

Lucas Pinho Badaró

# Cubol: Uma Aplicação Móvel com Realidade Aumentada para auxiliar o desenvolvimento do Pensamento Computacional

Vitória da Conquista/BA

2019

Lucas Pinho Badaró

# **Cubol: Uma Aplicação Móvel com Realidade Aumentada para auxiliar o desenvolvimento do Pensamento Computacional**

Trabalho de Conclusão de Curso apresentado como requisito para obtenção do título de Bacharel em Ciência da Computação na Universidade Estadual do Sudoeste da Bahia – UESB.

Universidade Estadual do Sudoeste da Bahia – UESB

Curso de Bacharelado em Ciência da Computação

Departamento de Ciências Exatas e Tecnológicas

Orientador: Profa. Dra. Cátia Mesquita Brasil Khouri

Vitória da Conquista/BA

2019

Lucas Pinho Badaró

Cubol: Uma Aplicação Móvel com Realidade Aumentada para auxiliar o desenvolvimento do Pensamento Computacional/ Lucas Pinho Badaró. – Vitória da Conquista/BA, 2019-

45 p. : il. (algumas color.) ; 30 cm.

Orientador: Profa. Dra. Cátia Mesquita Brasil Khouri

Trabalho de Conclusão de Curso (Graduação)  
Universidade Estadual do Sudoeste da Bahia – UESB  
Curso de Bacharelado em Ciência da Computação  
Departamento de Ciências Exatas e Tecnológicas, 2019.

1. Pensamento Computacional. 2. Ludicidade. 3. Realidade Aumentada. I. Profa. Dra. Cátia Mesquita Brasil Khouri. II. Universidade Estadual do Sudoeste da Bahia. III. Curso de Ciência da Computação. IV. Cubol: Uma aplicação móvel com realidade aumentada para auxiliar desenvolvimento do pensamento computacional.

Lucas Pinho Badaró

# **Cubol: Uma Aplicação Móvel com Realidade Aumentada para auxiliar o desenvolvimento do Pensamento Computacional**

Trabalho de Conclusão de Curso apresentado como requisito para obtenção do título de Bacharel em Ciência da Computação na Universidade Estadual do Sudoeste da Bahia – UESB.

Trabalho aprovado. Vitória da Conquista/BA, 28 de agosto de 2019:

---

**Prof. Dra. Cátia Mesquita Brasil  
Khoury**  
Orientadora

---

**Prof. MSc. Gidevaldo Novais dos  
Santos**  
Convidado 1

---

**Prof. Dr. Fábio Moura Pereira**  
Convidado 2

Vitória da Conquista/BA  
2019

*À minha mãe, por tudo.*

# Agradecimentos

Agradeço primeiramente a minha mãe, Wânia, que foi o motivo de eu ter persistido até o final do curso. Ela que me criou, que proveu tudo o que eu precisei e que sempre fez o papel tanto de mãe quanto de pai para comigo. Tudo que sou hoje, devo a ela.

Agradeço aos meus avós, Walter e Sônia, por serem minha bússola moral, e serem a minha inspiração como seres humanos.

Agradeço aos meus tios e tias, por todo amparo emocional e financeiro, em especial Wanessa e Deny por me abrigarem durante os tempos de ensino médio, e a Luciano e Iva pelo suporte quando cheguei em minha nova morada.

Agradeço aos meus irmãos, Anna e Mateus, por terem me ensinado o significado de família e pela contribuição ao meu amadurecimento.

Agradeço a minha namorada Flávia, por ter me apoiado em tudo que podia e pela paciência de ter ficado do meu lado a todo momento.

Agradeço a minha orientadora Profa. Dra. Cátia Mesquita Brasil Khouri e a Celina Pereira, pelo empenho dedicado ao meu trabalho de conclusão de curso, sem elas com certeza não teria chegado até o final.

E por fim, mas não menos importante, agradeço a todos os meus colegas e amigos, que de alguma forma contribuíram para que eu concluísse esse ciclo, especialmente ao meu parceiro de NTI, Rodrigo, por toda a ajuda prestada.

# Resumo

Com o passar dos anos o modelo pedagógico vigente se tornou ultrapassado, abrindo espaço para novas maneiras de se pensar a aplicação do conteúdo educacional. Essas linhas de pensamento reforçam: o controle da aprendizagem pelo aluno; o estabelecimento do elo emocional entre sujeito e objeto; e o aprendizado imersivo de forma não traumática. Para tanto, a utilização da tecnologia de Realidade Aumentada como instrumento educacional, permite a aplicação destas práticas. Esse trabalho de conclusão de curso demonstra o desenvolvimento do Cubol, um protótipo de aplicação móvel voltado para o desenvolvimento do Pensamento Computacional para alunos recém-ingressos em cursos de Ciência da Computação. O protótipo apresentado, utiliza técnicas de Realidade Aumentada implementadas com suporte do kit de desenvolvimento Google ARCore e é executado sobre o sistema operacional Android. As escolhas de projeto foram feitas com base no seu baixo custo, propício para metodologia de prototipação. Além disso foi utilizado o motor de jogos Unity para auxiliar o desenvolvimento. Ademais, abstrações construtivistas, construcionistas e lúdicas deram à aplicação ferramentas para construir uma aprendizagem intuitiva dos algoritmos propostos, sem a necessidade de um tutorial ou explicação prévia. Através do resultado da questão proposta, foi atingido o objetivo de implementar um artefato, que permite uma primeira experiência com lógica de programação na qual o usuário possa se divertir aprendendo.

**Palavras-chave:** Pensamento Computacional, Ludicidade, Realidade Aumentada.

# Abstract

Over the years the current pedagogical model has become outdated, opening space for new ways of thinking about the application of educational content. These lines of thought reinforce: student control of learning; the establishment of the emotional link between subject and object; and non-traumatic immersive learning. Therefore, the use of Augmented Reality technology as an educational instrument allows the application of these practices. This course completion paper demonstrates the development of Cubol, a mobile application prototype focused on the development of Computational Thinking for students new to Computer Science courses. The prototype presented uses Augmented Reality techniques implemented with support from the Google ARCore development kit and runs on the Android operating system. The design choices were made on the basis of low cost, suitable for prototyping methodology. In addition, the Unity game engine was used to aid development. In addition, constructivist, constructionist and playful abstractions gave the application tools to construct an intuitive learning of the proposed algorithms, without the need for a tutorial or previous explanation. with programming logic in which the user can have fun learning.

**Keywords:** Computational Thinking, Playfulness, Augmented Reality.

# Lista de ilustrações

Figura 1 – Scratch . . . . .	17
Figura 2 – Blockly . . . . .	18
Figura 3 – Code.org . . . . .	19
Figura 4 – Realidades Mistas de Kirner . . . . .	20
Figura 5 – Head-Mounted Display . . . . .	21
Figura 6 – Notable Women . . . . .	23
Figura 7 – Protótipo sobre relação entre terra e sol em RA . . . . .	23
Figura 8 – Design Interativo de Zimmerman . . . . .	26
Figura 9 – Mapa . . . . .	28
Figura 10 – Diagrama de Caso de Uso . . . . .	30
Figura 11 – Diagrama de Estado . . . . .	31
Figura 12 – Diagrama de Sequência . . . . .	31
Figura 13 – Cubol . . . . .	32
Figura 14 – Primeiro Protótipo . . . . .	35
Figura 15 – Segundo Protótipo . . . . .	36
Figura 16 – Terceiro Protótipo . . . . .	37
Figura 17 – Protótipo Final . . . . .	38
Figura 18 – Testes . . . . .	39

# Lista de abreviaturas e siglas

API: *Application Programming Interface*

HMD: *Head-Mounted Displays*

LLK: *Lifelong Kindergarten*

MIT: *Massachusetts Institute of Technology*

QP: *Questão de Pesquisa*

RA: *Realidade Aumentada*

RC: *Realidade Cruzada*

RM: *Realidades Mistas*

RV: *Realidade Virtual*

SDK: *Software Development Kit*

# Sumário

<b>1</b>	<b>INTRODUÇÃO</b>	<b>12</b>
<b>2</b>	<b>REFERENCIAL TEÓRICO</b>	<b>15</b>
<b>2.1</b>	<b>Conceitos Pedagógicos</b>	<b>15</b>
2.1.0.1	Scratch	17
2.1.0.2	Blockly	18
2.1.0.3	Code.org	18
2.1.1	Pensamento Computacional	19
<b>2.2</b>	<b>Realidades Mistas</b>	<b>20</b>
2.2.1	Realidade Virtual	21
2.2.2	Realidade Aumentada	22
2.2.3	Realidade Cruzada	24
<b>2.3</b>	<b>Plataformas e Ferramentas</b>	<b>24</b>
2.3.1	Android	24
2.3.2	Unity e Motores de Jogos	25
2.3.3	ARCore	25
<b>3</b>	<b>CUBOL</b>	<b>26</b>
<b>3.1</b>	<b>Metodologias</b>	<b>26</b>
<b>3.2</b>	<b>Propósito do Projeto</b>	<b>27</b>
<b>3.3</b>	<b>Características Gerais do Jogo</b>	<b>27</b>
3.3.1	Mapa	27
3.3.2	Personagem	28
3.3.3	Percurso	28
3.3.4	Objetivo	29
3.3.5	Comandos	29
<b>3.4</b>	<b>Especificação dos Requisitos</b>	<b>29</b>
3.4.1	Requisitos Funcionais	29
3.4.2	Requisitos Não Funcionais	29
3.4.3	Atores do Protótipo	30
3.4.4	Casos de Uso	30
3.4.5	Estados do Protótipo	30
3.4.6	Sequência de Processos	31
<b>3.5</b>	<b>Características da Aplicação</b>	<b>32</b>
<b>3.6</b>	<b>Realidade Aumenta Aplicada ao Jogo</b>	<b>33</b>
3.6.1	Interação	33

3.6.2	Animação . . . . .	33
3.6.3	Poder Imersivo . . . . .	33
<b>3.7</b>	<b>Desenvolvimento . . . . .</b>	<b>33</b>
3.7.1	<i>Game Design Document</i> . . . . .	33
3.7.2	História . . . . .	34
3.7.3	Modelagem . . . . .	34
3.7.4	Textura . . . . .	34
3.7.5	Colisões . . . . .	34
3.7.6	Animações . . . . .	34
3.7.7	Inserção na Realidade Aumentada . . . . .	34
3.7.8	Compilação e <i>Building</i> . . . . .	35
3.7.9	Protótipos . . . . .	35
3.7.9.1	Primeiro Protótipo . . . . .	35
3.7.9.2	Segundo Protótipo . . . . .	36
3.7.9.3	Terceiro Protótipo . . . . .	36
3.7.9.4	Protótipo Final . . . . .	37
3.7.10	Testes . . . . .	39
<b>4</b>	<b>CONSIDERAÇÕES FINAIS . . . . .</b>	<b>40</b>
4.1	Dificuldades do Projeto . . . . .	41
4.2	Trabalhos Futuros . . . . .	41
	<b>REFERÊNCIAS . . . . .</b>	<b>42</b>

# 1 Introdução

Desde a Grécia antiga, quando surgiram as primeiras teorias educacionais, vários modelos de ensino foram criados e desenvolvidos. No decorrer dos anos, as civilizações mudaram e começaram a adotar novos meios de aprendizagem. Hoje, a sociedade como um todo adota o modelo de escola tradicional como principal concepção pedagógica. Neste modelo, temos a figura do professor como centro de transmissão do conhecimento e centro de verdade absoluta (PAPERT; HAREL, 1991).

Entretanto, com a evolução do pensamento crítico, da filosofia e principalmente da tecnologia, esse modelo então vigente, passou a ser alvo de críticas. Deste modo, foram surgindo cada vez mais novas teses e teorias (BECKER, 1992; PAPERT, 1986; LUCKESI, 2000) que questionam o arquétipo defasado, seguido pela maioria.

Dentre as teorias que surgiram, se destacam Construtivismo (BECKER, 1992) e Construcionismo (PAPERT, 1986), ambas trazendo novas sugestões de abordagens com as quais o conhecimento deveria ser disseminado. Seja defendendo o papel ativo do sujeito (aluno) na criação e modificação do objeto do conhecimento, seja na construção do próprio conhecimento através de uma ação concreta feita com um computador, estas novas linhas de pensamento acabaram por mudar o *status quo* da pedagogia.

Seguindo a mesma linha de pensamento, a Ludicidade foi introduzida como um componente muito importante para a aprendizagem. Na pedagogia, ela se dá como a forma de desenvolver a criatividade e os conhecimentos por meio de jogos, fazendo com que o aluno, ao brincar, fique exposto a um ambiente extremamente favorável ao desenvolvimento físico e cognitivo. Com base nesta imersão, o sujeito acaba por assimilar o conteúdo de maneira orgânica (LUCKESI, 2000).

