

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ**  
**DEPARTAMENTO ACADÊMICO DE INFORMÁTICA**  
**CURSO DE BACHARELADO EM SISTEMAS DE INFORMAÇÃO**

LUCAS CYULIK

**COMO APRESENTAR PROGRAMAÇÃO PARA CRIANÇAS? ENVOLVER PARA  
DESMISTIFICAR, VALORIZAR PARA IMPLEMENTAR**

TRABALHO DE CONCLUSÃO DE CURSO

**CURITIBA**

**2016**

LUCAS CYULIK

**COMO APRESENTAR PROGRAMAÇÃO PARA CRIANÇAS? ENVOLVER PARA  
DESMISTIFICAR, VALORIZAR PARA IMPLEMENTAR**

Monografia apresentada à disciplina de Trabalho de Conclusão de Curso de Bacharelado em Sistemas de Informação, da Universidade Tecnológica Federal do Paraná, como requisito parcial para obtenção do título de bacharel em Sistemas de Informação.

**Orientadora:** Prof. Dra. Marília Abrahão Amaral

**CURITIBA**

**2016**



## TERMO DE APROVAÇÃO

### **“COMO APRESENTAR PROGRAMAÇÃO PARA CRIANÇAS? ENVOLVER PARA DESMISTIFICAR, VALORIZAR PARA IMPLEMENTAR”**

por

**“Lucas Cyulik”**

Este Trabalho de Conclusão de Curso foi apresentado às **14 hs** do dia **15 de dezembro** de **2016** como requisito parcial à obtenção do grau de Bacharel em Sistemas de Informação na Universidade Tecnológica Federal do Paraná - UTFPR - Câmpus Curitiba. O aluno foi arguido pelos membros da Banca de Avaliação abaixo assinados. Após deliberação a Banca de Avaliação considerou o trabalho **Aprovado**.

<hr/> <p><b>Prof. Marília Abrahão Amaral</b> (Presidente - UTFPR/Curitiba)</p>	<hr/> <p><b>Prof. Maria Claudia Figueiredo Pereira Emer</b> (Avaliador 1 - UTFPR/Curitiba)</p>
<hr/> <p><b>Prof. Mariangela de Oliveira Gomes Setti</b> (Avaliador 2 - UTFPR/Curitiba)</p>	<hr/> <p><b>Prof. Leyza Elmeri Baldo Dorini</b> (Professor Responsável pelo TCC – UTFPR/Curitiba)</p>
<hr/> <p><b>Prof. Leonelo Dell Anhol Almeida</b> (Coordenador(a) do curso de Bacharelado em Sistemas de Informação – UTFPR/Curitiba)</p>	

“A Folha de Aprovação assinada encontra-se na Coordenação do Curso.”

## **AGRADECIMENTOS**

Primeiramente agradeço a Deus e a Virgem Maria, que durante essa caminhada sempre me guiaram e iluminaram, dando força e conforto para enfrentar todas dificuldades e provações que apareceram.

Agradeço as professoras Mariangela de Oliveira Gomes Setti e Maria Claudia Figueiredo Pereira Emer pela disponibilidade e o esforço nas correções durante a banca de TCC 1 e TCC 2 e no decorrer desta monografia, assim como nas trocas de informações que fizeram diferença no decorrer do trabalho.

Agradeço acima de tudo a minha orientadora Marília Abrahão Amaral por sempre estar do meu lado, apoiando e incentivando, nunca me deixando desistir e nem desviar do caminho, mesmo nos momentos de maiores dificuldades. Sou grato por todas as horas de orientação e pela extrema dedicação para com seus alunos. Esse trabalho não seria possível sem a ajuda dela.

Grato aos meus pais, que sempre me apoiaram em todos os momentos, sempre me ajudando quando pensava que não conseguiria. Também sou grato aos colegas que em meio a conversas sugeriram mudanças e incrementos de grande valia neste trabalho.

*“Deixando a vida futura de estar velada pela dúvida, o homem perceberá melhor que, por meio do presente, lhe é dado preparar o futuro. ”*

Chico Xavier

## RESUMO

CYULIK, Lucas. Como Apresentar Programação para Crianças? Envolver para Desmistificar, Valorizar para Implementar. 2016. 68 f. Trabalho de Conclusão de Curso – Bacharelado em Sistemas de Informação, Universidade Tecnológica Federal do Paraná. Curitiba, 2016.

Esta pesquisa apresenta uma proposta de desenvolvimento de um jogo para ensino de programação para crianças com técnicas participativas, que permitem colocar nas mãos do usuário o desenvolvimento do artefato. Para isso, foi proposta a aplicação de técnicas participativas com indivíduos pertencentes ao público-alvo, crianças estudantes do ensino básico e fundamental, com faixa etária de 5 a 9 anos. As análises das atividades realizadas permitiram o desenvolvimento de um modelo que originou um jogo desenvolvido utilizando a ferramenta Construct 2. Com o artefato foi possível realizar uma validação com profissionais da área da educação para averiguar se houve, ou não, sucesso no desenvolvimento do jogo.

