camada de visão (View), controlador (Controller) e modelo (Model). A camada de visão, é responsável pela interface com o usuário, o controlador é responsável por todas as regras e validações da aplicação e é o controlador que faz a ponte entre a visão e o modelo, e a camada de dados que é responsável pelo armazenamento das informações em um banco de dados e também pela comunicação *bluetooth*.

Tendo em mente o padrão de arquitetura mencionado, realizou-se a modelagem da aplicação. A figura 14 mostra o fluxo de interações entre o usuário e os módulos da arquitetura.

- **Visão:** camada que manipula os eventos da interface do usuário, se comunica exclusivamente com o controlador. Portanto, qualquer comunicação que a visão precise realizar com a camada do modelo, deve ser intermediada pelo controlador. Esta camada é composta por três submódulos:
 - **CRUD Tabelas:** a subcamada é a responsável por realizar os cadastros do aplicativo, como cadastro de escolas, turmas, e alunos. Essa informação é importante pois, para a avaliação dos alunos é necessário armazená-los no aplicativo.
 - **CRUD Cenários:** aqui é onde o professor pode adicionar, remover e editar cenários, a fim de adequá-los à realidade dos alunos, criando cenários customizados para seus alunos.
 - **Cenários de Utilização:** é o módulo que as crianças utilizam. Os cenários de utilização são compostos por um tabuleiro e um quadro de opções de comandos. O aluno precisa utilizar os comandos a fim de completar o cenário de ensino.
 - **Consultas:** essa subcamada é responsável por trazer ao professor o resultado dos cenários realizados pelos alunos. Possui diversos filtros para o professor filtrar os dados como desejar.
- **Controlador:** é a camada responsável por fazer o intermédio de informações, entre a visão e o modelo. É no controlador também que são realizadas validações dos cenários de ensino e também validações de dados informados pelo usuário.
 - **Intermédio:** subcamada que realizará o intermédio de informações oriundas da visão para o modelo e também do modelo para visão. A realização da modularização é importante, a fim de deixar o código estruturado e tornar mais fácil a manutenção e adição de componentes.
 - **Validações:** aqui é principalmente validados os dados da visão, referentes aos cenários e também os dados dos cadastros que foram inseridos da visão. É aqui também que é tratado as mensagens de retorno para visão que vieram do modelo.

Figura 14 – Modelagem da Aplicação



- **Modelo:** última camada da hierarquia, o modelo é dividido em duas grandes partes. A primeira é a base de dados do aplicativo, e a outra é a comunicação *bluetooth*. O modelo devolve as informações que forem solicitadas destes dois lugares e retorna para o controlador que então devolve para onde a informação foi solicitada.
 - **Base de Dados:** A base de dados é a primeira subcamada do modelo, é aqui que estará toda a lógica de armazenamento, consulta e exclusão de informações das tabelas do software.
 - **Comunicação:** aqui será implementado a lógica para a comunicação *bluetooth*, com o robô que será utilizado no projeto.

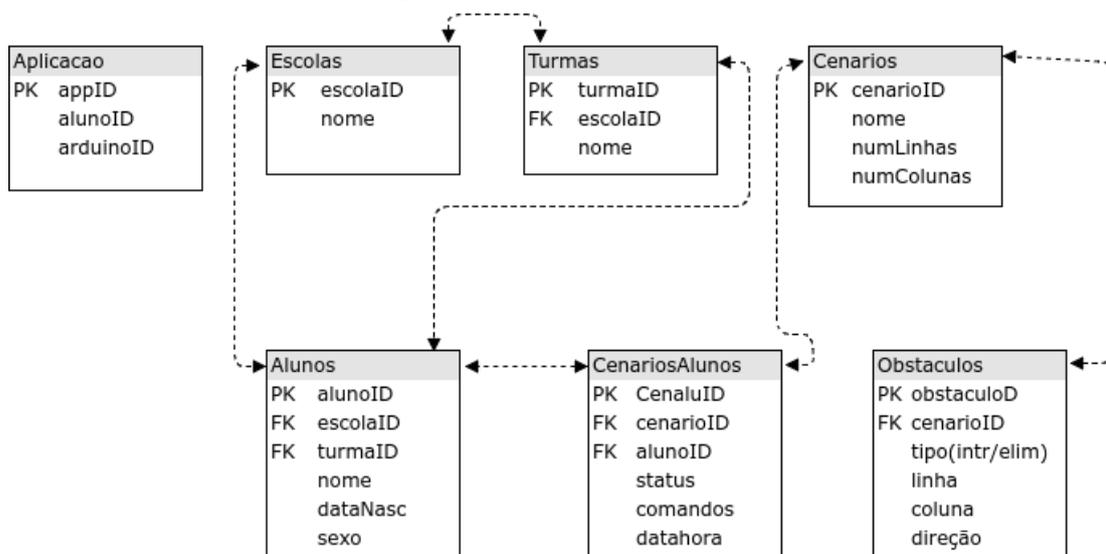
3.2 PERSISTÊNCIA DOS DADOS

A persistência dos dados é realizada com a utilização do *SQLite*. O Android por padrão, tem suporte completo a biblioteca para criação e acesso ao banco de dados *SQLite*. Essa familiarização do Android com o *SQLite* é fator fundamental para a utilização desta ferramenta no projeto.

Segundo (COSTA; SANTOS, 2015) "a biblioteca *SQLite*, é um gerenciador de banco de dados independente de servidor e sem necessidade de configurações". Ele armazena toda a estrutura e configuração do banco de dados, inclusive os dados no mesmo arquivo. Isso o torna leve e seguro.

Tablets e smartphones não possuem um *hardware* de um computador, logo, a simplicidade e eficiência são características que é preciso levar em consideração. Para tanto, a definição das tabelas do aplicativo foram definidas conforme a Figura 15

Figura 15 – Tabelas do Aplicativo



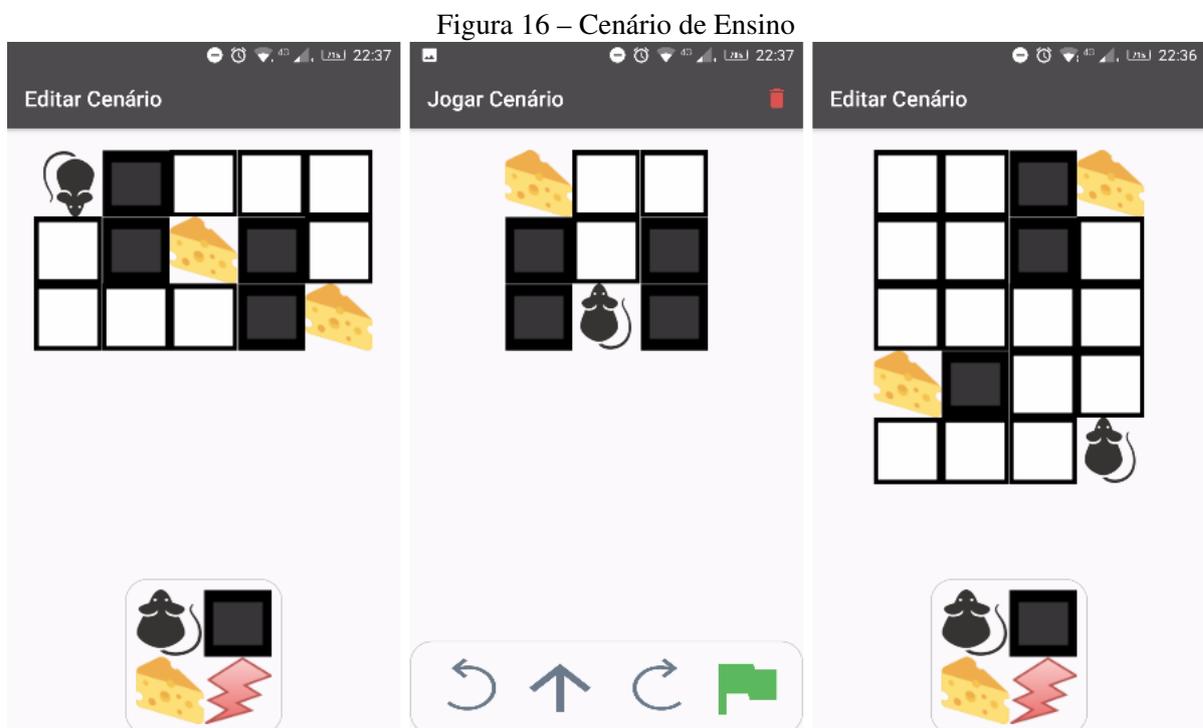
As tabelas do aplicativo ilustradas Figura 15 são descritas abaixo:

- **Aplicacao:** essa é a tabela que guarda as informações que são necessárias pela aplicação. Nela é armazenado a indentificação do aluno que está utilizando o aplicativo, e também a identificação do *Arduino* que está sendo utilizado.
- **Escolas:** como meio de organizar o aplicativo para o professor, o aplicativo tem uma tabela para o cadastro de Escolas. Com isso é possível ao professor ter o controle do aprendizado dos alunos por escola, ao qual ele leciona.
- **Turmas:** assim como a escola, são várias turmas ao qual o professor está inserido. Com a tabela de turmas, o professor cadastra todas as turmas ao qual ele trabalha em uma determinada escola.
- **Cenarios:** os cenários são cadastrados pelo professor, o objetivo de elaborar cenários com maior ou menor dificuldade. Com isso é possível, um ensino dirigido ao aluno. Para tanto, o professor tem uma interface no aplicativo para a inserção, edição e exclusão de cenários de utilização. O professor também define, o número de linhas e colunas do tabuleiro e também qual o início e fim do cenário.
- **Obstaculos** atrelados aos cenários os obstáculos, como o nome sugere, são os obstáculos em um determinado cenário. Eles possuem um tipo que define se o obstáculo é de eliminação, ou um obstáculo intransponível. Cada obstáculo definido pelo professor é armazenado nesta tabela, junto com a posição definida pelo professor.