Uma forma imersiva de aplicar conteúdo educacional, é o emprego de Realidade Aumentada em ambientes educacionais, contribuindo de maneira significativa na percepção, interação e motivação dos usuários (KIRNER; ZORZAL, 2005). RA é uma experiência interativa, na qual objetos que residem no mundo real sofrem modificações por informação perceptiva criada por computadores, trazendo uma experiência imersiva única para o usuário (KIRNER; CERQUEIRA; KIRNER, 2012).

Existe um consenso em apontar para o grande auxílio da tecnologia de RA, em relação à manutenção do interesse e incremento da motivação do aluno para com o assunto estudado (PINHO, 1996). Além desses fatores, segundo Pantelidis (1995), o maior poder de compreensão pelos estudantes com o uso de tecnologias de Realidades Mistas <sup>1</sup> em comparação com outras mídias, é devido ao fato da tecnologia permitir mais

---

<sup>1</sup> Realidade Mista, como o próprio nome já diz, é a lacuna de simulações entre elementos virtuais e a

possibilidades de experiências e ao fato de dar ao educando a possibilidade de desenvolver seu conhecimento com base em seu próprio ritmo.

A RA proporciona ao usuário uma interação agradável e efetiva, sem a necessidade de treinamento, por trazer para o ambiente real os elementos virtuais, aumentando a visão que ele tem do mundo real (ZORZAL et al., 2006). A fim de que esse ambiente possa ser possível, são necessárias técnicas de visão computacional, computação gráfica e Realidade Virtual, gerando assim a correta sobreposição de objetos virtuais no ambiente real (AZUMA, 1993).

Portanto, a RA é uma tecnologia em expansão, com grande potencial de exploração, e que pode contribuir de maneira significativa na área da educação. Ela garante uma grande capacidade de criação de jogos, e permite uma interação natural de fácil adaptação, livre de dispositivos auxiliares (ZORZAL et al., 2006). Entretanto, mesmo com o grande potencial de desenvolvimento, os jogos educativos com RA têm sido pouco explorados (ZORZAL et al., 2006).

As abordagens educacionais supracitadas vêm sendo aplicadas, nos últimos anos, para o desenvolvimento do *pensamento computacional* e do raciocínio lógico na aprendizagem de programação. O conceito de pensamento computacional vem sendo discutido nos últimos anos e refere-se a uma visão de mundo em termos de uma série de problemas que possuem soluções computacionais. A ideia é levar os alunos a pensar nos problemas de forma analítica e desenvolver soluções em forma de algoritmos (EASTERBROOK, 2014).

O pensamento computacional deve ser desenvolvido por estudantes do curso de Ciência da Computação, que devem ser capazes de elaborar formas efetivas de solucionar diversos tipos de problemas computacionais. Para isto, precisam utilizar os conhecimentos essenciais na identificação, validação e resolução de problemas, de maneira efetiva e eficiente (HERNANDEZ et al., 2010).

Similarmente, Allan e Kolesar (1996) dizem que o passo que precede o aprendizado de algoritmos e técnicas de desenvolvimento de software, é o desenvolvimento de estruturas cognitivas que levem à resolução de problemas gerais utilizando ferramentas computacionais. Para MUSSER e SHAUGHNESSY (1997), existe uma linha de estratégias para a resolução destes problemas: tentativa e erro, uso de padrões, resolver um problema mais simples, trabalhar em sentido inverso, e simulação.

A fim de alcançar os resultados esperados na formação destes estudantes, manifesta-se adequada a utilização de uma ferramenta voltada especificamente a promover o desenvolvimento das estruturas cognitivas supracitadas, ao mesmo tempo em que instrumenta o aluno com técnicas de resolução de problemas gerais (HERNANDEZ et al., 2010). A experiência de desenvolvimento do pensamento computacional deveria ser não traumática,

---

realidade, definido por Milgram e Kishino (1994) como a mescla de mundos reais.

principalmente no primeiro contato com a disciplina. Para isto, o processo didático-pedagógico deve ser simples e imersivo. Um exemplo de como trazer imersão a uma aplicação é trazendo características de Realidade Aumentada para a mesma (KIRNER; CERQUEIRA; KIRNER, 2012).

Alguns livros didáticos já a adotaram na forma como se apresentam, através de aplicativos que permitem ampliar o conhecimento por meio desta tecnologia (SHELTON; HEDLEY, 2002). O resultado da aplicação é a criação de um novo ambiente de aprendizado, que torna o estudo mais divertido e interessante. Com isto, os alunos se engajam mais nos estudos, além disso, memorizam e entendem melhor o conteúdo proposto. Por exemplo, o **livro aumentado** proposto por Shelton e Hedley (2002) explica a relação da Terra com o Sol em termos de inclinação axial e solstícios para alunos graduação.

Levando em conta a perspectiva apresentada, para este trabalho foi definida a questão de pesquisa: *Como aplicar conceitos construtivistas e construcionistas utilizando Realidade Aumentada de forma lúdica para fomentar o interesse de estudantes iniciantes em linguagem de programação?*

Buscando responder à questão colocada, o trabalho tem como objetivo projetar e implementar um protótipo de uma aplicação que favoreça a aprendizagem orgânica do Pensamento Computacional, além de objetivamente atrair a atenção do aprendiz através da imersão. Para a construção do protótipo serão aplicados tanto os conceitos pedagógicos de ensino como o Construcionismo, Construtivismo e Ludicidade; quanto táticas de RA. Deste modo, espera-se aprimorar o modo como se aprende linguagem de programação, aproveitando os novos conceitos e tecnologias que estão disponíveis.

Na trajetória rumo ao objetivo, buscou-se compreender e aplicar o estado da arte de RA; implementar e documentar um protótipo de uma aplicação lúdica em RA; e analisar o resultado do protótipo final como uma ferramenta educacional.

## 2 Referencial Teórico

Os modelos educacionais sofreram drásticas mudanças no decorrer do século passado, porém, só nas últimas décadas vemos maior força e abrangência do tradicionalismo educacional (PATTO, 1993). Desde então, este modelo pedagógico permanece presente. O modelo de educação tradicional é constituído por escolas organizadas em forma de classes e professores razoavelmente bem preparados, que aplicam exercícios realizados disciplinadamente pelos alunos (SAVIANI, 2018).

Com o passar dos anos, diversos esforços vêm sendo empreendidos no sentido de mudar a realidade da educação tradicional e neste capítulo serão citados alguns destes.

### 2.1 Conceitos Pedagógicos

Em paralelo à evolução dos métodos tradicionais de ensino, surgiu o *Construtivismo*, uma postura diferente em relação à aquisição do conhecimento. Esta postura se baseia na ideia de que nada está pronto, deste modo a própria interação do indivíduo com o meio físico e social, pode alterar tanto o meio, quanto o sujeito (BECKER, 1992).

O Construtivismo é apresentado por Piaget, que coloca o professor em uma posição mediadora em um processo. Ao invés de passar o conteúdo expositivamente, o professor organiza o trabalho didático-pedagógico para o aluno, que por sua vez assume o controle da aprendizagem (SPONHOLZ, 2003). Nesta visão interacionista do ensino, a ação parte do sujeito para o meio.

No *Construcionismo*, introduzido por Papert (1986), é apresentada uma abordagem segundo a qual o indivíduo aprende fazendo, e ao fazê-lo, mantém-se motivado pois está envolvido afetivamente com o assunto e o meio, tornando a absorção do conteúdo mais significativa. Esta abordagem tem semelhanças com o Construtivismo mas considera, adicionalmente, a conexão emocional que é mantida entre o sujeito e o meio. O Construcionismo atualmente é aplicado pelo Massachusetts Institute of Technology (MIT) no projeto Lifelong Kindergarten (KINDERGARTEN, 2019), que tem desenvolvido novas tecnologias, atividades e estratégias para engajar jovens a aprender criativamente. A abordagem é chamada por eles de "4 Ps" (RESNICK, 2014), que são justificados pelos tópicos a seguir:

- **Project (Projeto)** - Pessoas aprendem melhor quando estão trabalhando em projetos que façam sentido;
- **Peers (Pares)** - O aprendizado floresce como uma atividade social, com pessoas compartilhando ideias, colaborando em projetos e construindo o trabalho uns dos

outros;

- **Passion (Paixão)** - Quando as pessoas trabalham em projetos com os quais se importam, acabam por aprender mais no processo;
- **Play (Ludicidade)** - Aprendizagem envolve experimentação lúdica, tentar coisas novas, mexer com materiais, testar limites, correr riscos, repetir de novo e de novo.

Por fim, para o Construcionismo, os computadores são extremamente proeminentes para a aprendizagem, porque eles podem prover um largo alcance de excelente contexto para o aprendizado. A presença de computadores altera a natureza do processo, e têm o poder de desbalancear o equilíbrio da transferência de conhecimento, tirando do sujeito centralizador, o centro de verdade absoluta (PAPERT; HAREL, 1991).

Nas últimas décadas, vários estudiosos vêm defendendo a aplicação de atividades lúdicas no ensino, tanto de crianças, quanto de jovens ou adultos (LUCKESI, 2000; FORTUNA, 2011; VERISSIMO; SANTOS, ; SANTOS, 2001). Além disso, segundo Gomes (2008), "o brincar é próprio do ser humano". Portanto, brincadeiras e jogos vêm sendo objeto de interesse da filosofia, antropologia, psicologia, pedagogia e neurociências ao longo dos anos.

Para Luckesi LUCKESI (2000), a *Ludicidade* “é representada por atividades que propiciam experiência de plenitude e envolvimento por inteiro, dentro de padrões flexíveis e saudáveis”. Além disso, ele afirma que essa experiência vai além dos limites do ego e, que "o ato de brincar não só é revelador do inconsciente, ele também é catártico, ou seja, ele é libertador” (LUCKESI, 2005). Isto porque, quando se brinca, são expressados e liberados os conteúdos do inconsciente, procurando a restauração de possibilidades de vida saudável, livre dos bloqueios impeditivos.

Seguindo a linha de pensamento de atividade lúdica, o pesquisador Roger Schank tem desenvolvido softwares educativos com metodologias que se preocupam em como o aluno constrói o seu conhecimento e com a sua motivação. Ele desenvolveu as arquiteturas de aprendizado, que fazem com que o aprendizado seja natural e se adapte na busca de estimular diferentes maneiras de como aprender por si só (SCHANK, 1997). Além disso, ressalta também a importância de se aprender com simulação através do computador, visto que por intermédio desta experiência a aprendizagem pode ser alegre e construída através de erros, de forma não traumática.

São apresentadas por Schank (1997), diversas vantagens da aprendizagem através de jogos computacionais, seriam elas: aprender com os erros, a não humilhação pelo erro, experiência personalizada, motivação e a não memorização. O professor passa a ser um facilitador, os estudantes trabalham de forma independente com os computadores, e portanto trazendo interação entre os próprios estudantes dentro do grupo.

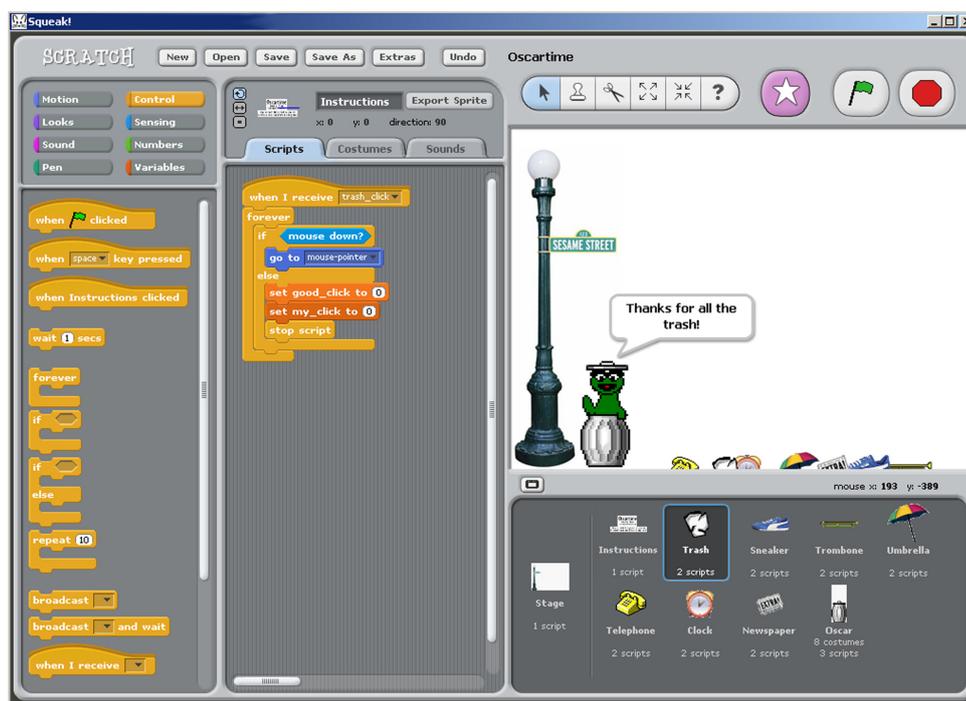
A era da informação trouxe uma enxurrada de mudanças na transmissão de conhecimento, bem como o avanço e a massificação da tecnologia. Métodos inovadores e ferramentas lúdicas tornam a aprendizagem mais acessível, como por exemplo, Scratch, Blockly e Code.org, os quais auxiliam de forma criativa o desenvolvimento do PC, sem que haja distinção de idade, classe social ou intelectual, apresentados a seguir.