**Palavras-chave:** Ensino de Programação. Técnicas Participativas. Design Participativo. Jogos Educativos.

## LISTA DE FIGURAS

FIGURA 1 - Scratch Web site (Fonte: <a href="http://scratch.mit.edu/">http://scratch.mit.edu/</a> ) .....	21
FIGURA 2 - Ambiente de Programação Scratch (Fonte: <a href="http://scratch.mit.edu/">http://scratch.mit.edu/</a> ) .....	22
FIGURA 3 - Fluxo de atividades (Fonte: Autoria Própria) .....	37
FIGURA 4 - Roboquedo (Fonte: Grupo PET-CoCE).....	40
FIGURA 5 - Interação com o Roboquedo (Fonte: Autoria Própria) .....	41
FIGURA 6 - Desenvolvimento da Técnica PICTIVE (Fonte: Autoria Própria) .....	43
FIGURA 7 - Desenho Grupo A (Fonte: Autoria Própria).....	45
FIGURA 8 - Desenho Grupo B (Fonte: Autoria Própria).....	46
FIGURA 9 - Mockup preliminar, Fases (Fonte: Autoria Própria) .....	47
FIGURA 10 - Mockup preliminar, Jogo (Fonte: Autoria Própria).....	48
FIGURA 11 - Elementos: Caminho, Setas, Lista e Botão Executar (Fonte: Autoria Própria) .....	49
FIGURA 12 - Telas do jogo (Fonte: Autoria Própria).....	50
FIGURA 13 - Telas de tutorial (Fonte: Autoria Própria).....	50
FIGURA 14 - Construct 2 (Fonte: <a href="https://www.scirra.com/">https://www.scirra.com/</a> ).....	51
FIGURA 15 - Exemplo de Layout e Event Sheet (Fonte: <a href="https://www.scirra.com/">https://www.scirra.com/</a> ) .....	52
FIGURA 16 - Event sheet Fase 1 (Fonte: Autoria Própria) .....	54
FIGURA 17 - Componentes do Menu de Fases (Fonte: Autoria Própria).....	55
FIGURA 18 - Componentes do Jogo Fase 1 (Fonte: Autoria Própria).....	56

## **LISTA DE ABREVIATURAS E SIGLAS**

**ACM DL:** *Association for Computing Machinery Digital Library*

**CARD:** *Collaborative Analysis of Requirements Design*

**LLK:** *Lifelong Kindergarten*

**MIT:** *Massachusetts Institute of Technology*

**Pert/CPM:** *Program Evaluation and Review Technique/Critical Path Method*

**PET-CoCE:** Programa de Educação Tutorial - Computando Culturas em Equidade

**PICTIVE:** *Plastic Interface for Collaborative Technology Initiatives through Video Exploration*

**TCC:** Trabalho de Conclusão de Curso

**TI:** Tecnologia da Informação

**UTFPR:** Universidade Tecnológica Federal do Paraná



## SUMÁRIO

<b>1. INTRODUÇÃO.....</b>	<b>11</b>
1.1. OBJETIVO GERAL .....	12
1.2. OBJETIVOS ESPECÍFICOS .....	12
1.3. MOTIVAÇÃO.....	13
1.4. ORGANIZAÇÃO DO DOCUMENTO.....	14
<b>2. REFERENCIAL TEÓRICO .....</b>	<b>15</b>
2.1. JOGOS DIGITAIS NA EDUCAÇÃO.....	15
2.2. ENSINO DE COMPUTAÇÃO .....	17
2.2.1. <i>Scratch</i> .....	20
2.3. DESIGN PARTICIPATIVO .....	23
2.3.1. <i>Interface Metaphors Game</i> .....	27
2.3.2. <i>CARD Game</i> .....	28
2.3.3. <i>Icon Design Game</i> .....	28
2.3.4. <i>Pictive</i> .....	29
2.3.5. <i>Interface Theater</i> .....	30
2.4. TRABALHOS CORRELATOS.....	30
<b>3. METODOLOGIA.....</b>	<b>35</b>
3.1. CARACTERIZAÇÃO DA PESQUISA .....	35
3.2. REFERENCIAL TEÓRICO.....	35
3.2.1. <i>Pesquisa Bibliográfica</i> .....	35
3.2.2. <i>Revisão de Literatura</i> .....	36
3.3. DEFINIÇÃO DO PÚBLICO-ALVO.....	36
3.4. DEFINIÇÃO DAS TÉCNICAS E MÉTODOS DE DESIGN PARTICIPATIVO .....	37
3.5. PASSOS ESTABELECIDOS .....	37
<b>4. AS ATIVIDADES .....</b>	<b>39</b>
4.1. PLANEJAMENTO.....	39
4.1.1. <i>Roboquedo</i> .....	40
4.2. O QUE ACONTECEU .....	41
4.2.1. <i>PICTIVE</i> .....	42
4.2.2. <i>Inteface Metaphors Game + CARD Game</i> .....	43
4.2.3. <i>Resultados das Aplicações das Técnicas</i> .....	44
4.3. ANÁLISE.....	46
<b>5. ARTEFATO .....</b>	<b>49</b>
5.1. PROJETO.....	49
5.1.1. <i>Construct 2</i> .....	51
5.2. DESENVOLVIMENTO.....	53
5.3. ARTEFATO X TÉCNICAS.....	54
<b>6. VALIDAÇÃO PRÉVIA .....</b>	<b>57</b>
6.1. LEGIBILIDADE .....	57

6.2.	JOGABILIDADE.....	58
6.3.	APRENDIZAGEM.....	59
6.4.	SUGESTÕES DE MUDANÇA .....	59
<b>7.</b>	<b>CONCLUSÕES.....</b>	<b>60</b>
7.1.	TRABALHOS FUTUROS.....	61
	<b>REFERÊNCIAS.....</b>	<b>62</b>
	<b>APÊNDICE A – ENTREVISTA GUIADA .....</b>	<b>68</b>

## 1. INTRODUÇÃO

A língua artificial esperanto, criado pelo médico judeu Ludwik Lejzer Zamenhof no fim do século 19, surgiu com a promessa de se tornar uma língua que unificaria as nações, de modo que todos os povos se comunicassem usando o idioma criado (BOULTON, 1980). Porém, a popularidade desejada não foi atingida, e tempos depois a língua inglesa emergiu como uma nova esperança de uma língua unificadora, no entanto a quem diga que são as linguagens de programação que deveriam receber esse título (GIBSON, 2012).

Para Gibson (2012) nos próximos 50 anos a computação deve atingir o patamar de linguagem universal, para tanto ele se baseia no crescimento da robotização dos empregos, mudando a antiga dinâmica de trabalho. Sendo assim a programação se torna uma ferramenta de aprendizado fundamental nas escolas, permitindo aos jovens lidarem melhor com as futuras mudanças no mercado, e desenvolvendo assim uma plataforma comum para a comunicação e entendimento (SEEHORN et al., 2011).