- **Alunos** os alunos de uma turma em uma escola estão cadastrados nesta tabela. Informações importantes como a data de nascimento e o sexo, são armazenadas a fim de ter estatísticas referentes a estes critérios.
- **CenáriosAlunos** é desta tabela que é retirada as informações de sucesso ou não de cada aluno. Conforme o aluno vai progredindo nos cenários de ensino, as informações sobre os comandos utilizados e a data e hora do desenvolvimento do cenário são armazenadas aqui.

3.3 CENÁRIOS DE UTILIZAÇÃO DOS ALUNOS

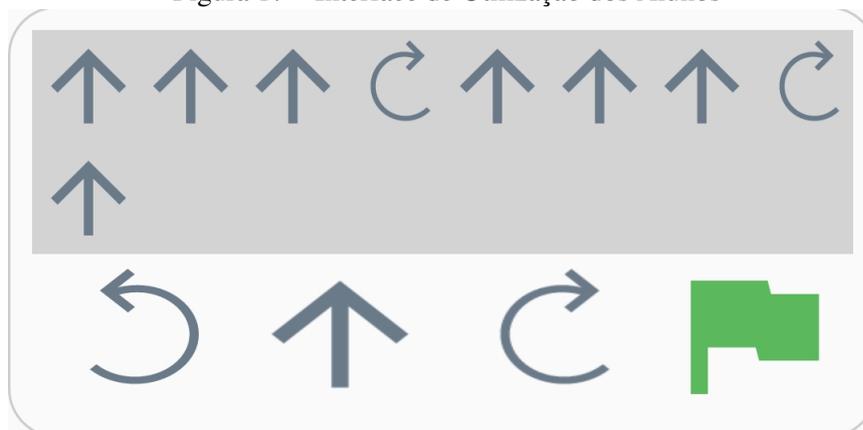
Cenários de Ensino são os cenários de utilização dos alunos. Este nome foi dado aos desafios propostos pelo aplicativo. Cada cenário é composto por um tabuleiro, representado por n linhas e m colunas. Um ponto de início é representado por um ratinho. O fim é representado pelo queijo. Podem existir k queijos em um cenário, e o cenário só termina após o ratinho comer todos os queijos. E, por fim, pode-se inserir os obstáculos. A Figura 16 ilustra três cenários de utilização. As informações de cada cenário são armazenadas em tabelas e com isso é possível a inserção, a edição e exclusão de cenários pelo professor.



Os cenários serão solucionados pelos alunos, tendo como possibilidade uma lista de comandos que serão disponíveis a eles. A Figura 17 ilustra a interface que o aluno terá disponível.

O aluno conta com quatro comandos para a resolução dos cenários. Cada comando terá uma ação que será ilustrada no aplicativo e também pelo robô. Os três primeiros comandos são os

Figura 17 – Interface de Utilização dos Alunos



comandos que serão utilizados para movimentar o robô. A seta para frente envia o comando para o robô mover um espaço para frente. As setas de giro laterais fazem o robô girar 90 graus na direção apontada pela seta, sem se mover do espaço onde ele está.

A interface de utilização é dividida em 2 partes a primeira, acima com o fundo cinza, são os comandos já selecionados pelo aluno. A segunda é a parte inferior, com o fundo em branco, são os comandos disponíveis pelo aluno.

Para selecionar um comando, o aluno precisa tocar no símbolo que ele deseja inserir. Ao tocar no símbolo, ele vai para o quadro dos comandos selecionados. Tocando no comando dentro do quadro de comandos, o comando é excluído.

Para a execução dos comandos montados pelo aluno é preciso tocar na bandeira verde. É preciso montar toda a lógica para resolução do cenário antes tocar na bandeira.

Após tocar na bandeira o ratinho, tanto no aplicativo como o robô, irão se mover executando os comandos selecionados no quadro de comandos.

A figura 18 ilustra exemplos de cenários que podem ser criados com o aplicativo.

Os cenários de utilização são o coração do trabalho. É possível fazer um cenário de ensino de qualquer tamanho. A dificuldade atrelada ao cenário, será também definida pelo professor. Após a definição do tamanho do cenário, é preciso definir na matriz qual será o ponto de partida e o ponto de término do cenário.

Para agregar dificuldade ao cenário pode-se colocar obstáculos nele. Pode também inverter a orientação do robô e com isso forçar o aluno a fazer os comandos espelhados, tornando a complexidade maior.

O formato de cenários como proposto foi pensado com base em dois fatores determinantes:

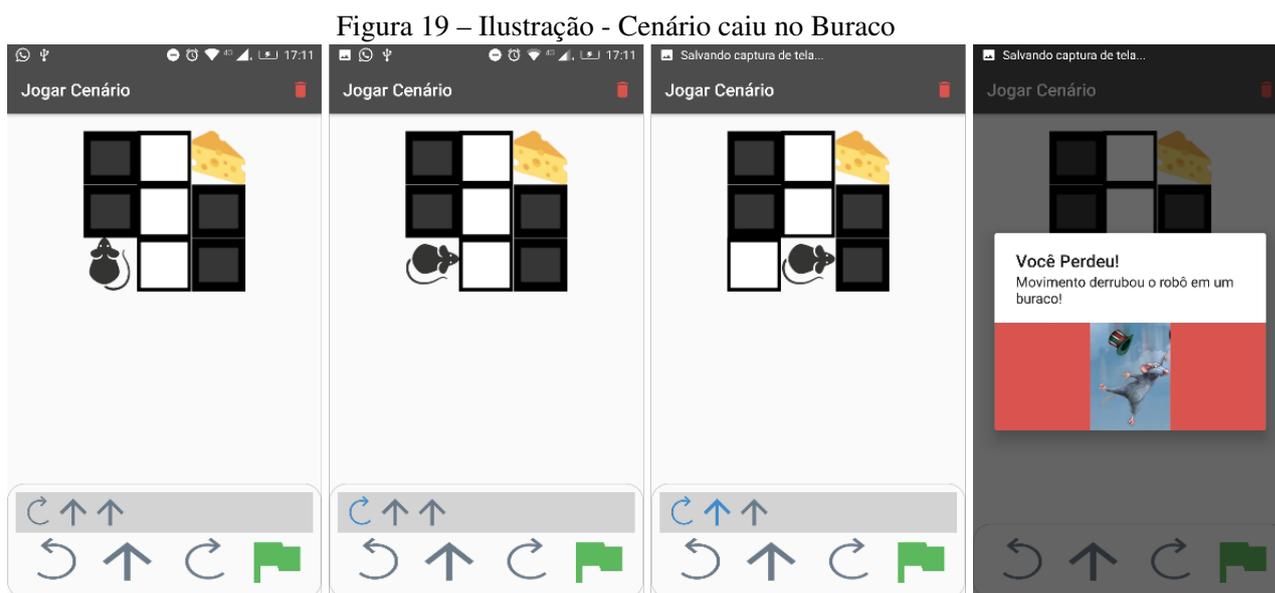
- O primeiro é a disparidade de aprendizado em uma turma de aula. Em uma turma existem alunos que aprendem rápido e outros que o tempo de aprendizado é mais lento. Para o professor esse fator é um problema pois é preciso preparar atividades para todos os alunos. O fato de poder customizar os cenários de ensino é a chave para um bom aprendizado.
- O segundo é o fato de que o aplicativo está enviando os comandos para um robô. As salas

3.3.1 Resolução dos Cenários

Após clicar na bandeira verde, um laço é iniciado e os comandos são enviados para o robô via *bluetooth*, um a um. No aplicativo, será ilustrado o movimento do robô. A seguir serão ilustrados como o aplicativo trata as possibilidades de solução do aluno.

Existem quatro possibilidades de resultado após o aluno clicar na bandeira verde. A primeira é o robô cair em um buraco. A segunda é sair para fora do cenário. A terceira é se movimentar pelo cenário e não terminar o objetivo. E a quarta e última é comer todos os queijos e terminar o cenário com sucesso.

Caso os comandos derrubem o robô em um buraco, uma mensagem aparece na tela em vermelho e um desenho de um ratinho caindo no buraco é mostrado na mensagem. A figura 19 mostra um exemplo do que ocorre no aplicativo quando o robô cai no buraco. O aplicativo executa um som junto com a tela ilustrada. Dois são os áudios executados ao cair em um buraco: "Oh seus comandos me derrubaram no buraco" e "Oh socorro caí no buraco".