### 2.1.0.1 Scratch

Desenvolvido pelo grupo de pesquisa Lifelong Kindergarten no Laboratório de Mídias do MIT, Scratch é um ambiente de programação multimídia que encoraja estudantes de primeira viagem a implementar animações, jogos e arte interativa (MALAN; LEITNER, 2007).

O sistema de programação Scratch (Figura 1) tem como objetivo ajudar os usuários a criarem intuições sobre programação de computadores (MALONEY et al., 2010). Tanto o ambiente, quanto a linguagem de programação trabalham juntos para criar um sistema rápido de aprendizagem, simples e sem a necessidade de explicação prévia.

Figura 1 – Scratch



Fonte: (MALAN; LEITNER, 2007).

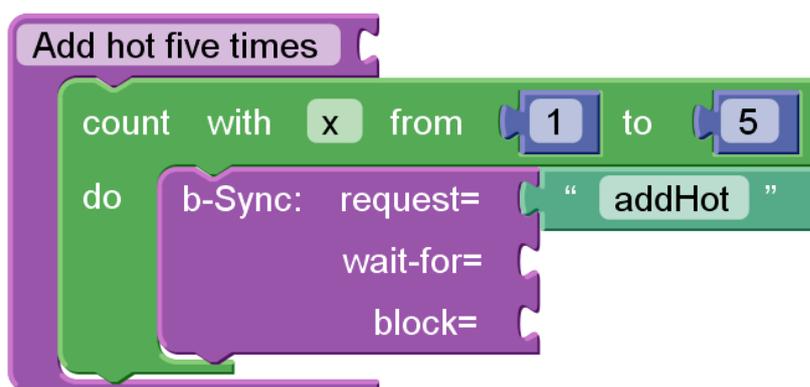
O layout da interface do usuário com sua paleta de comandos colorida, convida os usuários a programar através da linguagem dos blocos, em que é permitido os usuários se concentrarem em problemas ao invés de se esforçarem apenas para obter a compilação do programa (MALONEY et al., 2010).

Embora seu principal objetivo seja o auxílio no desenvolvimento da fluência tecnológica entre jovens, é notória sua presença em um ambiente de programação no ensino superior no processo de aprendizagem do Pensamento Computacional. A ferramenta permite que estudantes programem com um mouse, apresentando construções programáveis como blocos (peças de quebra cabeças) que só funcionam em conjunto se forem sintaticamente apropriadas. Dentre esses blocos existem declarações fundamentais, expressões booleanas, condições, laços e variáveis (MALAN; LEITNER, 2007).

### 2.1.0.2 Blockly

Durante a década passada, houve um grande aumento de linguagens baseadas em blocos usadas para auxiliar a exposição inicial a conceitos de programação para novos alunos, sejam crianças ou adultos (TROWER; GRAY, 2015). Estas linguagens servem como um escudo de frustrações para estudantes que estão aprendendo programação, fazendo com que o sujeito pense somente no conteúdo computacional que está sendo passado, ao invés da sintaxe em si. Ademais, a combinação da visualização e customização são excelentes para demonstrar técnicas que precisariam de uma maior abstração do código e, provavelmente estruturas mais complexas (MARRON; WEISS; WIENER, 2012). Além disso, o ambiente Google Blockly foi construído com princípios semelhantes à linguagem Scratch (Figura 2).

Figura 2 – Blockly



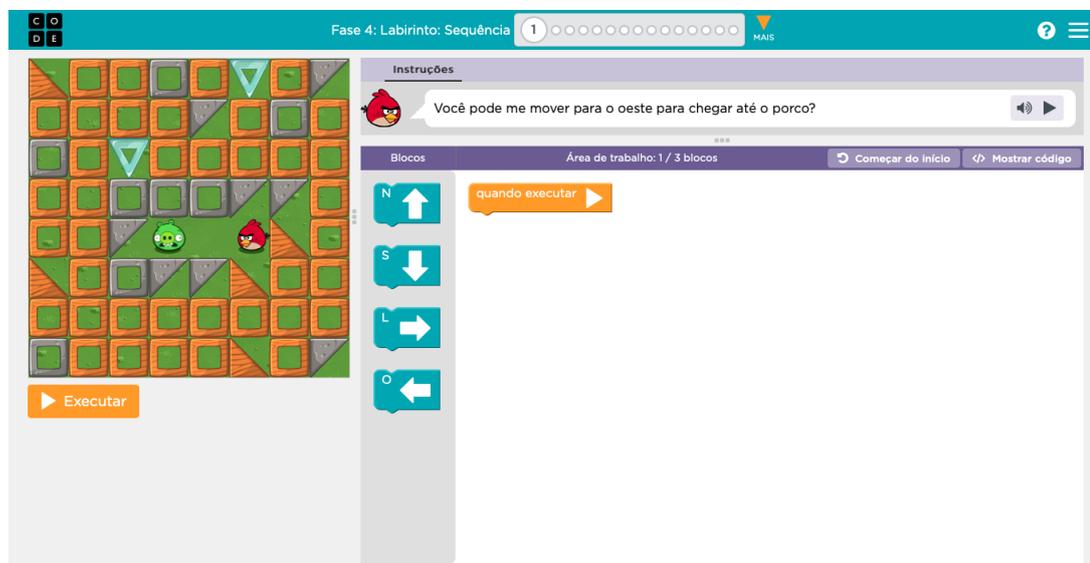
Fonte: (MARRON; WEISS; WIENER, 2012).

### 2.1.0.3 Code.org

O Code.org (Figura 3) é uma plataforma de aprendizagem online (CODE.ORG, 2019), que oferece cursos para alunos de todas as idades. Atualmente, são noventa mil turmas de primeiro e segundo grau, somando assim, três milhões de alunos envolvidos. Nos exercícios propostos são abordados conceitos introdutórios da ciência da computação,

como laços, eventos, condicionais, algoritmos, cidadania digital, entres outros assuntos (WILSON, 2015).

Figura 3 – Code.org



Fonte: (CODE.ORG, 2019).

Segundo Kalelioğlu (2015), o site oferece muitas ferramentas fáceis de usar, tornando-o assim, adequado para estudantes que querem aprender programação. Além disso, existem outras ferramentas como a utilização de personagens populares, gameificação e até mesmo dicas para soluções, que funcionam de forma motivacional, ao mesmo tempo em quem monitoram o progresso do usuário. Deste modo, os conceitos de computação são assimilados com mais facilidade, tornando a experiência agradável.

### 2.1.1 Pensamento Computacional

O Pensamento Computacional remete a uma visão de mundo em termos de uma série de problemas que possuem soluções computacionais. Os cientistas da computação buscam abordar os problemas de um modo particular, buscando soluções algorítmicas, em termos de manipulação de dados e controle de processos (EASTERBROOK, 2014).

Para Wing (2006), o pensamento computacional é a habilidade de aplicar técnicas e conceitos da ciência da computação para resolver problemas comuns como refazer os passos para localizar um objeto perdido, colocar na mochila itens necessários para uma viagem ou decidir entre adquirir um carro ou andar de táxi. Em suma, o pensamento computacional se equipara às habilidades básicas de aritmética, leitura e escrita.

Easterbrook (2014) acrescenta que essas habilidades são acompanhadas por um conjunto de atitudes mais amplas, incluindo a capacidade de lidar com complexidade e problemas em aberto, tolerância à ambiguidade, e capacidade de trabalhar com os outros

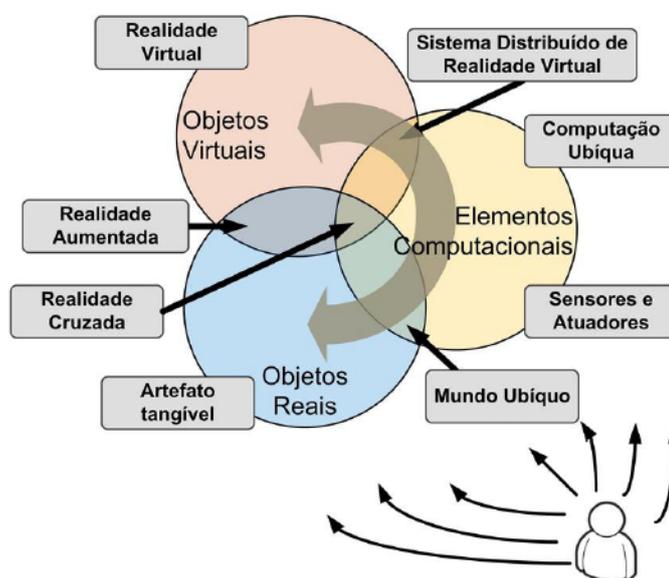
para alcançar um objetivo comum. Hoje o termo foi expandido para incluir o pensamento com muitos níveis de abstrações, o uso de matemática para desenvolver algoritmos, e examinar o quão bem uma solução é dimensionada em diferentes tamanhos de problemas (DENNING, 2009).

Para Ramos et al. (2015), o que parece ser comum às diferentes abordagens relativas ao desenvolvimento do pensamento computacional, é a adoção de uma grande diversidade de suportes lógicos e tecnologias, permitindo a criação e o desenvolvimento de ferramentas auxiliares para sua aprendizagem. Dentre estas ferramentas, se destacam o jogos digitais que, por serem de natureza colaborativa, possibilitam estimular a resolução de conflitos e a tomada de decisão coletiva para atingir um objetivo comum (RODRIGUEZ et al., 2015).

## 2.2 Realidades Mistas

O formato de interação do usuário com a informação também vem sofrendo modificações, permitindo interações de várias maneiras com as aplicações virtuais. Uma tecnologia interativa que é relativamente nova, é o conjunto de realidades chamado de Realidades Mistas, que podem ser observadas na Figura 4.

Figura 4 – Realidades Mistas de Kirner



Fonte: (KIRNER; CERQUEIRA; KIRNER, 2012).

Este conjunto de realidades inclui Realidade Virtual, Realidade Aumentada e Realidade Cruzada, tecnologias vanguardistas que podem fornecer uma experiência divertida e educativa, além de enriquecer a experiência, proporcionando uma interação intuitiva com os artefatos utilizados (KIRNER; CERQUEIRA; KIRNER, 2012).

Uma questão crucial em aplicações de Realidades Virtuais, Aumentadas e Cruzadas, é a exibição das informações a partir do ponto de vista atual do usuário e, mais particularmente, de acordo com a posição dos olhos do usuário. Com o objetivo de atender a este requisito, o *Head-Mounted Display* (Figura 5) é um componente usado na cabeça como um capacete e/ou com fone de ouvido que acaba se tornando de extrema importância para que este tipo de sistema funcione como uma forma de experiência de imersão em outra realidade (ITO; KLINKER, 2014).

Figura 5 – Head-Mounted Display



Fonte: Próprio autor.

### 2.2.1 Realidade Virtual

Realidade Virtual é uma interface de computador avançada que envolve simulação em tempo real e interações através de canais multissensoriais (BURDEA; COIFFET, 2003). Por conseguinte, foi a primeira opção de interface tridimensional a surgir que permite a interação natural usando as mãos com ambientes virtuais renderizados em monitores, projeções ou através de HMD.

Para interagir com os elementos virtuais, é necessário ter dispositivos multimodais, tais como luvas com sensores e recursos de rastreamento, mouse, óculos estereoscópicos, dispositivos com acelerômetros, etc. Em ambientes de RV, o usuário pode ver o mundo

virtual através de uma janela renderizada em telas de monitores ou telas de projeção. Outra opção é a inserção do utilizador no mundo virtual através de HMD ou quartos de projeção, chamados de cavernas (*caves*).

Quando o utilizador está totalmente imerso dentro do mundo virtual, a RV é chamada de imersiva; caso o usuário esteja parcialmente inserido no mundo virtual, através do monitor ou equivalente, a RV é chamada de não imersiva (KIRNER; CERQUEIRA; KIRNER, 2012).