Não só no ambiente corporativo, mas em outros contextos existem *softwares* envolvidos, seja na agricultura, ou nas áreas do entretenimento, a computação está presente, mesmo quando um automóvel é ligado, ou o micro-ondas é utilizado por alguém para fazer pipoca. Devido a essa grande presença em algum momento os profissionais das mais diversas profissões terão contado com programadores e assim é importante conseguir se expressar na mesma linguagem, e acima de tudo entender as possibilidades e limitações que a computação apresenta (SEEHORN et al., 2011).

A programação acabou por se tornar uma questão cultural, necessária para o entendimento do mundo, e assim deve ser tratada como qualquer outro assunto do cotidiano (VALENTE et al., 1999). Se no colégio são ensinados conteúdos curriculares de biologia, não é para que todas as pessoas se tornem profissionais da área médica, mas para que tenham conhecimentos básicos e não sejam totalmente leigas quando recebem e emitem informações, por exemplo, do cotidiano, como Zica ou Dengue. Seguindo essa lógica a programação pode ser ensinada como uma ferramenta de apropriação da tecnologia, e não necessariamente para que estudantes se tornem profissionais da área de computação.

No entanto, quando se fala de ensino de computação nas escolas a linguagem de programação muitas vezes não é o ponto principal a ser atingido, o processo de aprendizagem

que a programação proporciona no desenvolvimento do raciocínio lógico é o principal desdobramento (WING, 2012). A programação de alto nível, aquela que está mais próxima da linguagem humana, tornou os códigos mais legíveis e a disposição das mais diversas linguagens na *internet* tornou o aprendizado das mesmas mais difundido, sendo assim um ponto importante é ensinar as crianças<sup>1</sup> como as estruturas funcionam e como resolver os problemas (FRANÇA; SILVA; AMARAL, 2012).

Segundo Gibson (2012), assim que as crianças começam o processo de alfabetização em seu idioma já podem ser introduzidas a programação. Esse relato se baseia no fato de que crianças mais jovens aprendem novas linguagens mais facilmente do que crianças mais velhas (JOHNSON; NEWPORT, 1989). Apesar de parecer complexo, o início do aprendizado de programação costuma ser lúdico, por meio de jogos e brincadeiras que buscam mostrar como funciona a lógica que permeia os códigos (SCAICO et al., 2012b).

Quando se trata da criação de sistemas computacionais para o uso das pessoas, nos últimos anos surgiram diferentes paradigmas de *design* que inserem o usuário final no processo de concepção, entre eles o Design Participativo (MELO; BARANAUSKAS, 2006). O uso de técnicas participativas busca facilitar o desenvolvimento, por meio da coleta, análise e realização do projeto juntamente com a participação de usuários, desenvolvedores e outros interessados (CAMARGO; FAZANI, 2014). A inclusão propiciada pelo Design Participativo permite a visualização da diversidade e multiplicidade de usuários, evidenciando diferentes ideias e como elas interagem (MELO; BARANAUSKAS, 2006)

Diante deste contexto, esta proposta, tem como objetivos as ações listadas nas próximas subseções.

### 1.1. OBJETIVO GERAL

Apresentar programação para crianças utilizando técnicas participativas.

### 1.2. OBJETIVOS ESPECÍFICOS

Como objetivos específicos, tem-se:

- Definir o público-alvo e quais serão os participantes das seções colaborativas/participativas;

---

<sup>1</sup> Para este trabalho, de acordo com Gibson (2012) e França, Silva e Amaral (2012), entende-se como criança pessoas da faixa etária de 2 a 15 anos e que estão em processo de alfabetização.

- Elencar técnicas participativas para apoiar o desenvolvimento do artefato;
- Propor o desenvolvimento de um artefato, do tipo jogo educacional, a partir das técnicas de Design Participativo;
- Validar qualitativamente o artefato proposto.

### 1.3. MOTIVAÇÃO

A educação como era tratada nos séculos passados se difere de hoje, o fluxo de informações aos quais os estudantes são expostos é maior do que nas últimas décadas, a *internet* proporcionou a busca por conteúdos sobre os mais diversos assuntos (COUTINHO; LISBÔA, 2011). São diversas as ferramentas e formas de ensino encontradas no mercado, entre elas estão os jogos digitais, apontados como uma ferramenta que se enquadra na realidade dos estudantes de hoje, e que ajudam a otimizar o desenvolvimento cognitivo e lógico do indivíduo. (ADACHI; WILLOUGHBY, 2013).

Quando se trata do ensino de computação por meio de jogos digitais existem algumas ferramentas no mercado que se destacam, como o Scratch, uma linguagem de programação desenvolvida em 2003 no *Media Lab* do *Massachusetts Institute of Technology* (MIT), que de uma maneira lúdica busca apresentar conceitos computacionais e matemáticos para crianças acima dos oito anos de idade (RESNICK, 2009).

Apesar de existirem jogos digitais de qualidade reconhecida, é raro encontrar um destinado ao ensino introdutório de computação para crianças que possuem algum ou nenhum conhecimento linguístico ou de matemática (TAROUCO et al., 2004). Visto isto constatou-se a necessidade de desenvolver um modelo de jogo digital, que de forma lúdica introduzisse conceitos simples de computação a crianças, buscando assim o desenvolvimento do raciocínio lógico e de tomada de decisões.

Dentro do cenário descrito anteriormente, o Design Participativo surge como uma opção de abordagem democrática, que por meio do uso de métodos colaborativos permite o desenvolvimento de um modelo que busca refletir a visão do público-alvo. Assim é possível fugir do *design* solitário, que é concebido a partir da visão do desenvolvedor ou *designer* sobre como o usuário deve utilizar o modelo ou artefato proposto.

#### 1.4. ORGANIZAÇÃO DO DOCUMENTO

O presente trabalho está organizado da seguinte maneira: primeiramente, o levantamento bibliográfico sobre o tema do projeto é apresentado. Depois, a metodologia de realização do trabalho é exposta. Posteriormente, as atividades realizadas são explanadas e o modelo explicado. A seguir, expõe-se o processo de desenvolvimento do artefato. Após, é apresentada a validação prévia do artefato desenvolvido. Por fim, as considerações finais do trabalho e as referências bibliográficas são apresentadas.