Se os comandos mandarem o robô para fora do cenário, uma imagem de um rato pensativo aparece e um dos sons aleatórios é emitido. "Oh não sai para fora do cenário" e "estou perdido, onde eu vim parar". A figura 20 ilustra esse processo no aplicativo.

Se os comandos do aluno fizerem o robô andar pelo cenário mas não completem o objetivo, uma mensagem aparece na tela. Um som também é emitido: "Humm estou sentindo o cheiro do queijo, ele está por perto" e "Humm que fome! onde está o queijo?". A figura 21 mostra esse processo no aplicativo.

Conforme o ratinho vai comendo os queijos um som emitido: "Ham Ham que queijo delicioso" e "Humm que delicia de queijo". Ao completar o cenário uma mensagem de sucesso aparece na tela, a figura 22 ilustra a mensagem de sucesso.

Figura 20 – Ilustração - Cenário Saiu para fora do cenário

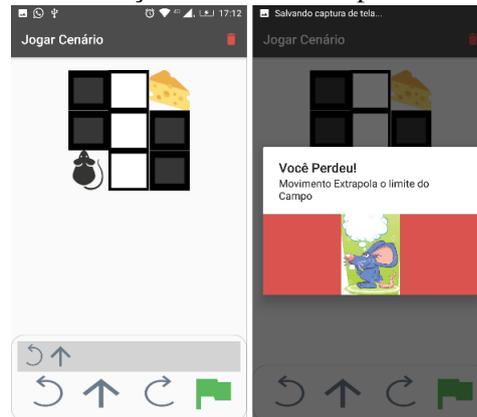


Figura 21 – Ilustração - Cenário Quase lá

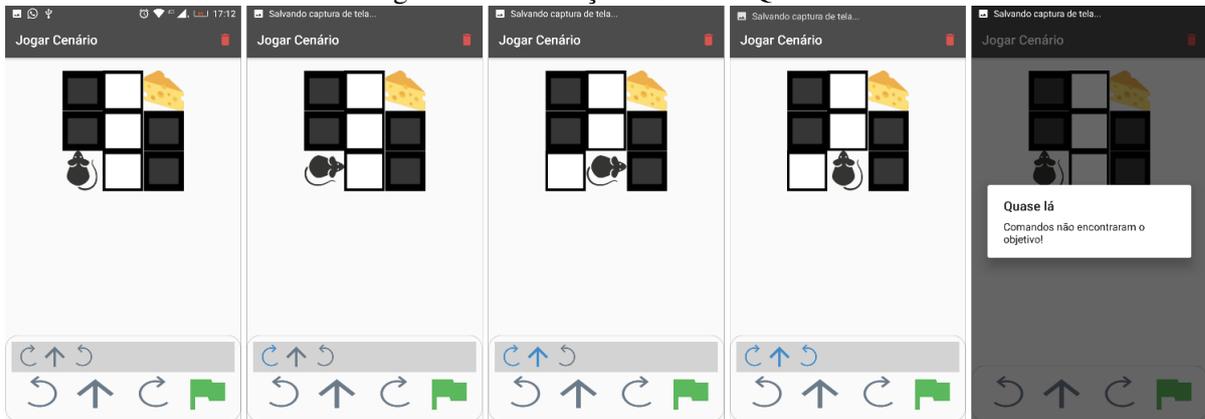
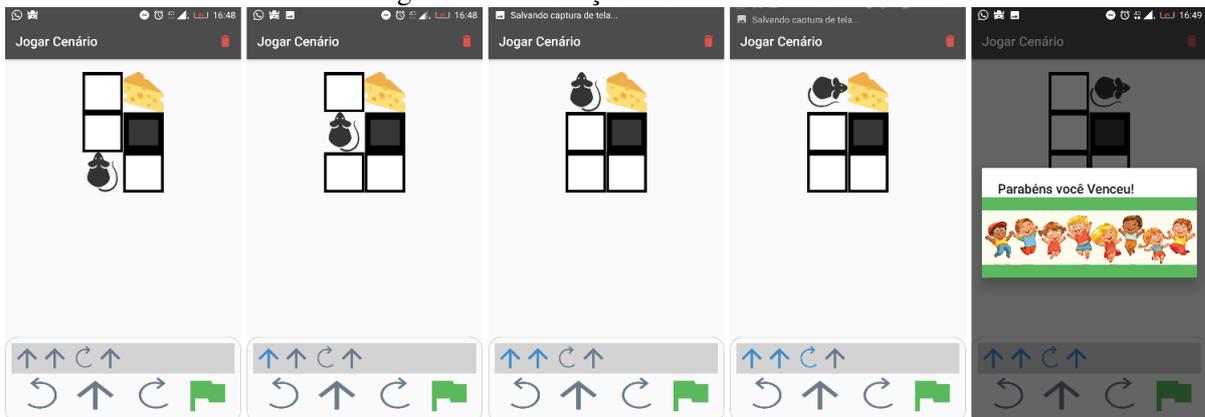


Figura 22 – Ilustração - Cenário Sucesso



Após a execução do cenário, os comandos que foram utilizados pelo aluno ou pelos alunos, são armazenados na tabela 'CenáriosAluno'. Com isso é possível fazer as consultas dos resultados dos cenários.

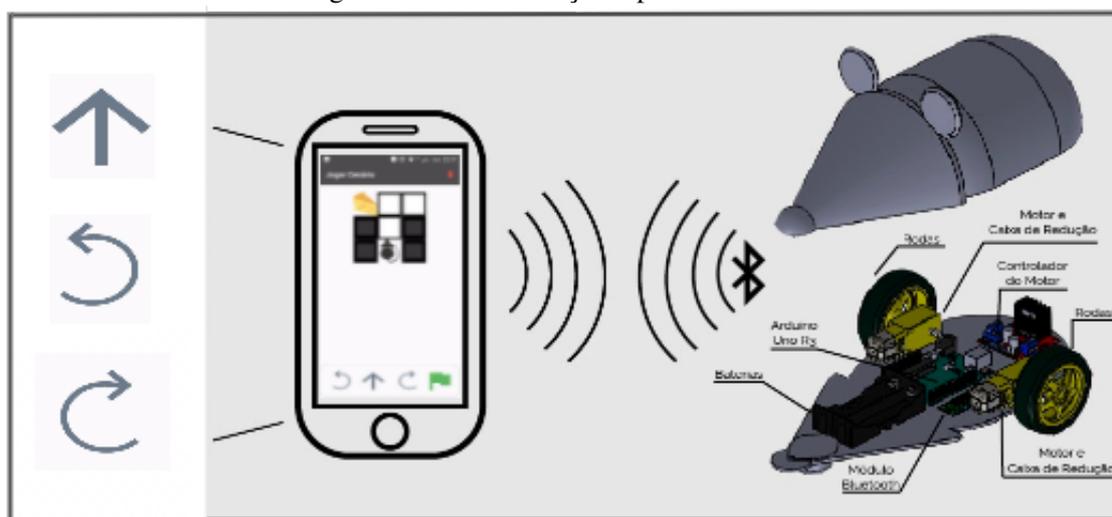
3.4 COMUNICAÇÃO COM O ROBÔ

Para a comunicação com o robô é utilizada a tecnologia *bluetooth*. O *Arduino*, conta com um módulo *bluetooth* que faz a comunicação com o aplicativo.

Para utilização do *bluetooth* no Android é utilizada uma classe que estende os métodos da classe *threads*. Essa classe é a intermediadora da comunicação *bluetooth*. Também nessa classe é utilizado um objeto do tipo *BluetoothAdapter*, e com ele então é feita a comunicação com o *Arduino*.

Quando o usuário executar o programa clicando na bandeira verde, o aplicativo entra em um loop nos comandos selecionados pelo aluno, e envia um a um os comandos para serem interpretados pelo robô. Para fins de comunicação, são utilizadas internamente as letras "w" para movimento para cima, "a" para o giro 90 graus à esquerda, e "d" para o giro 90 graus para direita.