Um exemplo de um projeto de RV é o trabalho “Discovery RV”(DISCOVERY, 2019) concebido para dar acesso interativo e explorar novos lugares, personagens e ideias. Com vídeos de RV, o usuário aprende histórias e imerge em experiências inéditas.

### 2.2.2 Realidade Aumentada

Realidade Aumentada é um conceito que sofreu algumas alterações durante o decorrer do tempo. Segundo Azuma (1997), é primariamente um sistema que permite que o usuário veja o mundo real tal como objetos virtuais sobrepostos ou compostos com a realidade, tendo como base três características: uma mesclagem entre elementos visuais e reais, promoção de uma interação em tempo real e um registro em forma tridimensional.

Este conceito está envolvido com a evolução tecnológica, pois, anteriormente, essas características de Realidade Aumentada eram concebidas somente através de elementos tridimensionais. Porém, com o desenvolvimento das interações visuais, táteis e de áudio, o conceito de RA vem ficando mais amplo, sendo considerada uma interface baseada em informação gerada em computador (imagens estáticas e dinâmicas, sons espaciais e sensações táteis) com o uso do ambiente real, provido por dispositivos tecnológicos e usando interação natural no mundo real, como pode ser observado na Figura 4 (KIRNER; KIRNER, 2011).

Uma maneira de trazer informação virtual para o ambiente físico do usuário é a utilização de câmeras, que têm como finalidade a captura de uma transmissão ao vivo do mundo real, permitindo ao computador adicionar e manipular informações virtuais para o mundo real (KIRNER; CERQUEIRA; KIRNER, 2012).

O Notable Women (RIOS, 2019) é um experimento que utiliza tecnologia de realidade aumentada, que permite a visualização interativa de mulheres americanas históricas, que foram deixadas de fora da moeda dos Estados Unidos da América. O usuário foca uma cédula de dólar, com a câmera do dispositivo móvel, e são projetadas digitalmente na tela representações visuais de bustos de mulheres históricas, sobrepondo os bustos originais. A aplicação dá ao usuário a impressão de que os artefatos observados existem no ambiente real como na Figura 6.

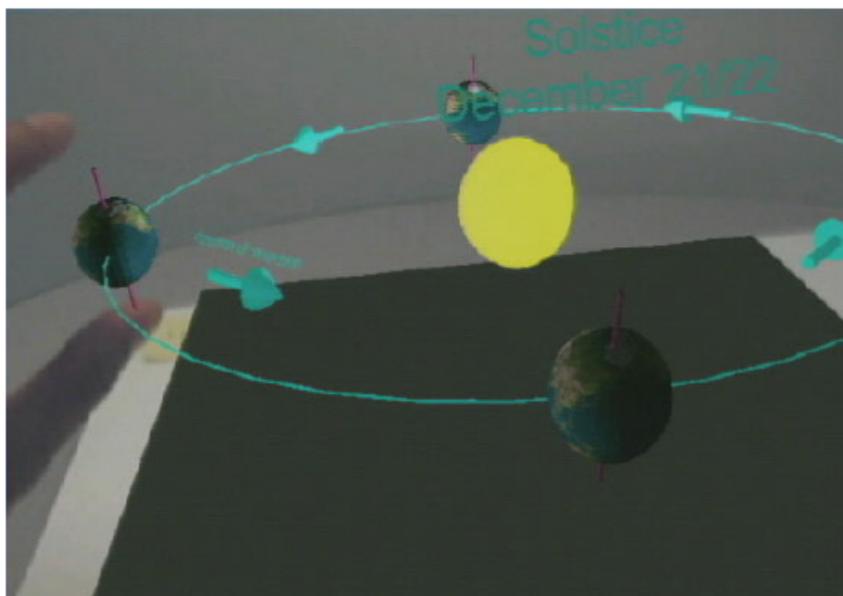
Figura 6 – Notable Women



Fonte: (RIOS, 2019).

Outra aplicação que usa tecnologia de RA para o ensino, é o protótipo de livro aumentado demonstrado por Shelton e Hedley (2002). Neste ambiente, estudantes são apresentados a uma nova maneira de aprender geografia imersivamente, com animações que são plotadas na tela do dispositivo móvel, que tem sua câmera apontada para as páginas físicas do livro (Figura 7).

Figura 7 – Protótipo sobre relação entre terra e sol em RA



Fonte: (SHELTON; HEDLEY, 2002).

Segundo Anami (2013), as vantagens em utilizar RA no aprendizado são:

- Envolve, estimula e motiva os estudantes;

- Viabiliza a execução e visualização de experiências não representáveis no mundo real e conceitos abstratos;
- Melhora a colaboração entre estudantes e educadores;
- Promove a criatividade e imaginação;
- Ajuda na participação do aluno no processo de aprendizagem, permitindo que ele possa adaptá-lo a seu ritmo;
- Cria um ambiente de aprendizado autêntico, adequado a vários estilos de aprendizagem.

### 2.2.3 Realidade Cruzada

Realidade Cruzada envolve um ambiente de realidade mista ubíquo, conforme visto na Figura 4, originado da fusão de uma rede de sensores e atuadores (que recebem e enviam dados relacionados ao mundo real) com mundos virtuais compartilhados, usando uma interface de realidade aumentada, na qual a troca de informações é bidirecional entre o mundo real e o virtual (PARADISO; LANDAY, 2009). A Realidade Cruzada torna possível a interação entre equipamentos remotos, assim como experimentos complexos difíceis de simular (KIRNER; CERQUEIRA; KIRNER, 2012).

Os maiores benefícios das interfaces de Realidades Mistas são as aplicações cuidadosamente desenhadas que podem prover interação intuitiva, sem a necessidade de grandes gastos.

## 2.3 Plataformas e Ferramentas

Para a implementação da aplicação, foram necessárias plataformas sobre as quais a mesma foi executada. Dentre estas plataformas serão apresentados: o sistema operacional Android, o motor de jogos Unity e o kit de desenvolvimento Google ARCore.

### 2.3.1 Android

O Android é um sistema operacional baseado em Linux que foi projetado para dispositivos móveis touchscreen, como smartphones e tablets. O sistema operacional é de código aberto, permitindo que o software seja livremente modificado e distribuído por fabricantes de aparelhos, operadoras de comunicação e desenvolvedores (ANDROID, 2019). Além disso, o Android conta com uma grande comunidade de desenvolvedores que tem como objetivo criar aplicativos que estendem as funcionalidades dos dispositivos, os quais são escritos principalmente em JAVA.

### 2.3.2 Unity e Motores de Jogos

O processo de desenvolvimento de um jogo é significativamente mais rápido quando usando um motor de jogos. Além disto a usabilidade da aplicação se torna superior, especialmente em aspectos de interação com o conteúdo RA e clareza da interface do usuário, possibilitando aos jogadores uma interação mais realista. Ainda, com o auxílio de uma plataforma de criação como o Unity, a expressão gráfica também é aprimorada usando os motores incorporados (KIM et al., 2014).

Unity é o motor de jogos (*game engine*) mais utilizado no mundo, oferecendo aos desenvolvedores de todo o mundo as ferramentas para criar experiências ricas em 2D, 3D, RV e RA. As experiências com o Unity atingem quase 3 bilhões de dispositivos em todo o mundo e foram instaladas 24 bilhões de vezes nos últimos 12 meses (UNITY, 2019b). Uma *game engine* se encarrega de lidar com o hardware gráfico, controla os modelos para serem renderizados, trata das entradas de dados do jogador, trata de todo o processamento de baixo nível e outros itens com os quais o desenvolvedor de jogos normalmente não precisa se preocupar (CLUA; BITTENCOURT, 2005).

### 2.3.3 ARCore

O ARCore é um kit de desenvolvimento criado pela Google que fornece *Software Development Kits* para muitos dos ambientes de desenvolvimento mais populares. Esses SDKs proporcionam *Application Programming Interface* nativas para todos os recursos essenciais de RA, como rastreamento de movimento, compreensão ambiental e estimativa de luz. Com esses recursos, o desenvolvedor pode criar experiências em Realidade Aumentada totalmente novas ou aprimorar aplicativos existentes com recursos RA (ARCORE, 2019).

## 3 Cubol

Nesse capítulo serão abordados os passos para o desenvolvimento do Cubol. Serão relatadas as principais decisões tomadas no decorrer do processo de desenvolvimento do jogo, desde o projeto à sua execução, com suas devidas explicações, além de algumas descrições mais detalhadas a respeito dos elementos essenciais que compõem a sua arquitetura.

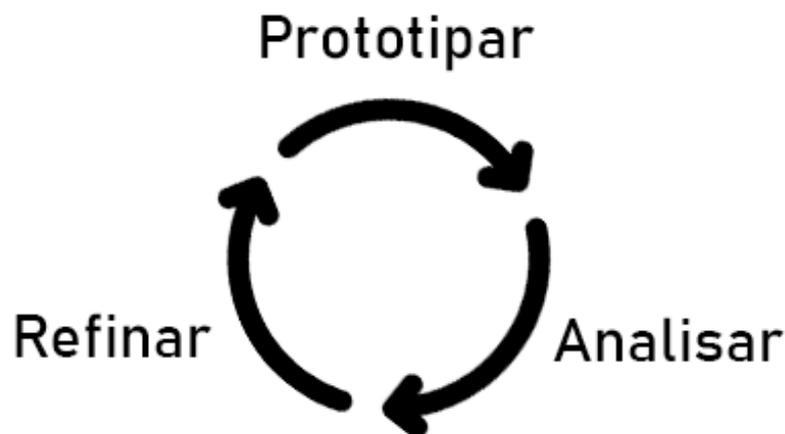
### 3.1 Metodologias

Pelo fato da pouca familiaridade com o assunto, optou-se por uma abordagem exploratória, desenvolvendo assim conhecimento como referência para o projeto.

Define-se pesquisa exploratória, na qualidade de parte integrante da pesquisa principal, como o estudo preliminar realizado com a finalidade de melhor adequar o instrumento de medida à realidade que se pretende conhecer. (QUEIROZ, 1992)

Já metodologia escolhida para ser utilizada na implementação do projeto foi a de Design Interativo de Jogos (Figura 8). O Design interativo é uma metodologia de design de projeto baseada em um processo cíclico de prototipação, análise, e refinamento. Neste tipo de metodologia o jogo é construído e desconstruído diversas vezes no decorrer do processo, até atingir o produto final (ZIMMERMAN, 2003).

Figura 8 – Design Interativo de Zimmerman



Fonte: Próprio autor. Adaptado de (ZIMMERMAN, 2003).

O objetivo é criar o sistema, utilizar, questionar, encontrar falhas e apontar soluções como alternativa. A cada etapa do desenvolvimento, temos uma sucessão de intervenções que são feitas, gerando várias versões do produto, culminando em um melhor acabamento final. Durante o processo são experimentados os protótipos que são gerados.

A aplicação foi projetada para funcionar sobre o sistema operacional Android com o motor de jogos Unity. A implementação, bem como o funcionamento nativo com o sistema operacional, foi auxiliada pelo kit de desenvolvimento Google ARCore.

## 3.2 Propósito do Projeto

O propósito do projeto de desenvolvimento do jogo educativo de realidade aumentada Cubol, é uma prototipação demonstrativa de um aplicativo de aprendizagem para alunos nos primeiros estágios de lógica de programação. Entretanto o artefato poderá ser utilizado por pessoas de todas as faixas etárias. Na escolha das tecnologias foi considerado utilizar hardware e software *open-source* e/ou de baixo custo, ao mesmo tempo em que essas ferramentas sejam condizentes com as necessidades do projeto.

## 3.3 Características Gerais do Jogo

O jogo consiste em um labirinto de blocos (Mapa) no qual o personagem principal - representado pelo cubo azul - deverá caminhar (Figura 9).