## 2. REFERENCIAL TEÓRICO

Na presente seção é abordado o referencial teórico, o qual é usado como base para pesquisa, e está distribuído da seguinte forma: 2.1 Jogos na Educação; 2.2 Ensino de Computação; 2.3 Design Participativo; 2.4 Trabalhos Correlatos.

### 2.1. JOGOS DIGITAIS NA EDUCAÇÃO

Desde a infância até a vida adulta, os jogos fazem parte do cotidiano de uma parcela dos seres humanos. Ao mesmo tempo que divertem os jogos motivam, aumentando assim a capacidade de retenção do que foi ensinado, logo podendo ser usados como uma ferramenta institucional eficiente (TAROUCO et al., 2004). As áreas cognitiva, motora, social, linguística e moral, são desenvolvidas de maneira mais dinâmica com uso de jogos, além de estimular a criatividade e cooperação de crianças e adolescentes (MORATORI, 2003).

A reação mais comum para os estudantes quando são propostas atividades com jogos é de alegria, o estudante se sente estimulado pelas regras ou desafios impostos, porém é preciso tomar cuidado com essa empolgação, que muitas vezes é ligada apenas a oportunidade de realizar atividades não vinculadas a sala de aula (MORATORI, 2003). Para Ronca e Escobar (1980), os jogos não devem ser vistos pelos docentes como forma de distração durante uma aula, mas sim como parte do planejamento do ensino visando o aperfeiçoamento e aumento da aprendizagem. Como os jogos permitem a prática significativa daquilo que está sendo aprendido, são um método eficaz para o aumento da motivação e da autoconfiança durante o processo de aprendizado (FERNANDES; FURQUIM; BARANAUSKAS, 1995).

A popularidade dos jogos aumentou durante os anos, os investimentos na área de jogos digitais demonstram que sua incorporação em atividades não comumente aplicadas, como ensino, não seria algo forçado, mas natural diante do atual cenário tecnológico (FARDO, 2013). As atuais gerações cresceram em contato com jogos digitais o que torna a aceitação desse tipo de entretenimento em outros campos da atividade humana mais natural, e também justifica a implementação de jogos a partir da perspectiva sociocultural (COLOMBO, 2015). Segundo Tarouco et al. (2004), com o rompimento do paradigma de ensino tradicional, e o surgimento do construtivismo, foi aberto espaço para o início da utilização da tecnologia no contexto educacional.

Os jogos digitais apresentam um sentido amplo de interação, pois possibilitam o aumento da experiência do estudante e impactam de maneira positiva sua visão sobre o assunto (FARDO, 2013). Por serem lúdicos, os jogos digitais necessitam ser atraentes, dando assim importância a narrativa, visto que esta tem a característica de envolver o jogador em uma espécie de universo paralelo, assimilando o conteúdo a ser aprendido (MARTINS et al., 2014). Jogar é participar de um universo de faz de conta, que exige do estudante autonomia e criatividade ao enfrentar os problemas desse mundo (TAROUCO et al., 2004).

Os jogos digitais apresentam uma imersão que pode ser comparada a literatura e ao cinema, quando se trata do enredo e da complexidade de elementos (MARTINS et al., 2014). A faixa etária que constitui o público-alvo é tão importante quanto a conteúdo que pretende-se ensinar, quando se trata da concepção de jogos digitais para ensino (MORATORI, 2003).

Uma das principais características dos jogos digitais é o *feedback* imediato interativo, que permite que o estudante se auto-avale e rapidamente possa mudar suas estratégias buscando o acerto (SALEN; ZIMMERMAN, 2012). Para tanto, Moratori (2003) diz que os jogos devem propiciar situações desafiadoras quanto a resolução de problemas, para haver o real envolvimento com atividade.

É necessário dar uma clara recompensa ao jogador, que o estimule a se empenhar na tarefa proposta (MARTINS et al., 2014). As estratégias de recompensas permitem ao docente ter uma visão geral sobre o desempenho dos estudantes e aos estudantes terem um reconhecimento por suas conquistas (SEIXAS et al., 2014).

Dentro das salas de aula existe uma velada disputa travada entre docentes e tecnologia pela atenção dos estudantes, tornando assim um desafio ao sistema de ensino adaptar as práticas pedagógica para se adequarem as mudanças que as tecnologias digitais impõem (MARTINS et al., 2014). Os jogos digitais são baseados em uma abordagem autodirigida, em que o estudante aprende por si só, tendo o docente como moderador do processo de aprendizado (TAROUCO et al., 2004).

Acima de tudo os jogos digitais devem ser vistos como uma ferramenta de auxílio no processo de ensino no qual os estudantes são vistos como agentes ativos na aprendizagem, mas não deixando de lado a importância do docente, visto que a interferência positiva do mesmo é importante durante o acompanhamento do estudante (MORATORI, 2003). Segundo Tarouco et al. (2004), é importante os docentes terem domínio da tecnologia, afim de atender o objetivo de ter os jogos como ferramenta de auxílio, bem fundamentada e criteriosa.



O processo de *design* de jogos educacionais, segundo Figueiredo et al. (2015), pode ser facilmente adaptado para o ensino de programação, assim como qualquer disciplina de computação.

## 2.2. ENSINO DE COMPUTAÇÃO

O Brasil apresenta um déficit de profissionais da área de computação (BARCELOS; SILVEIRA, 2012). O profissional da computação é aquele que analisa problemas em diversas áreas e propõem soluções computacionais para resolvê-los, para tanto é preciso ter obtido um bom conhecimento em sala de aula para aplicar no desenvolvimento da solução (DA SILVA; MARTINS, 2004).

Quando se trata do aprendizado de computação, a programação é uma das disciplinas essenciais, e que constitui a base para muitos campos da área de informática (MOREIRA; FAVERO, 2009). Uma das bases para o ensino de programação é a disciplina algoritmos, que aborda princípios da lógica de programação, visando o aprendizado da capacidade do estudante de analisar e resolver problemas (RAPKIEWICZ et al., 2006). Porém, disciplinas como lógica e algoritmos costumam ter altos índices de evasão e reprovação, sendo necessário que os docentes dessa disciplina forneçam propostas que busquem diminuir esses índices (HINTERHOLZ, 2009).