Figura 23 – Comunicação Aplicativo x Robô

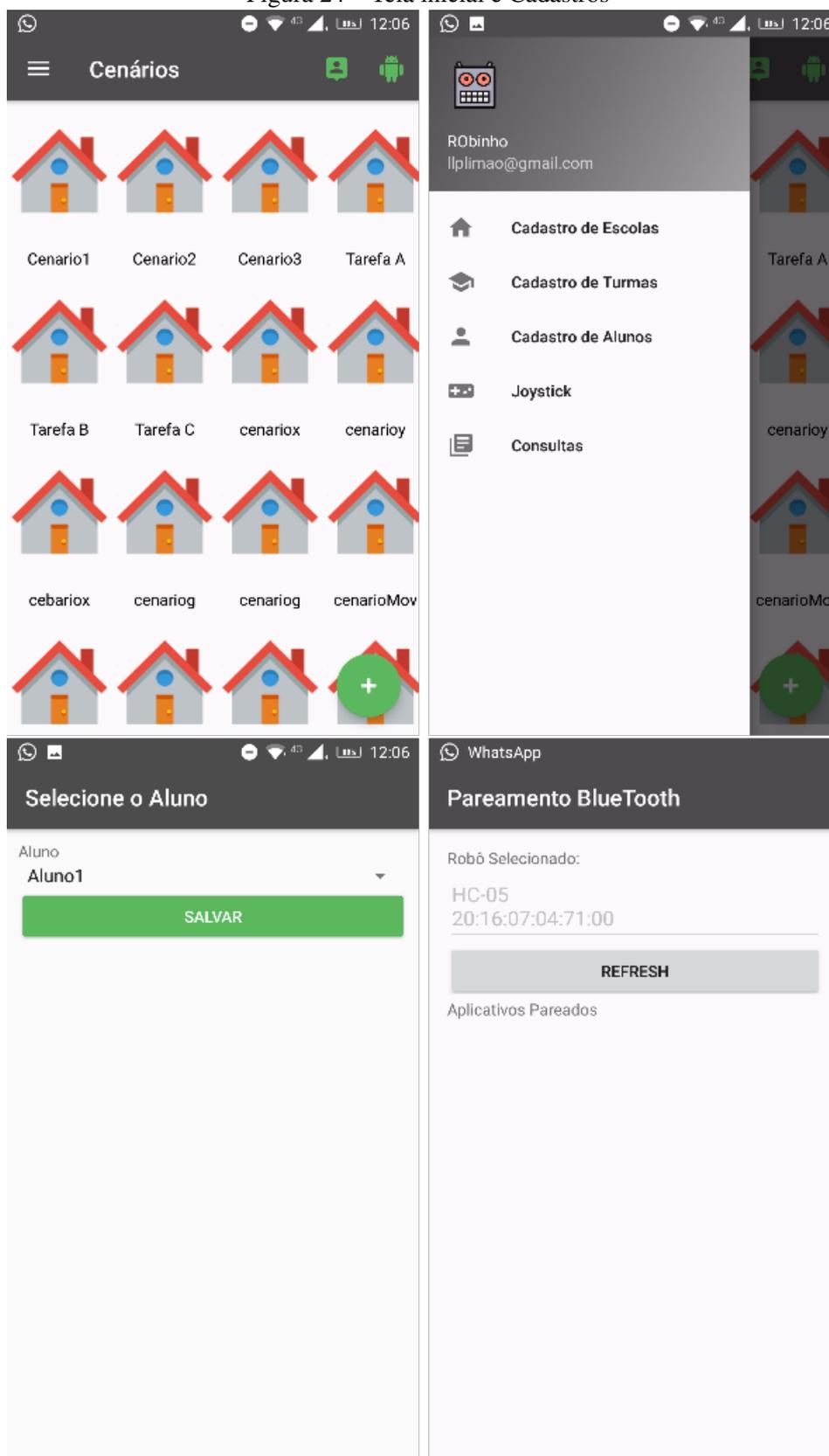


3.5 PROTÓTIPOS DE TELA

Nesta seção são apresentados imagens do aplicativo para exemplificar melhor as funcionalidades do aplicativo. O aplicativo desenvolvido pode ser usado seguindo-se quatro passos. O primeiro passo compreende o cadastro de escolas, das turmas e dos alunos. O segundo passo compreende a criação e edição dos cenários de utilização. O terceiro passo permite a utilização e programação dos cenários pelos alunos. O quarto passo envolve consultas dos programas desenvolvidos pelos alunos.

A figura 24 ilustra a tela inicial do aplicativo que contém os cenários de utilização. O botão verde na parte inferior da tela é utilizado para adicionar um novo cenário. Para edição de um cenário é preciso fazer um *long tap* no cenário em questão. Existem outros dois botões no canto superior direito. O primeiro é o ícone de um robô que acessa o cadastro dos robôs do aplicativo,

Figura 24 – Tela inicial e Cadastros



e o outro é uma figura de pessoa onde pode ser configurado o aluno atual do aplicativo. O menu lateral se abre ao apertar as três barras no canto superior esquerdo.

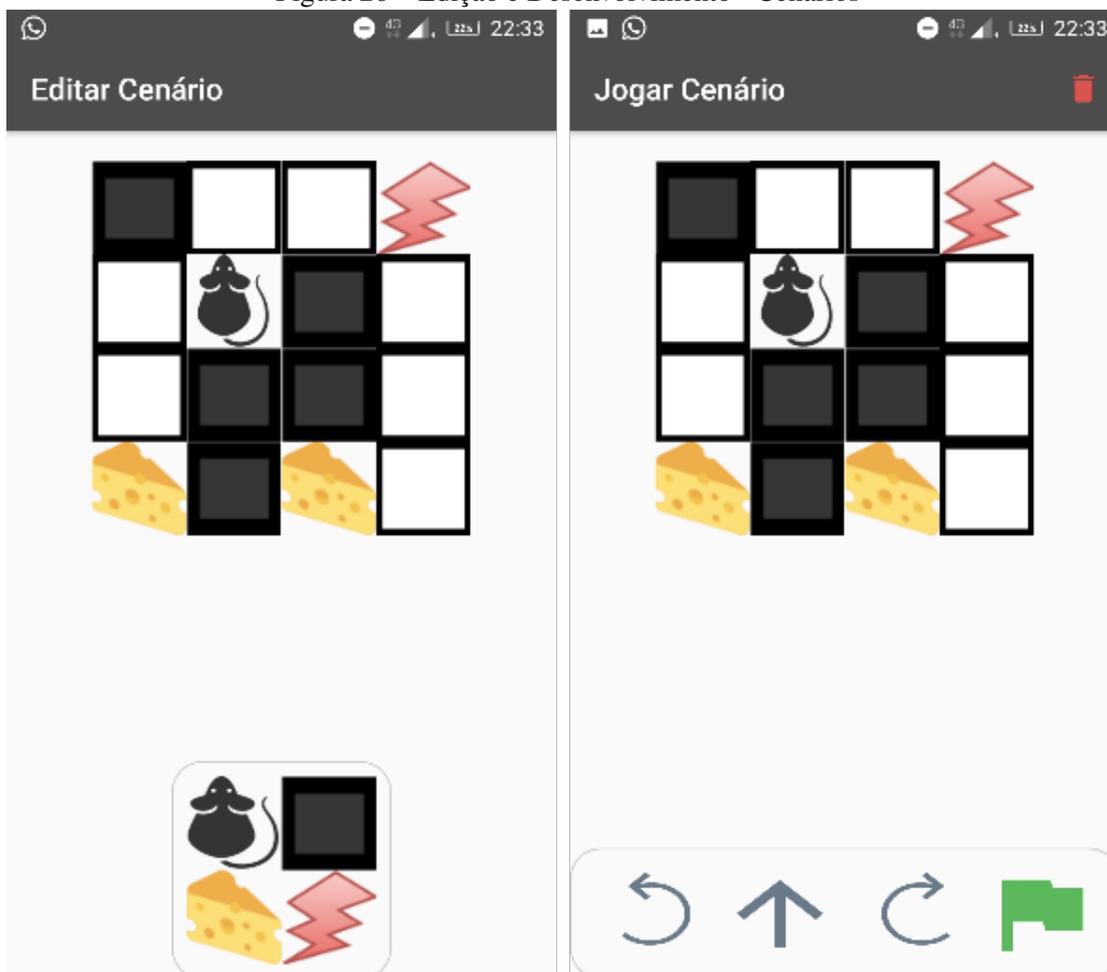
Figura 25 – Página Principal e Consultas

The figure displays four screenshots of a mobile application interface, arranged in a 2x2 grid. Each screenshot shows a different screen with a dark header bar and a white content area.

- Top Left: Cadastro de Cenários**
 - Header: Mensagem de +55 51 9599-7001 @ Equipe Rocket
 - Form: Nome (cenariog), Num Linhas (5), Num Colunas (3)
 - Buttons: OBSTÁCULOS (blue), SALVAR (green), EXCLUIR (red)
- Top Right: Cadastro de Escolas**
 - Header: Mensagem de +55 51 9599-7001 @ Equipe Rocket
 - Form: Nome (empty)
 - Buttons: SALVAR (green)
- Bottom Left: Cadastro de Turmas**
 - Header: Mensagem de +55 51 9599-7001 @ Equipe Rocket
 - Form: Escola (Santa Mônica), Nome (empty)
 - Buttons: SALVAR (green)
- Bottom Right: Edição de Alunos**
 - Header: Mensagem de +55 51 9599-7001 @ Equipe Rocket
 - Form: Escola (Santa Mônica), Turma (Turma1), Nome (Aluno1), Sexo (Masculino), Data de Nascimento (2012-10-18)
 - Buttons: SALVAR (green), EXCLUIR (red)

A figura 25 exibe a tela de edição dos cenários, os cadastros de escolas, turmas e alunos. Os cadastros são acessados através do menu ilustrado na figura 24. A Partir do cadastro de cenário é possível acessar a tela de edição dos cenários, clicando no botão azul, obstáculos. O acesso aos cadastros é através do botão verde que fica na parte inferior das telas de consulta.