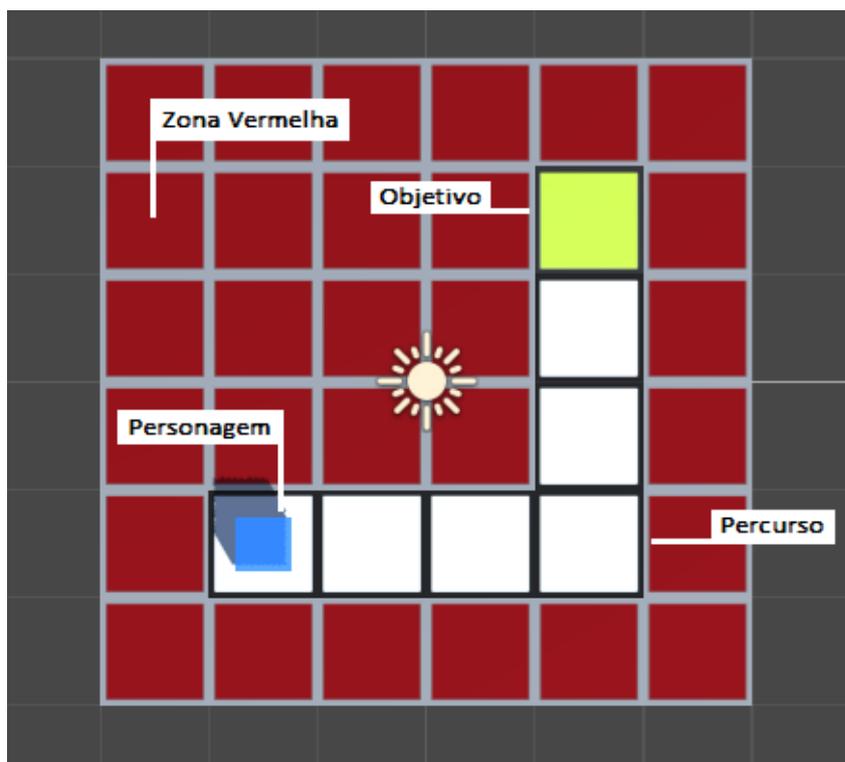
O objetivo do personagem é, partindo do ponto de origem (A), alcançar seu destino (ponto B), representado, no mapa, pelo bloco verde. Para isso, ele deverá analisar o mapa e determinar o percurso que o levará a atingir seu objetivo. A partir daí ele deverá definir um conjunto finito e ordenado de operações, que o levará a percorrer o caminho correto.

### 3.3.1 Mapa

Os mapas presentes no jogo desempenham uma função de extrema importância, sendo uma das principais ferramentas de atração para o jogador, que de forma lúdica, clara e objetiva instigará o usuário a completar a partida. Além disso, através desse instrumento, será possível avaliar a evolução do usuário. Sendo assim, cada mapa será um nível do jogo e cada um deles terá uma dificuldade diferente. Entretanto, em todos os mapas o objetivo será sempre o mesmo: atingir o bloco verde.

Sua caracterização será formada por três diferentes tipos de blocos, sendo eles: os blocos brancos, dos quais poderão ser feitos os trajetos, possibilitando o caminho do personagem; os blocos vermelhos, que representarão as áreas em que o personagem não

Figura 9 – Mapa



Fonte: Próprio autor.

deverá percorrer; e o bloco verde, que será uma peça única, representando o almejado objetivo final.

### 3.3.2 Personagem

O personagem no Cubol, será representado por um simples cubo azul, ligeiramente transparente e de fácil identificação. O fato de se apresentar com aspectos minimalistas é proposital, pois com esse padrão o usuário permanece focado apenas na execução e no objetivo. Todos os comandos escolhidos, serão executados pelo personagem.

### 3.3.3 Percurso

O percurso corresponde à movimentação do personagem na partida, e a tendência é que a dificuldade do trajeto aumente para cada mapa proposto.

O percurso a ser traçado será inteiramente baseado no mapa, que pelas suas características funcionará como um parâmetro para avaliação do usuário. Para isto, o personagem deverá caminhar através do percurso, até chegar ao objetivo. Caso o personagem saia do percurso, tocando na zona vermelha, ou caso os comandos acabem e o mesmo não tenha chegado ao bloco verde, a partida será encerrada sem sucesso.

Para concluir o percurso com sucesso, o usuário deverá apresentar a cadeia correta de comandos que corresponderá ao algoritmo almejado.

### 3.3.4 Objetivo

O objetivo da partida será nitidamente exposto, e visualmente destacado no mapa como um bloco verde. O personagem deverá chegar ao objetivo para concluir o nível.

### 3.3.5 Comandos

Os blocos de comandos funcionarão como peças de quebra cabeças, que só executarão o percurso corretamente se forem sintaticamente apropriadas conjuntamente. Além disso, os blocos são baseados na cores e formatos do Blockly, sendo facilmente memorizados. O usuário deverá empilhar os comandos, dentre eles funções, parâmetros, constantes e laços, na ordem certa a fim de construir o algoritmo corretamente. Quando tiver terminado, o botão de execução deverá ser pressionado, com o intuito de começar a execução do percurso.

## 3.4 Especificação dos Requisitos

### 3.4.1 Requisitos Funcionais

- **RF01** - O protótipo deve permitir a escolha de mapa pelo usuário;
- **RF02** - O protótipo deve permitir a análise do percurso pelo usuário;
- **RF03** - O protótipo deve permitir a escolha dos comandos que serão utilizados;
- **RF04** - O protótipo deve executar o movimento;

### 3.4.2 Requisitos Não Funcionais

- **NF01** - O protótipo deve rodar apenas em dispositivos Android;
- **NF02** - O protótipo deve ser utilizado apenas na orientação vertical;
- **NF03** - O protótipo deve usar a aplicação ARCore como suporte secundário para o seu funcionamento;
- **NF04** - O protótipo deve utilizar a tecnologia de RA, por este motivo o dispositivo deve estar com câmera, acelerômetro e giroscópio funcionando.

### 3.4.3 Atores do Protótipo

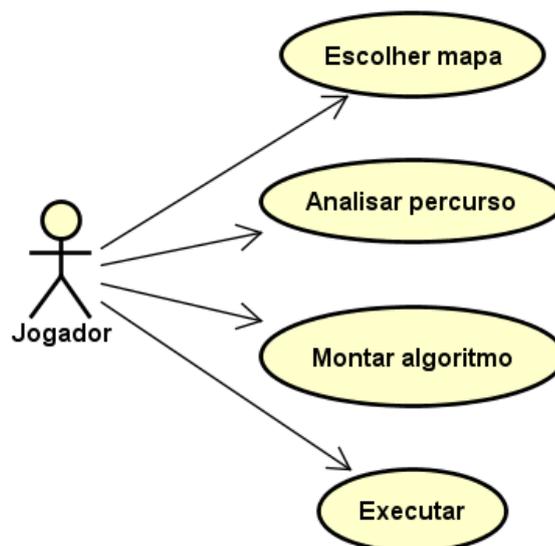
O jogador é o único ator que interage com o protótipo, e é responsável por todas as ações durante o fluxo do processo.

### 3.4.4 Casos de Uso

Como apresentado na Figura 10, o projeto conta com os seguintes casos de uso:

- Escolher o mapa que será utilizado durante a partida;
- Analisar o percurso e algoritmo que será usado;
- Montar o algoritmo com os comandos fornecidos;
- Executar a movimentação.

Figura 10 – Diagrama de Caso de Uso

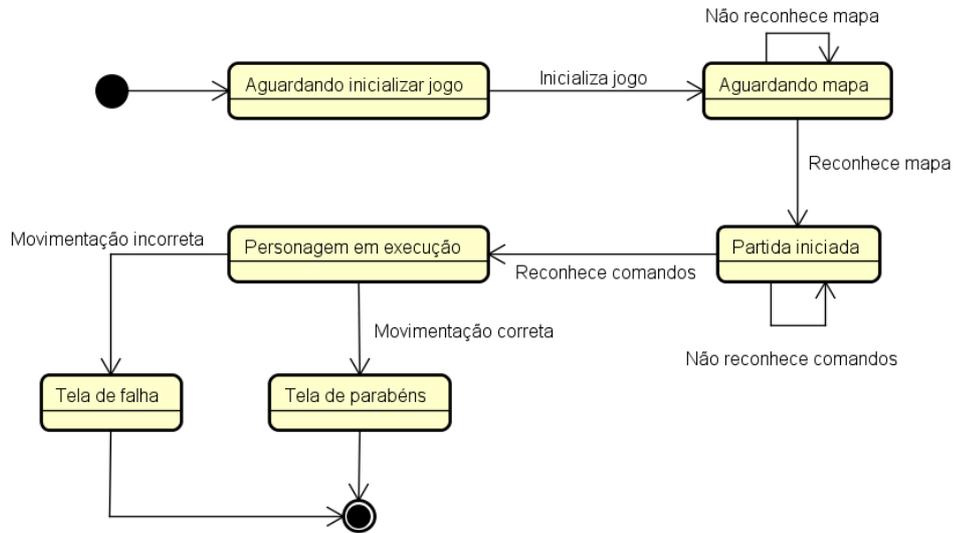


Fonte: Próprio autor.

### 3.4.5 Estados do Protótipo

Serão seis estados no total (Figura 11), além dos estados inicial e final. O usuário inicia a aplicação e entra no estado "Aguardando inicializar o jogo". Ao iniciar jogo, o estado "Aguardando mapa" é ativado. Quando o aluno ativa o mapa que será usado, a partida é iniciada ("Partida iniciada"). A partir daí, será esperado o algoritmo para a execução da movimentação. Caso o personagem chegue ao final do percurso, será ativado o estado de "Tela de parabéns", caso contrário, será ativado o estado de "Tela de falha".

Figura 11 – Diagrama de Estado

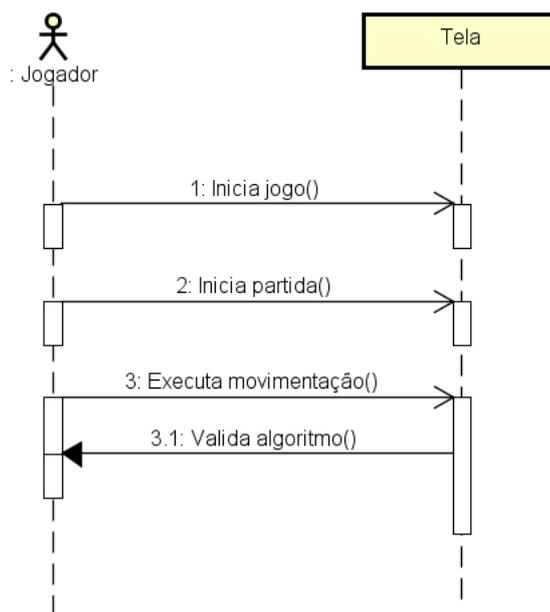


Fonte: Próprio autor.

### 3.4.6 Sequência de Processos

Na Figura 12, é apresentada a sequência global do comportamento do protótipo, ou seja, a representação das mensagens enviadas entre os objetos.

Figura 12 – Diagrama de Sequência

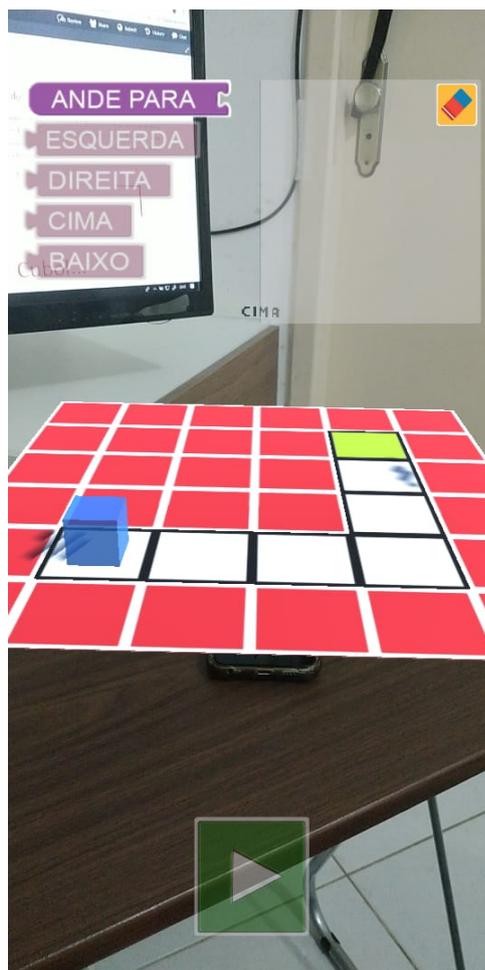


Fonte: Próprio autor.

### 3.5 Características da Aplicação

O aplicativo (Figura 13) pode ser utilizado através de um dispositivo móvel com sistema operacional Android, e que suporte o kit de desenvolvimento Google ARCore. Além destes, foi utilizado o motor de jogos Unity. Para utilizar a aplicação serão necessários os cartões físicos que debloqueiam os mapas, e que serão explicitados posteriormente.

Figura 13 – Cubol



Fonte: Próprio autor.

O jogo foi implementado com base nos quatro tipos de funcionalidades citadas a seguir:

- **Escolha do nível da tarefa** - O usuário poderá escolher dentre vários mapas que serão catalogados por nível de dificuldade;
- **Interação com o mundo real** - Dentro da aplicação, o usuário estará sempre interagindo com o mundo real, trazendo uma imersão para as atividades;
- **Escolha dos blocos de comandos** - Durante a partida, o usuário deverá escolher quais comandos e em qual ordem deverão ser executados;

## 3.6 Realidade Aumenta Aplicada ao Jogo

Esta seção descreve como a Realidade Aumentada funcionará no jogo Cubol.