Entre os problemas apontados para os altos índices de evasão em cursos de computação estão falta de motivação e desânimo com matérias que são etiquetadas como difíceis, além da dificuldade no aprendizado do raciocínio lógico, visto que os estudantes estão acostumados a decorar os conteúdos (RODRIGUES, 2002). Algumas vezes não é claro para os estudantes a importância de determinado conteúdo na sua formação, o que o leva a falta de interesse (BORGES, 2000). Em outros casos o início lúdico do ensino de programação desestimula os estudantes, visto que os que estes buscam cursos de computação querendo interagir com computador logo de cara (DA SILVA; MARTINS, 2004). Outro problema está na dificuldade que os estudantes têm em entender as abstrações que envolvem o processo de aprendizado de programação (JÚNIOR; RAPKIEWICZ, 2004).

Em um estudo realizado por Setti (2009) sobre desenvolvimento do raciocínio computacional, constatou-se que mesmo alguns alunos possuindo habilidade em resolver problemas matemáticos, esta não se mostra suficiente na elaboração do raciocínio algorítmico, a dificuldade em particionar o raciocínio matemático, aliada à dificuldade em perceber a regularidade das situações apresentadas, constituem uma barreira na aprendizagem de

algoritmos. A passagem de uma representação em linguagem natural, ou algébrica, para a correspondente em linguagem computacional é mais um fator produtor de obstáculos à aprendizagem, visto que a falta de equivalências na passagem da linguagem natural, ou algébrica, para linguagem computacional (SETTI, 2009).

Dentro da sala de aula o processo de aprendizagem é composto por quatro elementos: docente, estudante, conteúdo e variáveis ambientais, sendo que todas se relacionam, exercendo assim menor ou maior influência (MOREIRA, 1986). O entusiasmo do docente é essencial na hora do ensino, de maneira que o estudante se sinta estimulado a aprender, a maneira como o docente age dentro de sala repercute no processo de aprendizagem (SANTOS, 2001). As práticas de ensino devem ser condizentes com os interesses dos estudantes, tornando assim a aprendizagem ativa<sup>2</sup>, uma boa ferramenta para isso é o *feedback* imediato, que permite ao estudante acompanhar e corrigir suas atitudes de aprendizado quando necessário (SANTOS, 2001).

Uma característica do ensino superior no Brasil, e mesmo do exterior, é falta da integração curricular, deve-se buscar que as atividades de ensino e pesquisa se integrem na formação de um profissional de computação, sendo que a *internet* tem um papel fundamental nisso (DIMURO; COSTA; RODRIGUES, 2000). A forma como a computação é fundamentada se difere de outras áreas de ensino, quando estuda programação o estudante aprende mecanismos de raciocínio que auxiliaram na resolução de problemas, que muitas vezes ultrapassam os conceitos de computação, um exemplo é o uso de redes neurais da biologia para resolução de problemas usando inteligência artificial (BARCELOS; SILVEIRA, 2012).

Segundo Wing (2012) as ciências da computação deveriam ser consideradas uma ciência básica, de forma que deveriam ser introduzidos desde o ensino básico, para estimular o pensamento computacional, o que facilitaria na hora da abstração e resolução de problemas utilizando algoritmos, e também fomentaria o interesse pela área. Em um mundo com cada vez mais presença de dispositivos computacionais, o pensamento computacional surge como uma ferramenta de domínio da tecnologia, e se torna uma vantagem quando aprendido desde cedo (BARCELOS; SILVEIRA, 2012).

O ensino de programação na educação básica estimula habilidades como as capacidades de resolução, simplificação e interpretação de problemas, estimulando o raciocínio lógico do

---

<sup>2</sup> Para Santos (2001), aprendizagem ativa é uma prática pedagógica, no qual o estudante é visto como agente pró-ativo no processo de aquisição do conhecimento, e não um mero “recedor” de informações.

estudante, e geralmente é feito com o auxílio de jogos (SCAICO et al., 2012b). O aprendizado de programação no ensino básico proporciona, por meio do ensino da resolução de problemas e abstrações, o desenvolvimento de capacidades úteis a futuros profissionais das mais diversas áreas (SCAICO et al., 2012a).

Para Papert (1994), o ensino da programação para crianças, permite uma autonomia que até antes apenas os adultos podiam ter, ao programarem um computador as crianças se sentem no controle da máquina. Outro ponto citado é que as mais diversas áreas de atividade humana, como medicina e engenharia, passam por constantes mudanças e evolução, algo que pouco se nota no sistema de ensino, em que as tecnologias deveriam ser mais usadas e ensinadas (PAPERT, 1994).

Um exemplo de iniciativa para o ensino do pensamento computacional é o material de Computação Desplugada, que por meio de jogos com cartas, tabuleiros, etc. busca ensinar conceitos de computação sem o uso do computador (BELL et al., 2009). As atividades *desplugadas* ao mesmo tempo que buscam desenvolver o pensamento computacional, proporcionam a abstração dos conhecimentos apresentados, o que é muito positivo em um passo introdutório de ensino de computação para crianças (SCAICO et al., 2012b). Estas atividades também permitem remover barreiras econômicas, que são um grande problema no Brasil, já que não fazem uso do computador (BELL; WITTEN; FELLOWS, 2011).

Em uma pesquisa realizada com crianças de 5 a 12 anos diagnosticadas com altas habilidades<sup>3</sup> com características de inteligência lógico-matemática, foi constatado que o material de Computação Desplugada despertou interesse nas crianças, que conseguiram assimilar os conceitos de computação ensinados de maneira simples e rápida, além do fator de que os jogos prenderem a atenção dos participantes, que permaneceram focados na tarefa durante a aplicação da atividade (ANTUNES et al., 2014).

O início do ensino de programação pode ser de diversas maneiras, como: escolha de uma linguagem de programação próxima a natural; o uso de linguagem natural, por meio dos pseudocódigos; pelo uso de formas gráficas, como diagramas e fluxogramas (JÚNIOR et al.,

---

<sup>3</sup> Na pesquisa de Antunes et al. (2014), foram adotados os conceitos de Gardner (1989) para definição de pessoas diagnosticadas com altas habilidades ou superdotação, assim, diz-se que são aquelas que apresentam desempenho intelectual elevado, que pode ser caracterizado segundo a teoria de Inteligências Múltiplas de Gardner (1989) em áreas de inteligência, como linguística, lógico-matemática, espacial, interpessoal, intrapessoal, musical e corporal-cinestésica.