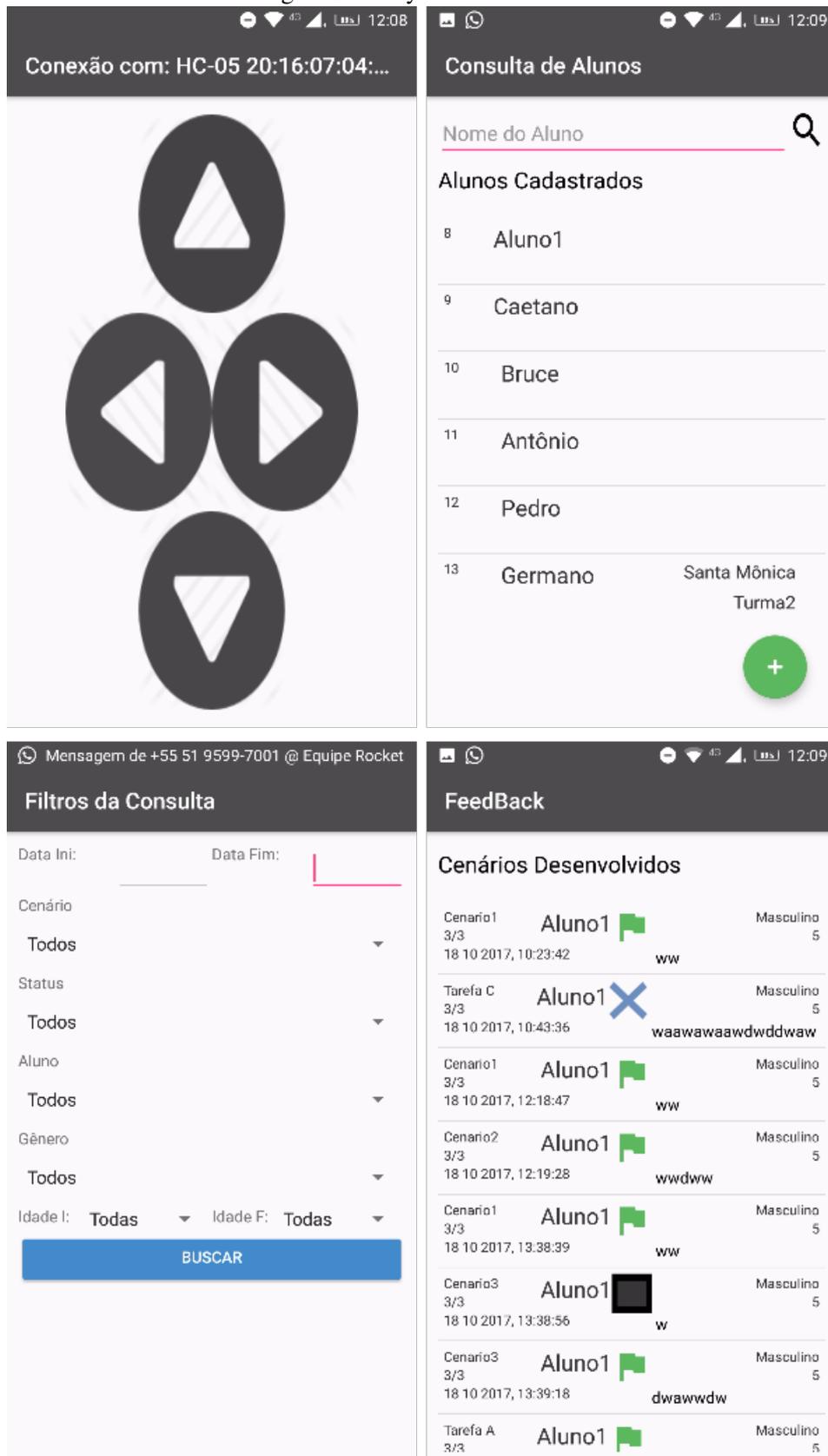
Figura 26 – Edição e Desenvolvimento - Cenários



A figura 26 mostra a edição dos cenários e a tela de utilização dos cenários. Para editar um cenário é preciso clicar no botão obstáculos, ilustrado na figura 25 e para jogar um cenário é só clicar no cenário na tela inicial ilustrada na figura 24. Para editar um cenário basta tocar no ator que pretende-se utilizar e tocar na posição desejada.

A figura 27, mostra a tela do joystick que pode ser acessado pelo menu. O joystick é utilizado para brincar com o robô e a cada comando pressionado é enviado ao robô na hora. A figura também mostra a consulta de alunos e as consultas das resoluções dos cenários dos alunos. Na consulta é possível saber o aluno, o cenário os comandos utilizados, a data e hora que foi realizada entre outras informações.

Figura 27 – Joystick e Consultas



4 APLICAÇÃO

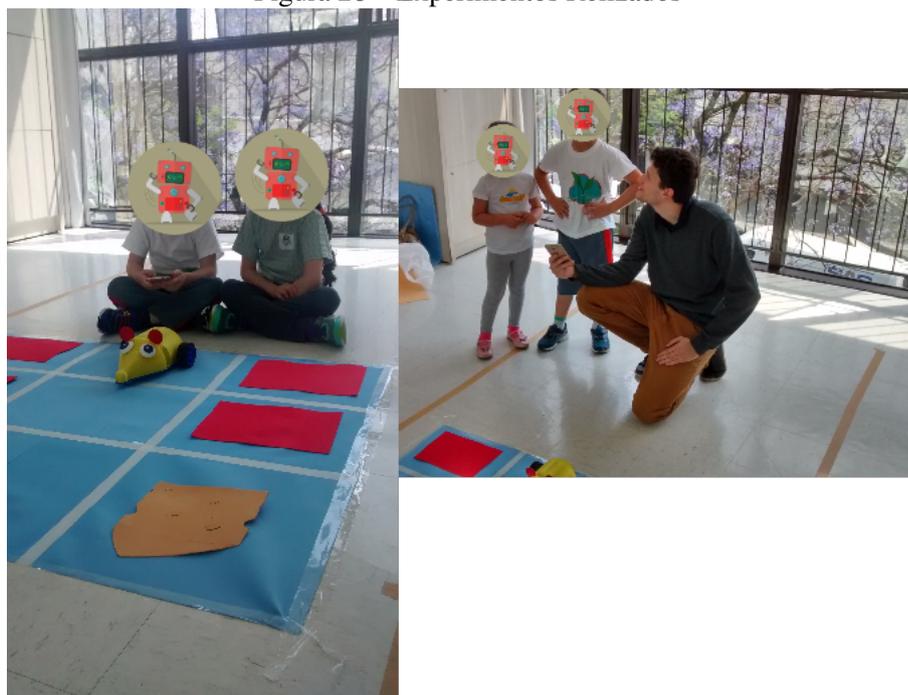
Neste capítulo é abordado a aplicação do projeto em sala de aula. É dividido em duas seções que são os experimentos realizados e materiais e métodos importantes para aplicação correta do projeto.

4.1 EXPERIMENTOS REALIZADOS

Durante os dias 18/10/17 e 25/10/17 foram realizados dois experimentos em uma escola de educação infantil. Os experimentos tiveram duração de quatro horas cada um. Foram analisados aspectos como a interação dos alunos com o projeto. E o tempo de duração de cada aluno na resolução dos cenários propostos.

Os cenários utilizados no experimento foram elaborados com a contribuição de Lucas Furstenu de Oliveira, professor de neurociência da UCS.

Figura 28 – Experimentos Realizados



O experimento contou com um total de 18 crianças. As crianças são divididas entre as turmas do pré e jardim. A turma do pré continha um total de 5 alunos, 3 meninos e 2 meninas. No jardim continham 13 crianças, 6 meninas e 7 meninos.

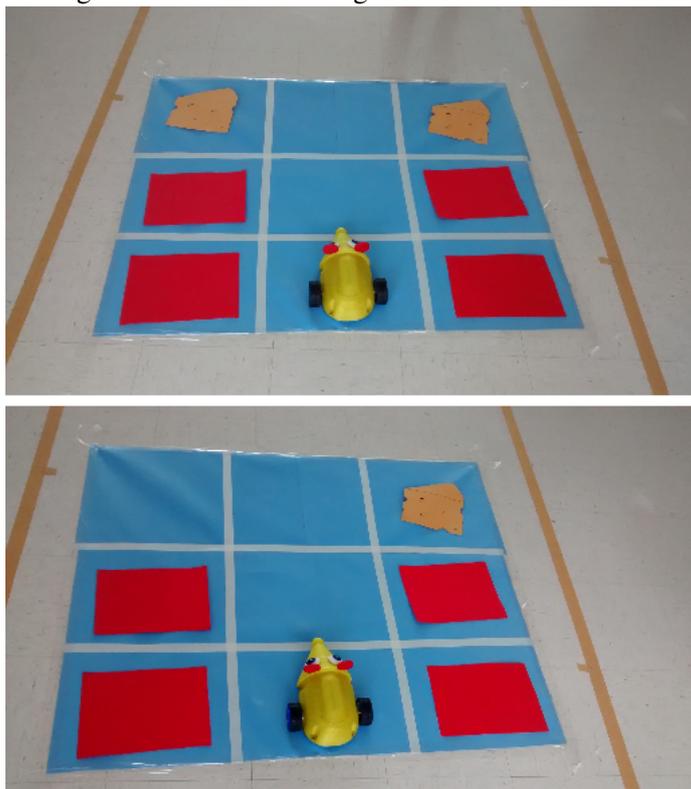
Nos dois experimentos realizados a metodologia utilizada foi a de dividir em grupos de dois ou três alunos, as crianças. Eles foram apresentados ao robô e ao aplicativo. O tempo de duração de cada grupo foi de aproximadamente 20 minutos. Logo no início já foi notória a empolgação dos alunos com o projeto.