### 3.6.1 Interação

A interação em realidade aumentada proposta pelo jogo ocorrerá de forma intuitiva. O usuário deverá apontar a câmera do dispositivo para o mapa previamente escolhido, e quando o mapa for carregado digitalmente na tela do aparelho, deverá analisar o percurso a ser seguido. Quando assim o fizer, e por conseguinte escolher a cadeia de comandos necessária para resolver o enigma, o usuário deve executar os comandos para verificar se a movimentação do cubo azul será feita corretamente.

### 3.6.2 Animação

A animação acontece assim que a cadeia for executada pelo usuário. Durante o processo, o personagem percorrerá o caminho proposto pelo algoritmo criado. Quando a animação terminar, o resultado de sucesso ou falha surgirá na tela.

### 3.6.3 Poder Imersivo

Um detalhe importante durante todo o processo da partida, é a possibilidade de análise que o usuário terá sobre a cena. Navegando fisicamente com o celular em mãos, o jogador poderá chegar mais perto, contar e dar voltas no mapa, a fim de chegar a uma conclusão favorável.

## 3.7 Desenvolvimento

### 3.7.1 *Game Design Document*

O *game design* é um documento que funciona como estrutura do jogo e deve conter todas as informações de funcionamento de todos os itens e funcionalidades presentes. O responsável pelo *game design* tem como objetivo apresentar uma descrição abrangente de todos os aspectos, para que assim, os respectivos responsáveis destes aspectos possam implementá-los (SCHUYTEMA, 2007).

Esta seção de Desenvolvimento, representa um breve resumo do *Game Design Document*, que descreve os detalhes do jogo Cubol.

### 3.7.2 História

O jogo se passa dentro dos circuitos eletrônicos de um computador, que tem a necessidade de transporte de bits nestes circuitos para o seu funcionamento. Um desses bits é chamado de Cubit, que deve chegar até o seu destino para o funcionamento correto da máquina. O problema é que o cubo não pode tocar nas áreas vermelhas, que não fazem parte da via, representada pelo percurso correto da arquitetura do sistema. Além disso, se Cubit parar pelo caminho, também vai acabar obstruindo a passagem de outros bits, causando assim a falha geral.

### 3.7.3 Modelagem

A modelagem levou em conta aspectos da arquitetura de um hardware computacional, e além disso, as estruturas poligonais e retas são predominantes para a melhor visualização quando a estrutura do mapa está em RA.

O aspecto minimalista foi proposital, fazendo com que o jogador não desvie sua atenção do seu objetivo.

### 3.7.4 Textura

Pelo fato da arte não ser o foco do protótipo, foram criadas texturas simples, porém de cores fortes, para facilitar o entendimento das peças que compõem a estrutura da partida.

### 3.7.5 Colisões

O bloco do jogador tem peso, gravidade e uma leve falta de atrito com o piso, deste modo pode deslizar livremente pelo mapa. Existem dois gatilhos de colisões: um na Zona Vermelha, representada pelo quadrado vermelho, que causa o fim do jogo com falha, e outro no Fim, representado pelo quadrado verde, que acarreta no sucesso da partida.

### 3.7.6 Animações

Pelo fato do kit de desenvolvimento ter execução de instância muito custosa para o dispositivo, foram usadas animações simples para movimentação do cubo, evitando assim travamento durante a execução. Além disso, esta simplicidade também representa bem a abstração de deslocamento de pulsos em um circuito eletrônico.

### 3.7.7 Inserção na Realidade Aumentada

Os elementos digitais serão inseridos na tela através de imagens aumentadas. Os mapas são cartões com *QR Code*, no espaço físico real, e ao serem capturados pela câmera

do dispositivo são transpostos para suas representações digitais. Durante uma partida, só será possível ser ativado um mapa por vez.

### 3.7.8 Compilação e *Building*

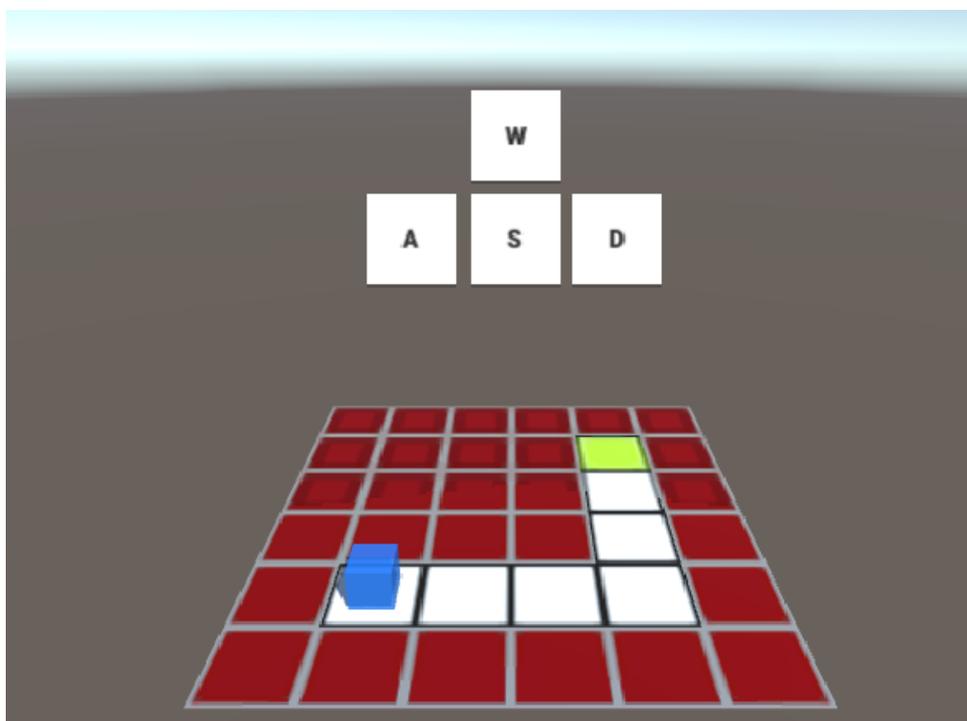
No processo de compilação são usados alguns *scripts*: para a construção e execução dos menus, para gerenciamento dos estados do jogo e para a colisão.

### 3.7.9 Protótipos

#### 3.7.9.1 Primeiro Protótipo

O primeiro protótipo (Figura 14) foi construído, ainda sem a RA, levando em conta a abstração do mapa bem como a movimentação livre do jogador. A movimentação do personagem é acionada com comandos simples: para baixo, cima, direita e esquerda. A partir destes comandos, o bloco azul avança apenas um quadrado do piso no sentido correspondente. Usando colisões de objeto, foram adicionados estados finais do jogo. Quando o personagem entra em contato com o quadrado vermelho (Zona Vermelha) a aplicação acusa fim do jogo. Já quando o personagem por fim chega ao quadrado verde (Fim) é finalizada a partida com sucesso.

Figura 14 – Primeiro Protótipo



Fonte: Próprio autor.

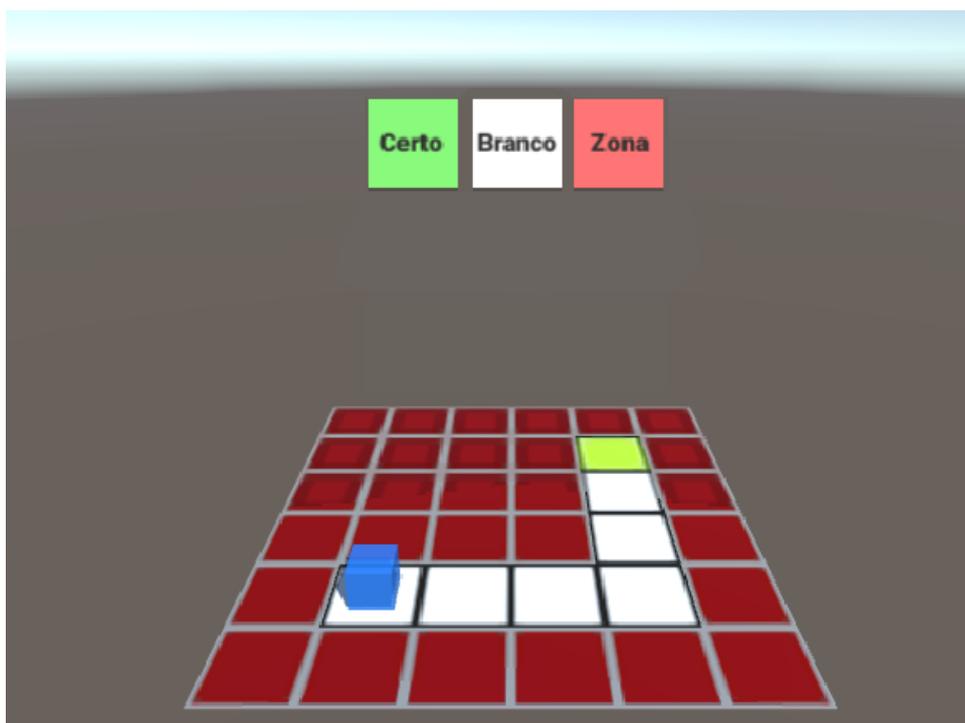
Aqui, foram usados direção e sentido fixo para onde o objeto deve ser movido (ponto futuro), representados pelos eixos X e Z do Unity. O ponto futuro é calculado com

base na distância a ser percorrida, e fica por conta do tamanho dos blocos utilizados para a formação do mapa. O mapa é formado por trinta e seis blocos (6x6), incluindo blocos de Percurso, blocos de Zona Vermelha e um bloco Fim. Com as ações citadas, foi possível testar toda a movimentação necessária para a implementação do protótipo final.

### 3.7.9.2 Segundo Protótipo

O segundo protótipo (Figura 15) teve botões representando tipos de percursos diferentes, adicionados ao seu menu de movimentos. O primeiro botão, de cor verde, faz o percurso corretamente até chegar ao Fim, concluindo com sucesso a partida. O segundo botão (branco), executa movimentos sobre o percurso correto, porém sem chegar ao quadrado verde, acarretando deste modo na falha da partida. O terceiro e último botão, vermelho, também termina a partida com falha, porque através dos movimentos leva o cubo azul até a Zona Vermelha. Neste protótipo foram testados o estados do jogo (Figura 11).

Figura 15 – Segundo Protótipo



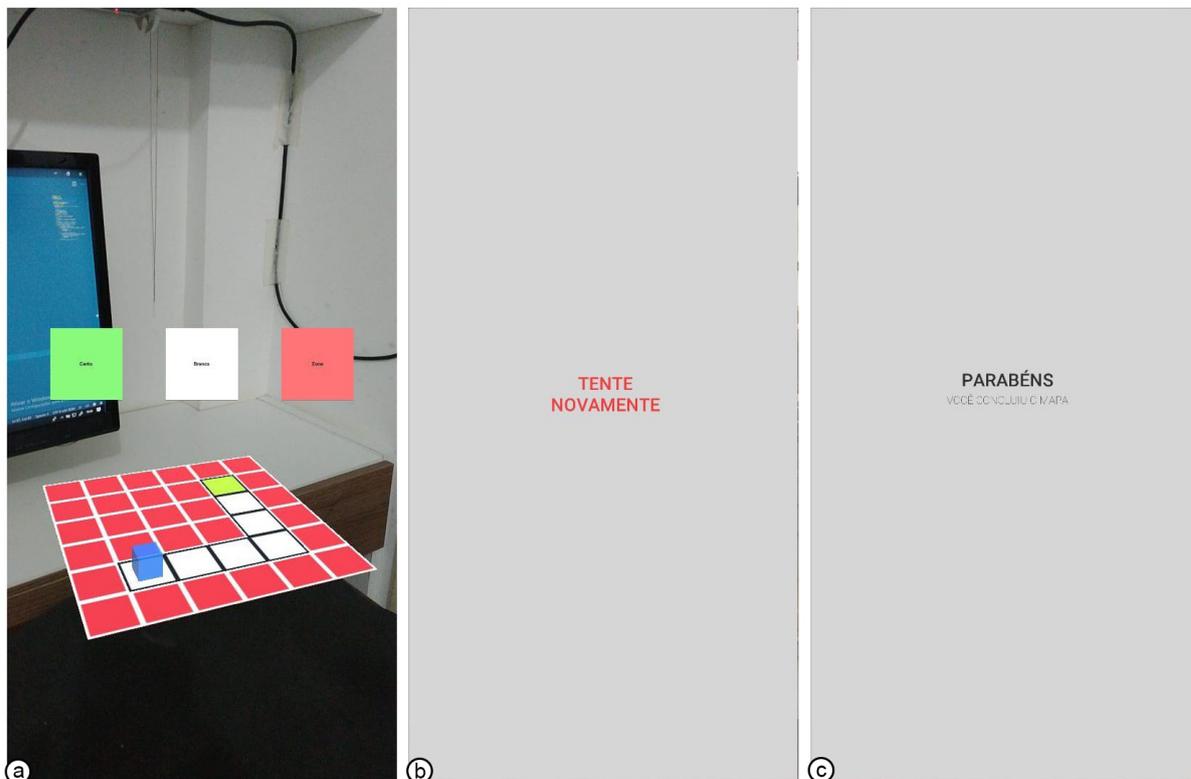
Fonte: Próprio autor.