2005). Independentemente do método escolhido para introduzir programação, é comum que o início do aprendizado seja sem o uso de computador (JÚNIOR et al., 2005).

Para programar é utilizada uma linguagem formal, ou seja, aquela que é bem interpretada e sem ambiguidade, visto que um computador trabalha com um conjunto de instruções a serem seguidas, e que não podem ter margem para múltiplas interpretações (JÚNIOR et al., 2005). A formalização das linguagens de programação é muitas vezes uma dificuldade enfrentada pelos estudantes, por isso é comum o início da aprendizagem se dar pelo desenvolvimento de algoritmos em linguagem natural, até que estudante consiga definir bem as especificações da solução (JÚNIOR et al., 2005).

Para que o conteúdo seja assimilado mais facilmente, e torne a aula atrativa e participativa a utilização de novas ferramentas costuma funcionar (HINTERHOLZ, 2009). Entidades abstratas como *loops*, ponteiros, vetores, tornam o aprendizado de conceitos iniciais de programação difícil, para tanto é comum o uso de métodos, técnicas e ferramentas que buscam auxiliar os estudantes no processo de aprendizagem (SAJANIEMI; KUITTINEN, 2003). O uso de ferramentas que auxiliem no aprendizado de programação de forma didática é uma boa opção, visto a importância dessa matéria nos cursos de computação (MOTA; PEREIRA; FAVERO, 2008).

Segundo Papert (1994) jogos podem ser uma ferramenta adequada, pois unem diversão e aprendizagem, além de que proporcionam a independência para criança aprender sem um adulto. São também ferramentas lúdicas que podem ser interessantes para início do ensino de programação (COLOMBO, 2015). O material de Computação Desplugada, é um exemplo do uso de jogos para ensino de programação, porém também existem linguagens digitais que desempenham esse papel, como o Scratch, descrito na Seção 2.2.1

### 2.2.1. Scratch

O Scratch é uma linguagem de programação visual *open source*, criada com a finalidade de tornar o aprendizado de programação mais simples e divertido. Tendo sido desenvolvido em 2003 pelo *Lifelong Kindergarten (LLK)*, um grupo do *Massachusetts Institute of Technology Media Lab* (<http://llk.media.mit.edu>), com apoio de fundos da *National Science Foundation*, Microsoft, Intel *Foundation* e Nokia (RESNICK, 2009).

Ele é ideal para pessoas que estão começando a programar, pois não exige conhecimento prévio de programação, e diferente da maioria das linguagens, que são baseadas em texto, o

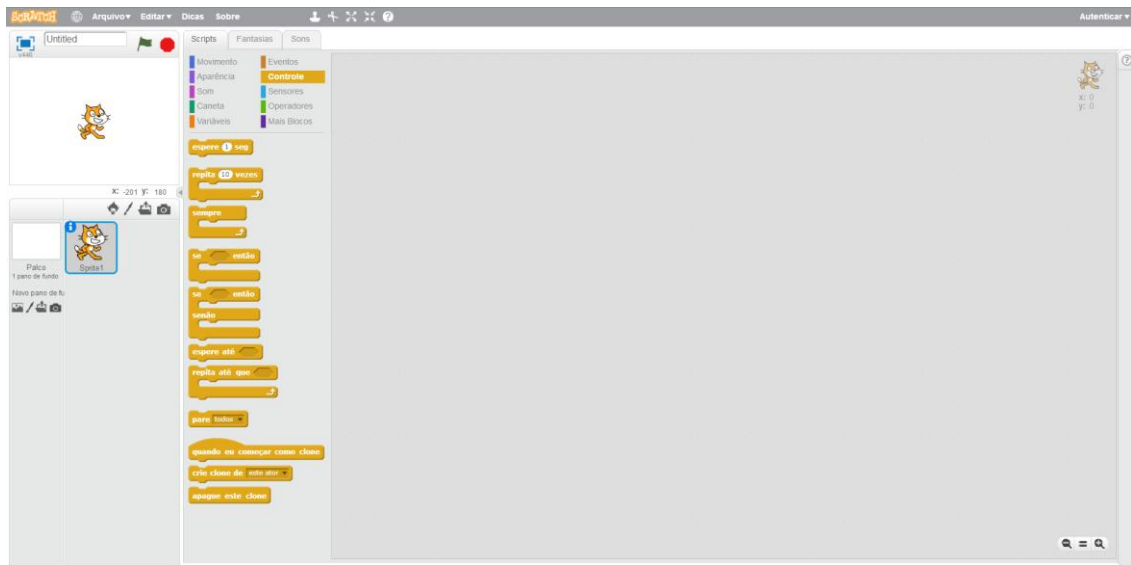
Scratch usa programação inteiramente visual (MALONEY et al., 2010). Apesar de ser visual ele utiliza a sintaxe comum a muitas linguagens de programação (MARJI, 2014).

A programação é efetuada por meio da criação de seqüências de comandos simples, que por se utilizar de uma *interface* gráfica correspondem a blocos de montar, que são encaixados e encadeados para produzir as ações desejadas, sendo possível trabalhar com imagens, fotos, desenhos, músicas, entre outras mídias (MARJI, 2014).

Desde 2013, quando foi lançado, o Scratch 2 está disponível *on-line* e como aplicação para Windows, OS X, e Linux. Para fazer *download* basta acessar o seu site (<http://scratch.mit.edu/>) e clicar no *link* EXPERIMENTE, como é apresentado na Figura 1. Também é possível partilhar as criações por meio de *websites*, inclusive um repositório chamado ScratchR do mesmo grupo que desenvolveu o jogo (MONROY-HERNÁNDEZ, 2007).