No primeiro momento foi explicado para cada grupo o funcionamento do robô e como são

realizados os movimentos dele através do aplicativo. Foi utilizada a tela do joystick e cada aluno teve a oportunidade de enviar comandos ao robô por um intervalo de tempo.

Após o primeiro contato dos alunos com o projeto, foi apresentado a eles o primeiro cenário de utilização. Ilustrado na figura 29 na parte inferior. O cenário é simples e possui a solução também apresentada na imagem 29. Como primeiro desafio, as crianças atingiram os objetivos propostos em um tempo relativamente curto. Porém foram auxiliadas a todo instante.

Figura 29 – Primeiro e Segundo Cenários Utilizados

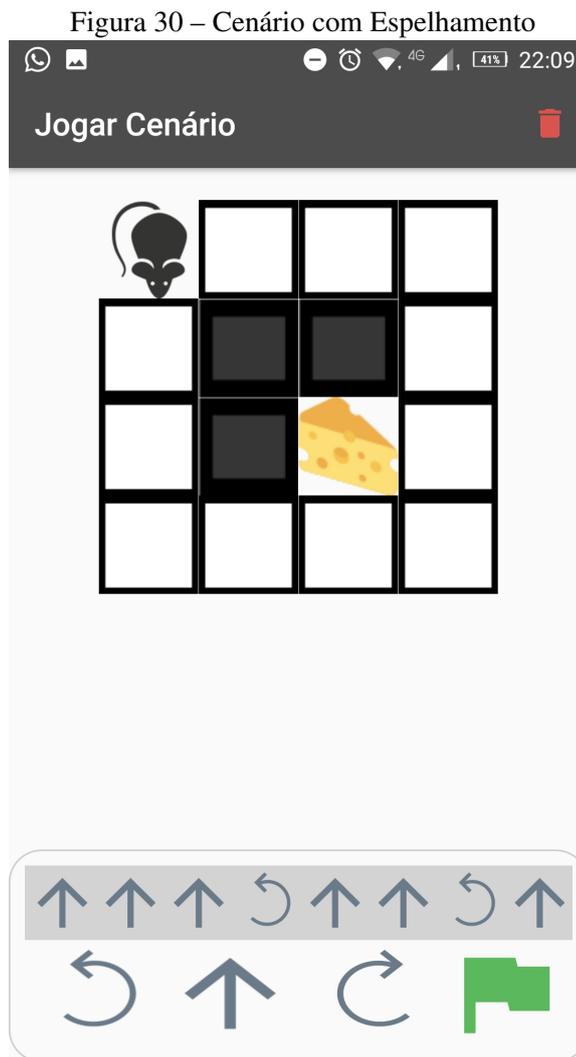


Para o segundo cenário de testes foi utilizado o cenário ilustrado na parte superior da figura 29. Esse cenário possui uma maior dificuldade por dois motivos. O primeiro ponto de dificuldade é que agora são 2 queijos para pegar. O segundo é o entendimento de ter que fazer o ratinho virar 180 graus para conseguir ficar de frente para o segundo queijo.

Ambos os cenários utilizados foram resolvidos pelos alunos, porém no segundo cenário foi necessário um acompanhamento maior pelo fator da maior dificuldade atrelada ao cenário.

No segundo dia de experimentos, após um primeiro dia onde não foram encontradas muitas dificuldades, foi trazida a ideia de montar um cenário complexo. Para isso foi montado o cenário da figura 30. A dificuldade desse cenário está em entender que o ratinho agora está virado ao contrário. Com isso seus movimentos são espelhados. Foi permitido aos alunos se moverem pelo cenário físico, a fim conseguirem resolver de maneira efetiva o que foi proposto. Nesse cenário como a abstração aumentou as crianças tiveram mais dificuldade na resolução.

Posterior a esse grande desafio pensou-se em mostrar novamente o segundo cenário desenvolvido no primeiro dia. E como era de se esperar as crianças tiveram menos dificuldade em



resolver esse mesmo cenário no segundo dia do experimento, inclusive resolvendo a tarefa mais rapidamente.

Como meio de incentivar os alunos foi dado a eles adesivos que mostravam que eles haviam resolvidos os desafios propostos.

Durante os experimentos foram constatadas algumas melhorias que foram implementadas no projeto. Com o objetivo de melhorar a interação dos alunos com o projeto.

A primeira mudança realizada no projeto foi a adição de imagens. No aplicativo foram adicionada imagens de um ratinho caindo e outra pensativo. Depois de alguns testes um aluno me perguntou: "Não acontece nada quando robô cai no buraco?". Daí surgiu a ideia.

A segunda alteração foi que não estava sendo enviado ao robô o comando para ele se chocar com a parede ou com os obstáculos. Com isso o robô parava sempre antes de fazer o comando ao qual faria isto acontecer. Após alguns testes os alunos ficavam sem entender direito o que ocorria de fato. O robô parava e não fazia nada. Com isso, foi adicionado os sons para facilitar o entendimento dos alunos com o robô. Também foi mandado para o robô o comando adicional que faz com que o robô se mova para o obstáculo ou para fora do cenário, com o objetivo de os

alunos visualizarem que realmente aconteceu um erro.

Como ponto positivo do trabalho ficaram pontos como:

- O layout utilizado pelos alunos. O layout do projeto ficou adequado para o entendimento dos alunos sobre os comandos disponíveis, deleção dos comandos e maneira de jogar.
- A edição dos cenários. Tivemos que em algumas oportunidades modificar os cenários com o objetivo de facilitar e dificultar os cenários para os alunos.
- Os sons e as imagens. Após a adição dos sons do robô e imagens ao término dos cenários, os alunos ficavam na expectativa de saber o que o robô ia falar, ou qual imagens iria aparecer. O que ajudou na abstração e interação dos alunos.
- A complexidade do projeto. A complexidade do projeto ficou certa com o público alvo, não tornando o projeto fácil demais e nem muito complicado. Tornando a complexidade algo maleável e que é possível aumentar ou diminuir com a edição dos cenários.
- Consulta dos dados realizados. Com a consulta dos dados foi possível ver os comandos utilizados por todos os alunos e com isso ficou de histórico e também para ver como os alunos resolviam cada problema.

Figura 31 – Tempos dos Alunos nos Cenários

Planilha de alunos	10/18/17			10/25/17		
	Tarefa 1	Tarefa 2	Observações sobre Tarefa 2	Tarefa 3	Tarefa 2	Observações sobre Tarefa 3
turma do pré						
Martina	00:01:11	00:00:38	Faltou 1 passo	00:05:36	00:02:23	mudou de posição
Enrico	00:02:05	00:04:30	duas tentativas	00:03:33	00:02:13	mudou diversas vezes de posição
Antonio	00:00:53	00:00:42	sem auxílio	00:05:44	00:00:41	mudou diversas vezes de posição
Jordana	00:01:02	00:02:47	duas tentativas	00:02:25	00:01:04	mudou de posição
Caetano	00:02:19	00:02:44		00:06:11	00:04:47	mudou de posição
turma do jardim						
Sofia	00:01:14	00:03:31			00:01:21	
Vinicius	00:01:02	00:03:19				
Alice	00:01:12	00:04:04			00:02:19	
Antônia	00:01:20	00:01:19				
Guilherme	00:01:03	00:02:45			00:01:00	
Maitê	00:01:39	00:01:52		00:04:03	00:04:47	não mudou de posição
Vitor	00:00:55	00:05:40		00:02:18	00:02:06	não mudou de posição
Eduarda	00:01:49	00:02:07		00:02:10	00:01:41	não mudou de posição
Germano	00:01:06	00:02:19				
Bruce	00:02:07	00:02:50				
Pedro	00:06:14	00:06:16				
Matheus+Dora	00:44:00					

A figura 31 mostra o tempo de cada um dos alunos nas tarefas realizadas. Mostra também algumas informações sobre a resolução dos cenários.

Ao olhar para a tabela de resultados, ficam alguns pontos importantes que é preciso notar:

- Pré e Jardim não tiveram largas diferenças em seus tempos. Mesmo com os alunos do pré sendo um pouco maiores seus tempos foram semelhantes aos alunos do jardim.

- Meninos e Meninas. Meninos e meninas tiveram seus tempos semelhantes.
- Notória melhora de uma semana para outra na tarefa 2. Após a primeira aula, na segunda oportunidade os alunos resolveram rapidamente a segunda tarefa. Está que no primeiro dia foi o grande desafio.
- O tempo da tarefa 3 mostra que é possível aumentar a dificuldade da aplicação apenas alterando a orientação do ratinho no cenário.

Ao término da experiência em sala de aula, ficou o sentimento de dever cumprido. O projeto foi bem recebido tanto pelas crianças como pela escola.