### 3.7.9.3 Terceiro Protótipo

No terceiro protótipo (Figura 16), foi implementada a primeira versão de RA, com o mapa sendo plotado através de um toque na superfície desejada. Todos os botões do protótipo anterior foram adicionados a este protótipo, para os testes de movimentação.

Também foram adicionadas as telas de falha e de parabéns, para a ativação dos estados finais do jogo.

Figura 16 – Terceiro Protótipo



(a) RA (b) Tela de Falha (c) Tela de Parabéns

Fonte: Próprio autor.

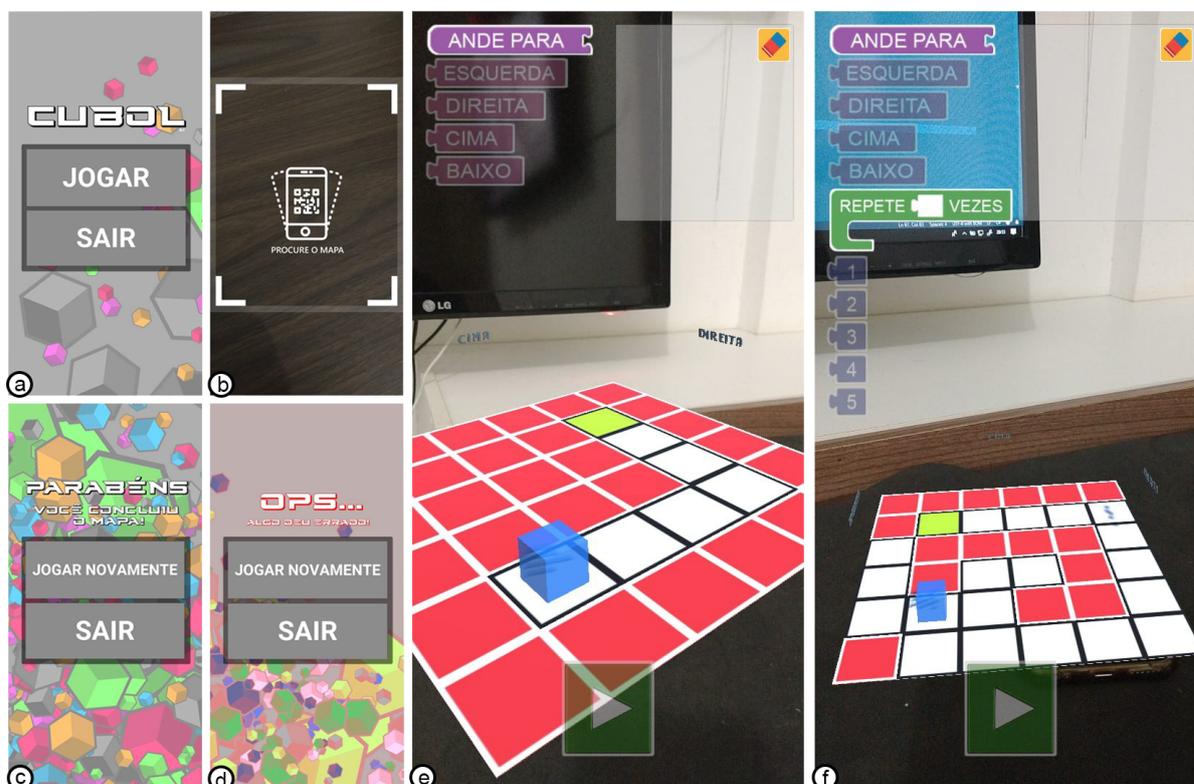
Nesta fase, houve dificuldade com a movimentação do personagem, porque antes o objeto se movimentava entre os eixos X e Z. Posteriormente, com o deslocamento da câmera, e por conseguinte do mapa, em terceira dimensão, o vetor de direção acabava se perdendo. A solução foi, em vez do ponto futuro ser a direção escolhida, passar a ser a mesma posição do bloco de destino para aquele movimento. Quanto à execução do jogo, alguns comportamentos indesejados foram observados durante a prototipação. No primeiro protótipo com RA, quando o objeto se movimentava pelo mapa acabava perdendo o senso de direção e não se movia para o local desejado. A solução foi provida com a inclusão do cálculo para antever o ponto futuro através de direções fixas do mapa (esquerda, direita, cima e baixo), que são especificadas visualmente para o usuário.

#### 3.7.9.4 Protótipo Final

O protótipo final (Figura 17), e versão 0.1 da aplicação, é o escopo desejado para o projeto. Nesta fase, a aplicação ganhou telas iniciais, além da função de reinicialização da

partida. Aqui foi implementada a RA com imagem, ou seja, o artefato é plotado na tela do dispositivo móvel quando a imagem correta referente a ele for focada.

Figura 17 – Protótipo Final



(a) Início (b) Procurando mapa (c) Parabéns (d) Falha (e) Mapa um (f) Mapa dois

Fonte: Próprio autor.

Foi desenhado um novo mapa, com um percurso mais longo, e um novo tipo de bloco de comando, o laço. Sendo assim, possuindo um nível de dificuldade maior. Também foram adicionados: pontos de direção e sentido para a localização do usuário, o painel de console para a visualização dos comandos escolhidos, o botão de limpar os comandos, o botão de executar e os blocos de comando.

O comando *ANDE PARA* é a abstração de um método, que recebe como parâmetro uma das constantes: *ESQUERDA*, *DIREITA*, *CIMA* e *BAIXO*. O laço de repetição, representado pelo bloco verde, também deve receber a quantidade de repetições que devem acontecer, através de blocos numerados de um a cinco.

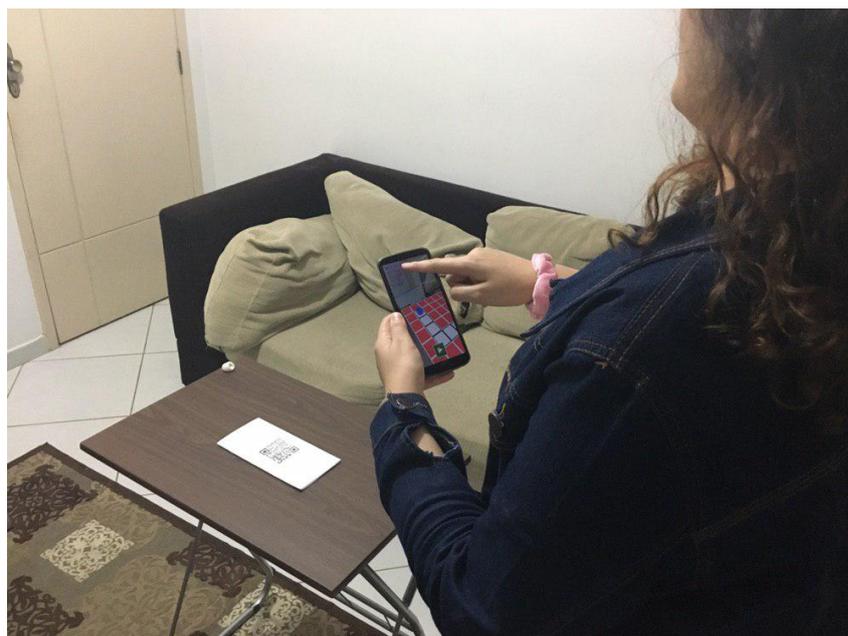
Uma dificuldade encontrada nesta etapa, era que o mapa acabava inclinando e o bloco azul caía da plataforma. Para corrigir, foi agregada uma propriedade de *kinematic*<sup>1</sup> ao objeto, para a sua fixação no ponto.

<sup>1</sup> Se a propriedade Kinematic estiver ativada, forças, colisões ou juntas não afetarão mais o corpo rígido do objeto (UNITY, 2019a).

### 3.7.10 Testes

Foram feitos testes com dois usuários (Figura 18), que não possuíam experiência com a aprendizagem do PC anteriormente. Foi constatado que mesmo sem a explicação prévia da ferramenta, os sujeitos conseguiram finalizar o percurso do primeiro mapa com sucesso, com no máximo duas tentativas.

Figura 18 – Testes



Fonte: Próprio autor.

Já o segundo mapa, que teve adicionado o bloco de repetição, acabou por necessitar de uma breve explicação de seu funcionamento.

Os usuários apresentaram sensação de satisfação ao resolverem os problemas propostos, tendo uma experiência imersiva com o protótipo. Outro fato que pareceu contar a favor do artefato, foi a linguagem colorida tanto do mapa, quanto dos blocos de comando, aumentando a curiosidade dos sujeitos para com o experimento.

## 4 Considerações Finais

A aplicação da Realidade Aumentada no auxílio ao desenvolvimento do Pensamento Computacional é importante porque tem o poder de criar um ambiente de aprendizado autêntico, adequado a diferentes estilos de aprendizagem. Esta abordagem ajuda na participação do aluno no processo de aprendizagem, promovendo a criatividade e imaginação entre os estudantes. Estes fatos ocorrem através de experiências abstratas que não podem ser concebidas no mundo real. Dada a importância do tema, torna-se necessário o desenvolvimento de alternativas ao processo ensino-aprendizagem de programação, e deste modo fazer com que o aluno consiga absorver mais facilmente a disciplina. Portanto é imprescindível desenvolver o PC de forma menos traumática e mais intuitiva.

Partindo destes princípios, foi proposto o Cubol, um jogo educativo que auxilia no desenvolvimento do Pensamento Computacional, utilizando técnicas de RA. A ferramenta permite a criação de algoritmos com o simples movimento da câmera do dispositivo móvel, apresentando construções que só funcionam em conjunto. Dentre os comandos existem métodos, parâmetros e laços.

Dentro do escopo deste trabalho foi desenvolvido um protótipo que possibilitou uma análise de como uma ferramenta lúdica com o propósito educativo pode auxiliar no desenvolvimento do Pensamento Computacional. Além disso, também permitiu a pesquisa e definição do estado da arte do conceito de Realidade Aumentada aplicado a modelos educacionais contemporâneos. Foram documentadas soluções utilizando hardware e software de baixo custo, sem perder a capacidade computacional do produto final. Todas as ferramentas utilizadas no protótipo visam o modelo de desenvolvimento de prototipação, facilitando o acesso e possível continuidade, modificação e expansão. O Software não teve custos para obtenção de licenças de uso, alcançando uma redução ainda mais significativa quanto ao custo final da solução. Dessa forma, as contribuições desse trabalho possibilitaram demonstrar a viabilidade de soluções de baixo custo.

Ao realizar os testes da ferramenta com pessoas que nunca tiveram contato com a aprendizagem do Pensamento Computacional, pelo fato dos usuários terem resolvido os problemas propostos, foi possível constatar o rápido crescimento na curva de aprendizado, apesar do contato com abstrações de linguagem de programação nunca antes experimentadas pelos mesmos.

Nesse sentido, constatou-se que a utilização de recursos lúdicos permitem aos estudantes aprenderem de forma imersiva e não traumática. Além disso, o ambiente é propício a atividades em pares ou em grupo, e auxilia o desenvolvimento do Pensamento Computacional, tornando o primeiro contato com a disciplina de programação em cursos

de Ciência da Computação prazerosa.

## 4.1 Dificuldades do Projeto

Devido ao fato de ser uma tecnologia relativamente nova, a maioria dos dispositivos móveis não dispõe de uma série de requisitos para a utilização da RA, como por exemplo giroscópio embutido no hardware. Aliado a isto, como a implementação foi feita com auxílio do SDK Google ARCore, só dispositivos mais novos e, portanto, mais custosos, suportam a aplicação, limitando o uso do aplicativo a tais aparelhos.

Outra dificuldade encontrada foi a falta de documentação sobre os artefatos utilizados. Isso se dá pelo fato de ainda ser uma tecnologia vanguardista, com pouco uso até então.

Devido às limitações de tempo, não foi possível realizar testes para aferir a influência do uso do Cubol no desenvolvimento do pensamento computacional, entre outras funcionalidades. Para dar continuidade ao trabalho, será necessário elaborar um plano de testes que permita validar devidamente aplicativo.