Figura 1 - Scratch Web site (Fonte: <http://scratch.mit.edu/>)

O ambiente de programação Scratch, apresentado na Figura 2, busca ser de fácil interação, dividido em 3 partes principais: o *Stage* (na parte superior esquerda), é apresentado o resultado da execução; a *Sprite List* (na parte inferior esquerda), são exibidos o nome e a miniatura de todos os *sprites* (objetos) do projeto; e a *Aba de Scripts*, que contém a aba *Blocks* (no centro), os blocos de programação estão presentes, e a *Script Area* (na direita), em que cabem os blocos de comando (MARJI, 2014).



**Figura 2 - Ambiente de Programação Scratch (Fonte: <http://scratch.mit.edu/>)**

No Brasil, em uma pesquisa com estudantes de graduação, o Scratch apresentou bom desempenho como uma abordagem alternativa ao processo de ensino introdutório de programação, entre os pontos positivos encontrados tem-se que o Scratch permite a visualização do código em tempo de execução, facilitando assim a assimilação do conteúdo (AURELIANO; TEDESCO, 2012). Outro estudo, também realizado com estudantes de ensino superior, levantou o ponto de que o Scratch oferece a possibilidade de aplicação em diversas áreas de tecnologia, como eletrônica e mecânica (MÉLO et al., 2011).

Pesquisas realizadas com estudantes de ensino fundamental e médio se mostraram ainda mais promissoras. Para estudantes do ensino fundamental a ferramenta despertou interesse e afinidade, e facilitou a compreensão dos conceitos ensinados no quadro (DE OLIVEIRA et al., 2014; DOS SANTOS; JÚNIOR, 2015). Os estudantes de Ensino médio, destacaram a linguagem e o ambiente de programação como sendo atrativos, assim havendo um aumento do interesse, e aplicando ao conteúdo ensinado, tornou mais claro o aprendizado dos conceitos (ALENCAR et al., 2014; SCAICO et al., 2013).

### 2.3. DESIGN PARTICIPATIVO

O Design Participativo é visto como uma área em amadurecimento e constante evolução entre os profissionais do *design* (KENSING; BLOMBERG, 1998). Os pesquisadores da área de Design Participativo exploram condições para participação do usuário na concepção e aplicação de sistemas baseados em computador no trabalho (KENSING; BLOMBERG, 1998). Este campo apresenta um ambiente animado de debates sobre as relações adequadas e práticas entre trabalho e tecnologia, e sobre as técnicas usadas para análise e projeto de tais relações (KENSING; BLOMBERG, 1998).

As abordagens participativas para projetos são defendidas como substitutos de projetos sistemáticos ou racionalistas no movimento da metodologia de *design* (EHN, 1988). Desde os primeiros dias de existência do Design Participativo, os sistemas baseados em computadores vêm se tornando cada vez mais parte integrante da vida profissional das pessoas, o que gerou debates nas décadas de 70 e 80 (KENSING; BLOMBERG, 1998).

Em meados de 1970 as abordagens de Design Participativo foram o centro do debate sobre métodos de concepção de projetos, e os argumentos em sua maioria eram teóricos, porém também práticos e políticos, tanto o existencialismo como o marxismo tinham influência na formação dos métodos de concepção daquela geração (EHN, 1988). No início de 1980, os *designers* de *software* escandinavos buscaram fazer os projetos de sistemas mais participativos e democráticos, o que os levou para prototipagem (SPINUZZI, 2002).

O chamado “desafio escandinavo” de fazer computadores mais democráticos inspirou outros que se tornaram interessados no *design* centrado no usuário, *designers* de ambos os lados do Atlântico começaram a empregar prototipagem como uma forma de incentivar a participação do usuário e como *feedback* de outras abordagens de *design* (SPINUZZI, 2002). Eles tinham como desafio desenvolver métodos de *design* realmente participativos, para tanto começaram a ser usadas simulações, como protótipos, maquetes e jogos organizacionais, o que permitiu aos *designers* gráficos articularem suas demandas de forma concreta, fazendo alterações nessas simulações (EHN, 1988).

No entanto, a prototipagem foi vista e implementada de maneira diferente na Escandinávia e nos Estados Unidos, assim como os resultados, que tendiam a ser interpretados de maneiras diferentes (SPINUZZI, 2002). Foram separadas assim duas linhas de pesquisa de Design Participativo, uma escandinava e outra americana (SPINUZZI, 2002).

A linha de pesquisa escandinava tem um marco na Suécia, em 1981, com o projeto UTOPIA, em que os pesquisadores viram o local de trabalho como um conflito entre capital (empregadores) e trabalho (empregados), em que os empregadores tinham vantagem (SPINUZZI, 2002). Apoiados pela União, os acadêmicos procuraram fornecer uma maneira pró-ativa para os trabalhadores para implementar a democracia industrial no seu local de trabalho, sendo que uma parte importante dessa estratégia foi a prototipagem, que fornecia uma linguagem comum para acadêmicos e trabalhadores, uma forma de tentar soluções iterativamente, e uma maneira para que os trabalhadores desenvolvessem democraticamente consenso sobre o que queriam de um sistema computadorizado (SPINUZZI, 2002).

Já a linha americana de pesquisa, começou nos Estados Unidos, em 1998, quando foi desenvolvido uma abordagem chamada Design Contextual, destinada a gestores e pessoal de TI (Tecnologia da Informação) de empresas norte-americanas, essa abordagem descrevia um projeto contextual como uma abordagem de *design* "centrado no cliente", que gira em torno de observar os aspectos do trabalho dos funcionários, desenvolver modelos de trabalho baseado nas observações, e fazer os novos *designs* de sistema sobre os modelos desenvolvidos (SPINUZZI, 2002). Uma vez que estes sistemas eram descritos e os protótipos construídos, os *designers* mantinham sessões com os trabalhadores para obter *feedback* e reunir ideias, sendo assim, embora denominados "co-criadores", os trabalhadores não projetavam o sistema, apenas usavam protótipos (SPINUZZI, 2002).