Por parte da escola o projeto ficou em aberto e voltaremos a ela para seguir com o projeto.

Os alunos ficaram imaginando qual seriam os próximos desafios e com isso formaram uma expectativa grande com o futuro do projeto.

4.2 MATERIAIS E MÉTODOS

O objetivo desta sessão é montar planos de utilização do aplicativo para uma futura utilização fixa em sala de aula. O plano de implantação do aplicativo passa por 3 aulas que foram projetadas e elaboradas com base nos experimentos realizados e pesquisas na área do ensino.

Notou-se principalmente que iniciar os estudos com os alunos partindo direto para o aplicativo, pode não ser o ideal. Para isso foi introduzida uma aula anterior a utilização do aplicativo. Portanto na aula 1 a ideia é apenas introduzir a ideia de programação, em forma de brincadeira, fazendo elas mesmos de robôs.

Tabela 3 – Plano de Aula 1

Objetivos	Conteúdo	Métodos/Estratégias	Recursos Didáticos	Avaliação
Inserir noções de lógica	Noções básicas de lógica aplicada a resolução de problemas	Criar em sala de aula um cenário de utilização físico, e utilizar os alunos como atores dos cenários	E.V.A. e fitas para confecção dos cenários de utilização	Serão avaliados a percepção e o engajamento dos alunos com a aula proposta

O plano de aula 1 tem por objetivo principal, a familiarização dos alunos com a lógica computacional. Através da resolução de cenários de ensino, de maneira física. Para isso os alunos serão os atores do cenário e seus colegas deverão orientá-los com os comandos do aplicativo até o objetivo final. O propósito é trazer para o mundo real o que será implementado no aplicativo posteriormente. De maneira lúdica, inserir o propósito do aplicativo para as crianças. A tabela 3, mostra o plano de aula da aula 1.

Tabela 4 – Plano de Aula 2

Objetivos	Conteúdo	Métodos/Estratégias	Recursos Didáticos	Avaliação
Utilização do aplicativo	Movimentos do aplicativo, movimentação do robô e resolução de problemas	Criar em sala de aula um cenário de utilização com a utilização do robô	E.V.A. e fitas para confecção dos cenários de utilização	Serão avaliados os alunos perante o conhecimento adquirido sobre o aplicativo e seu funcionamento

O objetivo principal da aula 2 é a familiarização dos alunos com o aplicativo e com o robô. Serão mostrados os primeiros passos do aplicativo, e também como fazer para o robô se movimentar. O propósito da aula 2 é fazer com que os alunos, orientados pelo conhecimento prévio da aula 1, identifiquem como resolver os cenários que serão montados na sala de aula. A tabela 4, mostra o plano de aula da aula 2.

Tabela 5 – Plano de Aula 3

Objetivos	Conteúdo	Métodos/Estratégias	Recursos Didáticos	Avaliação
Realização dos cenários de ensino	Cenários de utilização do aplicativo	Criar em sala de aula um cenário de utilização com a utilização do robô	E.V.A. e fitas para confecção dos cenários de utilização	Serão avaliados os alunos sobre o desenvolvimento dos cenários de utilização

A aula 3 será de desenvolvimento dos cenários de utilização, portando o objetivo dela é orientar e observar a evolução dos alunos nos cenários de utilização. A tabela 5, mostra o plano de aula da aula 3.

Após as 3 aulas concluídas é esperada principalmente o total entendimento dos alunos com o aplicativo e com os cenários de utilização, e também a evolução deles nos cenários de utilização propostos.

5 CONSIDERAÇÕES FINAIS

5.1 SÍNTESE

Para o presente trabalho foi realizado uma pesquisa sobre o software educacional, mostrando os principais tópicos relacionados ao assunto.

É notório que exista uma tendência de estudo e pesquisa nesta área, devido principalmente a procura no mercado de trabalho de profissionais nas áreas da tecnologia da informação e também ao fato de que o estudo da computação pode estimular habilidades como o pensamento computacional e o raciocínio lógico.

O pensamento computacional e raciocínio lógico, são habilidades importantes para qualquer pessoa no mundo atual, onde a tecnologia está cada vez mais presente no cotidiano das pessoas. Tais habilidades estão diretamente relacionadas à capacidade de organizar e resolver problemas, a capacidade de raciocinar, modularizar, e utilizar a lógica como ferramenta para o dia a dia.

Por outro lado, para a inserção da lógica computacional, principalmente na educação infantil, diversas foram as dificuldades encontradas. A falta de estruturação das escolas, a falta de capacitação dos professores, a dificuldade de atrair a atenção das crianças, a introdução de temas fundamentais como o lúdico e o concreto são os principais obstáculos para inserção da lógica computacional em sala de aula.

A utilização de softwares aliados a robóticas é fundamental para a inserção de conceitos de ciência da computação como um todo. Fundamentais, pois introduzem o lúdico na educação, trazem para o mundo real toda a abstração da área. É preciso pensar, que ensinar conceitos de ciência da computação é complicado até para alunos universitários. Autores como Moreira (2011), falam da importância do computador para a aprendizagem significativa. Como contraponto para a utilização de tecnologias em sala de aula, foram encontrados fatores relevantes como: o custo elevado para obtenção de tais equipamentos e a dificuldade que os professores sem capacitação podem encontrar com tais recursos.

Devido a esse custo elevado das ferramentas encontradas, e a falta de aplicações voltadas para crianças na educação infantil, foi proposto e elaborado um aplicativo, que com a utilização de robótica, tem por objetivo inserir os conceitos da ciência da computação, e com isso amenizar os problemas encontrados.

O aplicativo elaborado tem a função de dar suporte ao professor, com cenários de utilização maleáveis, que podem se ajustar ao que o professor desejar. Com o feedback da evolução dos alunos nos cenários de utilização. Os cenários de ensino são desafios que simulam um jogo, com o objetivo de cativar os alunos. E por fim a utilização de robótica educacional, que traz consigo o lúdico e o concreto, com o objetivo de facilitar o ensino para as crianças.

5.2 CONTRIBUIÇÕES ESPERADAS DO TRABALHO

Com este trabalho pretende-se desenvolver uma alternativa para inserção da lógica de programação para crianças. Um aplicativo que tenha a capacidade de oferecer ao professor a construção do conteúdo que ele achar conveniente. Que traga o feedback necessário para obtenção de informações sobre o desenvolvimento de seus alunos. Para os alunos espera-se que o aplicativo tenha capacidade de conquistá-los, que possam aprender se divertindo, que consigam interpretar a lógica por trás e com isso estimular seu raciocínio lógico, e desenvolver o pensamento computacional. Espera-se também que o aplicativo saia da área acadêmica e consiga se tornar um produto, agregando funcionalidades e ficando cada vez mais completo.

5.3 TRABALHOS FUTUROS

Existem algumas melhorias projetadas tanto no hardware como no software. A adição de uma rede de aparelhos onde cada aluno utilizaria o seu aparelho e o professor também teria o seu. Uma maior customização de imagens e sons do aplicativo, como a customização do robô para um gato, um cachorro ou qualquer coisa que a imaginação permitir. A utilização de sensores no robô, para testar proximidade, luz do ambiente. Enfim o projeto tem muito a crescer ainda.

Por outro lado se faz necessário também a ampliação dos testes e uma análise estatística dos resultados. Porém para isso é necessário que sejam realizados mais experimentos com o projeto.

REFERÊNCIAS

- ABELLÓN, M. **Professor**: as dificuldades para utilizar a tecnologia dentro da sala de aula das escolas públicas brasileiras. Disponível em: <<http://direcionalescolas.com.br/2015/08/06/professor-as-dificuldades-para-utilizar-a-tecnologia-dentro-da-sala-de-aula-das-escolas-publicas-brasileiras/>>. Acesso em: 17 junho 2017.
- AGUIAR, Y. de et al. Introdução à robótica e estímulo à lógica de programação no ensino básico utilizando o kit educativo lego® mindstorms. In: WORKSHOPS DO CONGRESSO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 2015. **Anais...** [S.l.: s.n.], 2015. p. 1418–1424. Disponível em: <<http://www.br-ie.org/pub/index.php/wcbie/article/view/6321>>. Acesso em: 10 março 2017.
- BEE-BOT. **Bee-bot support materials**. Disponível em: <<https://www.bee-bot.us>>. Acesso em: 15 março 2017.
- BLIKSTEIN, P. **O pensamento computacional e a reinvenção do computador na educação**. Disponível em: <<http://www.blikstein.com>>. Acesso em: 10 junho 2017.
- BOMBASAR, J. R. et al. Ferramentas para o ensino-aprendizagem do pensamento computacional: onde está Alan Turing? In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 2015. **Anais...** [S.l.: s.n.], 2015. Disponível em: <<http://br-ie.org/pub/index.php/sbie/article/view/5120/3525>>. Acesso em: 21 abril 2017.
- CAMBRUZZI, E.; SOUZA, R. M. de. Robótica educativa na aprendizagem de lógica de programação: aplicação e análise. In: XXI WORKSHOP DE INFORMÁTICA NA ESCOLA (WIE 2015), 2015. **Anais...** [S.l.: s.n.], 2015. Disponível em: <<http://www.br-ie.org/pub/index.php/wie/article/view/4981>>. Acesso em: 10 março 2017.
- CHAVES LIMA Árlon; SOUSA, D. F. Desenvolvimento do raciocínio lógico algoritmo na educação básica. In: II CONGRESSO NACIONAL DE EDUCAÇÃO, 2015. **Anais...** [S.l.: s.n.], 2015. Disponível em: <<http://www.br-ie.org/pub/index.php/wcbie/article/viewFile/6313/4422>>. Acesso em: 17 março 2017.
- CHEN, G. et al. Assessing elementary students' computational thinking in everyday reasoning and robotics programming. In: COMPUTERS & EDUCATION, 2017, Holanda. **Anais...** [S.l.: s.n.], 2017. p. 162–175. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0360131517300490>>. Acesso em: 15 março 2017.
- CODE.ORG. **Code**. Disponível em: <<https://code.org/>>. Acesso em: 10 junho 2017.
- CODESPARK. **Thefoos**. Disponível em: <<http://thefoos.com/>>. Acesso em: 13 maio 2017.
- CODIE. **Codie - cute personal robot that makes coding fun**. Disponível em: <<https://www.indiegogo.com/projects/codie-cute-personal-robot-that-makes-coding-fun>>. Acesso em: 17 março 2017.