## 4.2 Trabalhos Futuros

Para a progressão do trabalho, existem algumas sugestões para o aperfeiçoamento do protótipo proposto:

- **Aplicação de questionários** - Realização de testes, com aplicação de questionário para a constatação da eficácia da ferramenta;
- **Criação de novos mapas** - Criação de novos mapas, adicionando conseqüentemente mais dificuldade;
- **Adição de novos comandos** - Novos blocos de comandos podem ser adicionados, como expressões booleanas, estruturas condicionais e variáveis;
- **Rank** - Com o intuito de incentivar a competição entre os alunos, um *rank* classificatório pode ser implementado, com medição de tempo e quantidade de comandos usados para estímulo entre grupos;
- **Deploy** - *Deploy* de um aplicativo completo na loja de aplicativos do Android, para que o usuário possa baixar e utilizar.

## Referências

- ALLAN, V.; KOLESAR, M. Teaching computer science: A problem solving approach that works. ERIC, 1996. Citado na página 13.
- ANAMI, B. M. Boas práticas de realidade aumentada aplicada à educação. *Trabalho de Conclusão de Curso (Graduação em Ciência da Computação)*, Universidade Estadual de Londrina, v. 49, 2013. Citado na página 23.
- ANDROID. *Página home do site Android*. 2019. Último acesso em 28 de março de 2019. Disponível em: <<https://www.android.com/>>. Citado na página 24.
- ARCORE. *Página develop do site Google ARCore*. 2019. Último acesso em 27 de março de 2019. Disponível em: <<https://developers.google.com/ar/develop/>>. Citado na página 25.
- AZUMA, R. Tracking requirements for augmented reality. *Communications of the ACM*, Association for Computing Machinery, Inc., v. 36, n. 7, p. 50–52, 1993. Citado na página 13.
- AZUMA, R. T. A survey of augmented reality. *Presence: Teleoperators & Virtual Environments*, MIT Press, v. 6, n. 4, p. 355–385, 1997. Citado na página 22.
- BECKER, F. O que é construtivismo. *Revista de educação AEC, Brasília*, v. 21, n. 83, p. 7–15, 1992. Citado 2 vezes nas páginas 12 e 15.
- BURDEA, G. C.; COIFFET, P. *Virtual reality technology*. [S.l.]: John Wiley & Sons, 2003. Citado na página 21.
- CLUA, E. W. G.; BITTENCOURT, J. R. Desenvolvimento de jogos 3d: concepção, design e programação. In: *XXIV Jornadas de Atualização em Informática (JAI) Part of XXIV Congresso da Sociedade Brasileira de Computação*. [S.l.: s.n.], 2005. p. 22–29. Citado na página 25.
- CODE.ORG. *Página home do site Code.org*. 2019. Último acesso em 28 de março de 2019. Disponível em: <<https://code.org>>. Citado 2 vezes nas páginas 18 e 19.
- DENNING, P. J. Beyond computational thinking. *Communications of the ACM*, v. 52, n. 6, p. 28–30, 2009. Citado na página 20.
- DISCOVERY. *Página Home do site Discovery VR*. 2019. Último acesso em 4 de julho de 2019. Disponível em: <<https://www.discoveryvr.com/>>. Citado na página 22.
- EASTERBROOK, S. From computational thinking to systems thinking: A conceptual toolkit for sustainability computing. In: ATLANTIS PRESS. *ICT for Sustainability 2014 (ICT4S-14)*. [S.l.], 2014. Citado 2 vezes nas páginas 13 e 19.
- FORTUNA, T. R. *A formação lúdica docente e a universidade: contribuições da ludobiografia e da hermenêutica filosófica*. Tese (Doutorado) — Universidade Federal do Rio Grande do Sul, 2011. Citado na página 16.

- GOMES, D. V. Ludicidade na universidade-essa rima combina? uma experiência de formação lúdico-transdisciplinar na formação inicial de professores. *Ser professor na contemporaneidade: desafios, ludicidade e protagonismo*, v. 2, 2008. Citado na página 16.
- HERNANDEZ, C. C. et al. A disciplina de resolução de problemas na matriz curricular de ciência da computação. In: *XVI Workshop sobre Educação em Computação (WEI)*. In *Proceeding of the XXX Congresso da SBC, Belo Horizonte, MG*. [S.l.: s.n.], 2010. Citado na página 13.
- ITOH, Y.; KLINKER, G. Interaction-free calibration for optical see-through head-mounted displays based on 3d eye localization. In: *3DUI*. [S.l.: s.n.], 2014. p. 75–82. Citado na página 21.
- KALELIOĞLU, F. A new way of teaching programming skills to k-12 students: Code. org. *Computers in Human Behavior*, Elsevier, v. 52, p. 200–210, 2015. Citado na página 19.
- KIM, S. L. et al. Using unity 3d to facilitate mobile augmented reality game development. In: IEEE. *2014 IEEE World Forum on Internet of Things (WF-IoT)*. [S.l.], 2014. p. 21–26. Citado na página 25.
- KINDERGARTEN, L. *Página overview do site Lifelong Kindergarten*. 2019. Último acesso em 28 de março de 2019. Disponível em: <<https://www.media.mit.edu/groups/lifelong-kindergarten/overview/>>. Citado na página 15.
- KIRNER, C.; CERQUEIRA, C. S.; KIRNER, T. G. Using augmented reality cognitive artifacts in education and virtual rehabilitation. In: *Virtual Reality in Psychological, Medical and Pedagogical Applications*. [S.l.]: InTech, 2012. Citado 5 vezes nas páginas 12, 14, 20, 22 e 24.
- KIRNER, C.; KIRNER, T. G. Evolução e tendências da realidade virtual e da realidade aumentada. *Livro do XIII Pré-Simpósio de Realidade Virtual e Aumentada, Uberlândia*, p. 10–25, 2011. Citado na página 22.
- KIRNER, C.; ZORZAL, E. R. Aplicações educacionais em ambientes colaborativos com realidade aumentada. In: *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*. [S.l.: s.n.], 2005. v. 1, n. 1, p. 114–124. Citado na página 12.
- LUCKESI, C. C. Ludopedagogia-ensaios 1: educação e ludicidade. *Salvador: UFBA/FACED/Programa de Pós-Graduação em Educação*, v. 1, 2000. Citado 2 vezes nas páginas 12 e 16.
- LUCKESI, C. C. Ludicidade e atividades lúdicas: uma abordagem a partir da experiência interna. *Ludicidade: o que é mesmo isso*, p. 22–60, 2005. Citado na página 16.
- MALAN, D. J.; LEITNER, H. H. Scratch for budding computer scientists. In: ACM. *ACM Sigcse Bulletin*. [S.l.], 2007. v. 39, n. 1, p. 223–227. Citado 2 vezes nas páginas 17 e 18.
- MALONEY, J. et al. The scratch programming language and environment. *ACM Transactions on Computing Education (TOCE)*, ACM, v. 10, n. 4, p. 16, 2010. Citado na página 17.

- MARRON, A.; WEISS, G.; WIENER, G. A decentralized approach for programming interactive applications with javascript and blockly. In: ACM. *Proceedings of the 2nd edition on Programming systems, languages and applications based on actors, agents, and decentralized control abstractions*. [S.l.], 2012. p. 59–70. Citado na página 18.
- MILGRAM, P.; KISHINO, F. A taxonomy of mixed reality visual displays. *IEICE TRANSACTIONS on Information and Systems*, The Institute of Electronics, Information and Communication Engineers, v. 77, n. 12, p. 1321–1329, 1994. Citado na página 13.
- MUSSER, G. L.; SHAUGHNESSY, J. M. Estratégias de resolução de problemas na matemática escolar. *A resolução de problemas na matemática escolar*. São Paulo: Atual, p. 188–201, 1997. Citado na página 13.
- PANTELIDIS, V. Reasons to use virtual reality in education, vr in the schools 1 (1), 1995. URL: <http://www.soe.ecu.edu/vr/reas.html> (Revised 2000), 1995. Citado na página 12.
- PAPERT, S. *Constructionism: A new opportunity for elementary science education*. [S.l.]: Massachusetts Institute of Technology, Media Laboratory, Epistemology and . . . , 1986. Citado 2 vezes nas páginas 12 e 15.
- PAPERT, S.; HAREL, I. Situating constructionism. *Constructionism*, v. 36, n. 2, p. 1–11, 1991. Citado 2 vezes nas páginas 12 e 16.
- PARADISO, J. A.; LANDAY, J. A. Guest editors' introduction: Cross-reality environments. *IEEE Pervasive Computing*, IEEE, v. 8, n. 3, 2009. Citado na página 24.
- PATTO, M. H. S. A produção do fracasso escolar: histórias de submissão e rebeldia. In: *A produção do fracasso escolar: histórias de submissão e rebeldia*. [S.l.: s.n.], 1993. Citado na página 15.
- PINHO, M. S. Realidade virtual como ferramenta de informática na educação. *Anais do SBIE (Simpósio Brasileiro de Informática na Educação)*, Belo Horizonte, Minas Gerais, 1996. Citado na página 12.
- QUEIROZ, M. I. P. d. O pesquisador, o problema da pesquisa, a escolha de técnicas: algumas reflexões. *LANG, ABSG, org. Reflexões sobre a pesquisa sociológica*. São Paulo, Centro de Estudos Rurais e Urbanos, p. 13–29, 1992. Citado na página 26.
- RAMOS, E. et al. Pensamento computacional na escola e práticas de avaliação das aprendizagens. uma revisão sistemática da literatura. Universidade do Minho, 2015. Citado na página 20.
- RESNICK, M. Give p's a chance: Projects, peers, passion, play. In: *Constructionism and creativity: Proceedings of the Third International Constructionism Conference*. Austrian Computer Society, Vienna. [S.l.: s.n.], 2014. p. 13–20. Citado na página 15.
- RIOS, R. *Página Notable Women do site Experiments With Google*. 2019. Último acesso em 4 de julho de 2019. Disponível em: <<https://experiments.withgoogle.com/notablewomen>>. Citado 2 vezes nas páginas 22 e 23.
- RODRIGUEZ, C. et al. Pensamento computacional: transformando ideias em jogos digitais usando o scratch. In: *Anais do Workshop de Informática na Escola*. [S.l.: s.n.], 2015. v. 21, n. 1, p. 62. Citado na página 20.

- SANTOS, S. M. P. d. A ludicidade como ciência. *Petrópolis, RJ: Vozes*, v. 227, 2001. Citado na página 16.
- SAVIANI, D. *Escola e democracia*. [S.l.]: Autores associados, 2018. Citado na página 15.
- SCHANK, R. *Virtual Learning. A Revolutionary Approach to Building a Highly Skilled Workforce*. [S.l.]: ERIC, 1997. Citado na página 16.
- SCHUYTEMA, P. *Game design: A practical approach*. [S.l.]: Charles River Media, 2007. Citado na página 33.
- SHELTON, B. E.; HEDLEY, N. R. Using augmented reality for teaching earth-sun relationships to undergraduate geography students. In: IEEE. *The First IEEE International Workshop Augmented Reality Toolkit*. [S.l.], 2002. p. 8–pp. Citado 2 vezes nas páginas 14 e 23.
- SPONHOLZ, S. O professor mediador. *Revistas de Ciências Jurídicas e Sociais da UNIPAR*, v. 6, n. 2, 2003. Citado na página 15.
- TROWER, J.; GRAY, J. Blockly language creation and applications: Visual programming for media computation and bluetooth robotics control. In: ACM. *Proceedings of the 46th ACM Technical Symposium on Computer Science Education*. [S.l.], 2015. p. 5–5. Citado na página 18.
- UNITY. *Página docs do site Unity*. 2019. Último acesso em 3 de outubro de 2019. Disponível em: <<https://docs.unity3d.com/ScriptReference/Rigidbody-isKinematic.html>>. Citado na página 38.
- UNITY. *Página public relations do site Unity 3D*. 2019. Último acesso em 27 de março de 2019. Disponível em: <<https://unity3d.com/public-relations>>. Citado na página 25.
- VERISSIMO, A. C. B.; SANTOS, A. M. dos. Por que pensar o lúdico na universidade? Citado na página 16.
- WILSON, C. Hour of code—a record year for computer science. *ACM Inroads*, ACM, v. 6, n. 1, p. 22–22, 2015. Citado na página 19.
- WING, J. M. Computational thinking. *Communications of the ACM*, v. 49, n. 3, p. 33–35, 2006. Citado na página 19.
- ZIMMERMAN, E. *Design Research: methods and perspectives*, MIT Press, 2003. Citado na página 26.
- ZORZAL, E. R. et al. Realidade aumentada aplicada em jogos educacionais. In: *V Workshop de Educação em Computação e Informática do Estado de Minas Gerais-WEIMIG*. [S.l.: s.n.], 2006. Citado na página 13.