COSTA, D. P.; SANTOS, M. T. P. Comparativo entre gerenciadores de banco de dados para aplicação android. **Revista TIS**, São Carlos, p. 20–30, 2015. Disponível em: <<http://revistatis.dc.ufscar.br/index.php/revista/article/view/296/96>>. Acesso em: 24 maio 2017.

CSTA, C. S. T. A. **Csta k-12 computer science standards**. Disponível em: <<https://goo.gl/dtn7NS>>. Acesso em: 16 junho 2017.

FERREIRA, L. A. C. et al. Se-robô: aplicativo para robótica educacional de baixo custo. In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 2016, Passo Fundo. **Anais...** [S.l.: s.n.], 2016. p. 846–855. Disponível em: <<http://www.br-ie.org/pub/index.php/sbie/article/view/6814/4699>>. Acesso em: 17 março 2017.

FIALHO, N. N. Os jogos pedagógicos como ferramentas de ensino. In: CONGRESSO NACIONAL DE EDUCAÇÃO, 2008. **Anais...** [S.l.: s.n.], 2008. Disponível em: <<http://quimimoreira.net/Jogosmarço>> 2017.

FORBELLONE, A. L. V.; EBERSPÄCHER, H. F. **Lógica de programação**. [S.l.]: Pearson, 2005.

FRANÇA, R. S. de; AMARAL, H. J. C. do. **Proposta metodológica de ensino e avaliação para o desenvolvimento do pensamento computacional com o uso do scratch**. Disponível em: <<http://br-ie.org/pub/index.php/wie/article/view/2646/2300>>. Acesso em: 17 março 2017.

GOMES, T. C. S.; A. R. TEDESCO, P. C. de; MELO, J. C. B. de. Jogos no design de experiências de aprendizagem de programação engajadoras. In: JORNADA DE ATUALIZAÇÃO EM INFORMÁTICA NA EDUCAÇÃO, 2016, Uberlândia. **Anais...** [S.l.: s.n.], 2016. Disponível em: <<http://www.br-ie.org/pub/index.php/pie/article/view/6595/4506>>. Acesso em: 17 março 2017.

LEGO. **Legó mindstorms**. Disponível em: <www.lego.com>. Acesso em: 17 março 2017.

LIGHTBOT. **Lightbot**. Disponível em: <<https://www.lightbot.com>>. Acesso em: 13 maio 2017.

MAISONNETTE, R. A utilização dos recursos informatizados a partir de uma relação inventiva com a máquina: a robótica educativa. In: PROINFO - PROGRAMA NACIONAL DE INFORMÁTICA NA EDUCAÇÃO, 2002, Paraná. **Anais...** [S.l.: s.n.], 2002.

MATTEI, C. O prazer de aprender com a informática na educação infantil. In: INSTITUTO CATARINENSE DE PÓS-GRADUAÇÃO, 2003. **Anais...** [S.l.: s.n.], 2003. Disponível em: <<http://www.posuniasselvi.com.br/artigos/rev02-11.pdf>>. Acesso em: 17 junho 2017.

MINISTÉRIO DA EDUCAÇÃO. **Diretrizes curriculares nacionais para a educação infantil**. Brasília, 2010. Disponível em: <<http://ndi.ufsc.br/files/2012/02/Diretrizes-Curriculares-para-a-E-I.pdf>>. Acesso em: 17 jun. 2017.

MOREIRA, M. A. Aprendizagem significativa, campos conceituais e pedagogia da autonomia: implicações para o ensino. **Instituto de Física - UFRGS**, Porto Alegre, RS, Brasil, p. 1–29, 2011.

MUKHINA, V. **Psicologia da idade pré-escolar**. [S.l.]: Martins Fontes, 1996.

NUNES, D. J. **Ciência da computação na educação básica**. Disponível em: <<http://www.adufrgs.org.br/artigos/ciencia-da-computacao-na-educacao-basica/>>. Acesso em: 17 março 2017.

PERALTA, D. **Projeto programando e aprendendo na educação infantil**. 2015.

RAABE, A. L. A.; SILVA, J. M. C. da. Um ambiente para atendimento as dificuldades de aprendizagem de algoritmos. In: XXV CONGRESSO DA SOCIEDADE BRASILEIRA DE COMPUTAÇÃO E XIII WORKSHOP SOBRE INFORMÁTICA NA ESCOLA, 2005.

Anais... [S.l.: s.n.], 2005. p. 2326–2337. Disponível em: <<http://www.lbd.dcc.ufmg.br/colecoes/wei/2005/003.pdf>>. Acesso em: 17 junho 2017.

RICHTER, D. R. et al. Design de um brinquedo programável para crianças de 4 a 5 anos de idade através da metodologia design thinking. In: ESCOLA REGIONAL DE INFORMÁTICA DA SOCIEDADE BRASILEIRA DE COMPUTAÇÃO (SBC), 2016, Itajaí.

Anais... [S.l.: s.n.], 2016. Disponível em: <<http://anaiserimt.ic.ufmt.br/index.php/erimt/article/view/58/80>>. Acesso em: 10 março 2017.

ROBODASH. **Dash is a child's first real robot friend**. Disponível em: <<https://www.makewonder.com/dash>>. Acesso em: 17 março 2017.

ROCIO ZILLI, S. do. **A robótica educacional no ensino fundamental: perspectivas e prática**. 2004. Dissertação (Mestrado em Ciência da Computação) — Universidade Federal de Santa Catarina, 2004.

SANTOS, S. M. P. dos. **O lúdico na formação do educador**. [S.l.]: Petrópolis, 1999.

SCRATCH. **Scratch**. Disponível em: <<https://scratch.mit.edu/>>. Acesso em: 13 maio 2017.

SICA, C. **Ciência da computação no ensino básico e médio**. Disponível em: <<http://blogs.odiariorio.com/carlossica/2011/10/07/ciencia-da-computacao-no-ensino-medio/>>. Acesso em: 10 junho 2017.

SILVA, D. P. da et al. Aplicação de robótica na educação de forma gradual para o estímulo do pensamento computacional. In: WORKSHOPS DO CONGRESSO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 2016, Minas Gerais. **Anais...** [S.l.: s.n.], 2016. p. 1188–1197. Disponível em: <<http://br-ie.org/pub/index.php/wcbie/article/view/7044>>. Acesso em: 10 março 2017.

SILVA, E. G. da; LOPES, R.; RODRIGUES, A. Análise de ferramentas para o ensino de computação na educação básica. In: CONGRESSO DA SOCIEDADE BRASILEIRA DE COMPUTAÇÃO – CSBC 2014, 2014, Pernambuco. **Anais...** [S.l.: s.n.], 2014.

SILVA, V.; SOUZA, A.; MORAIS, D. Pensamento computacional: um relato de práticas pedagógicas para o ensino de computação em escolas públicas. **Revista Tecnologias na**

Educação, [S.l.], 2016. Disponível em: <<http://tecedu.pro.br/wp-content/uploads/2016/09/Art5-Pensamento-computacional-Um-relato-de-pr>>. Acesso em: 16 junho 2017.

SOARES, N. Crise? não no mercado de tecnologia da informação. **Estadão**, [S.l.], 2016. Disponível em: <<http://educacao.estadao.com.br/noticias/geral,crise-nao-no-mercado-de-tecnologia-da-informacao,10000023666>>. Acesso em: 15 junho 2017.

VALENTE, J. A. **O computador na sociedade do conhecimento**. [S.l.]: nied, 1999.

VYGOTSKY, L. S. **A formação social da mente**. [S.l.]: Martins Fontes, 1984.

WING, J. M. Computational Thinking. **Communications of the ACM**, [S.l.], v. 49, n. 3, p. 33–35, 2006. Disponível em: <<https://www.cs.cmu.edu/15110-s13/Wing06-ct.pdf>>. Acesso em: 21 abril 2017.