As duas abordagens de prototipagem possuem muitas diferenças, o projeto UTOPIA foi baseado na teoria marxista, já a abordagem de Design Contextual é enfaticamente capitalista (SPINUZZI, 2002). No projeto UTOPIA, os acadêmicos foram financiados com o dinheiro da concessão do governo; a abordagem de Design Contextual foi solicitada por gestores e pago dos cofres das empresas (SPINUZZI, 2002). O projeto UTOPIA tem o propósito de chamar os trabalhadores para o processo de projeto e torná-los os *designers*, e não apenas para testar, ratificar ou ajustar o trabalho de *designers* externos (SPINUZZI, 2002).

O projeto UTOPIA teve êxito na área acadêmica, porém não comercial (SPINUZZI, 2002). Ele buscava unir habilidade e criatividade na concepção participativa, com o objetivo de apoiar a qualidade do trabalho, no entanto, esse desenvolvimento ocorreu em um ambiente social e histórico escandinavo diferente daquele do debate de metodologia de projeto (EHN, 1988). Embora o projeto não tenha conseguido produzir um sistema de trabalho, ele produziu uma abordagem de *design* e uma variedade de técnicas para o trabalho de Design Participativo (SPINUZZI, 2002). No entanto, acadêmicos e pesquisadores americanos tentaram encontrar



outras maneiras de aplicar a abordagem escandinava, e ao fazer isso, eles modificaram suas técnicas e objetivos (SPINUZZI, 2002).

Segundo Kensing e Blomberg (1998), três questões principais dominam o discurso na literatura de Design Participativo: a política do projeto; a natureza da participação; e os métodos, ferramentas e técnicas para a realização de projetos de *design*. A política do projeto refere-se à introdução de sistemas baseados em computadores e a distribuição de poder no local de trabalho, segundo críticas, os computadores estão se tornando uma ferramenta de gestão de controle sobre a força de trabalho e que as novas tecnologias não estão sendo introduzidas para melhorar as condições de trabalho (KENSING; BLOMBERG, 1998).

Quanto a natureza da participação aborda-se que a avaliação de que os membros da organização devem ser envolvidos na concepção e implementação de tecnologia tem mudado ao longo do tempo, a participação de gerentes e gestores em projetos de *design* tem aumentado e acredita-se que a participação da gestão poderia silenciar as vozes dos trabalhadores (KENSING; BLOMBERG, 1998). Como forma de se conectar com trabalhadores e novas estratégias para envolver os gestores e profissionais de *design* no projeto cooperativo, alguns pesquisadores de Design Participativo começaram a reorientar os seus esforços tanto por meio da cooperação com as pessoas situadas em toda a hierarquia organizacional e não somente trabalhadores e seus sindicatos (KENSING; BLOMBERG, 1998).

Profissionais de *design* e gerentes notam que a habilidade e experiência dos trabalhadores precisa ser levada em conta na concepção e implementação organizacional de sistemas de computador, isso ajuda a garantir um melhor ajuste entre a tecnologia e a forma que a pessoa quer executar o seu trabalho (KENSING; BLOMBERG, 1998).

Quando se trata de métodos, ferramentas e técnicas para a realização de projetos de *design*, os autores descrevem o desenvolvimento de práticas que promovem a cooperação produtiva do *designer*, que por meio de experiências em diversas esferas levou ao desenvolvimento de métodos, ferramentas e técnicas em uso hoje em dia (KENSING; BLOMBERG, 1998).

Os temas explorados em Design Participativo podem ser vistos por meio das lentes dos beneficiários da pesquisa, sendo que os pesquisadores atuam em dois lados: eles estão interessados em projetar tecnologias e práticas experimentais úteis que são formadas por meio de interações com os participantes do local de trabalho, e também estão interessados no desenvolvimento de métodos de Design Participativo mais eficazes e práticas que podem ser adotadas por *designers* profissionais (KENSING; BLOMBERG, 1998).

No entanto, a preocupação com a participação dos trabalhadores no projeto que impulsiona pesquisadores de Design Participativo foi contestada nos últimos anos pelas condições econômicas que predominam no cenário internacional, em que a eficiência é valorizada em detrimento da qualidade de vida no trabalho (KENSING; BLOMBERG, 1998). Os comunicadores técnicos têm muitas vezes tratado Design Participativo como um monólito, contanto que os usuários estejam de alguma forma envolvidos, é Design Participativo (SPINUZZI, 2002). Mas os pesquisadores de interação humano-computador têm argumentado que o Design Participativo envolve também a relação entre os *designers* e usuários e a maneira pela qual os dois interagem (SPINUZZI, 2002).

Procedimentos democráticos e participativos formais para projetar artefatos computacionais para a democracia no local de trabalho não são suficientes, o processo de *design* também deve ser organizado de forma a torná-lo possível para os usuários comuns de utilizar sua habilidade prática no trabalho de *design*, e se divertir fazendo isso (EHN, 1988).

Os métodos de Design Participativo foram inicialmente desenvolvidos para promover a capacitação democrática, eles buscaram fornecer uma linguagem comum para os *designers* e trabalhadores, e permitir que os trabalhadores se tornassem co-projetistas do sistema, dando-lhes uma avenida para controlar a introdução de novas tecnologias de maneira que fosse benéfico para eles (SPINUZZI, 2002). O Design Participativo tem um forte caráter democrático, principalmente na abordagem escandinava, um dos principais componentes dessa abordagem consiste em ensinar os trabalhadores as possibilidades de integrar a tecnologia em seu trabalho, nestes termos, o projeto UTOPIA refletiu um desejo de poder democrático, em que os trabalhadores pudessem tomar decisões juntamente com a gestão (SPINUZZI, 2002).

Olhando as duas abordagens de Design Participativo existentes, encontra-se para abordagem americana o empoderamento funcional, e para abordagem escandinava um empoderamento democrático, simplificando, o empoderamento funcional mantém grupos de trabalhadores responsáveis pelos resultados de determinadas tarefas, e em troca dá-lhes um grau de poder sobre como executar as tarefas, já o empoderamento democrático dá aos trabalhadores um papel de tomada de decisões no planejamento operacional, bem como na mudança organizacional e tecnológica (SPINUZZI, 2002).

Apesar dos resultados destes projetos iniciais, os trabalhadores continuam a ter dificuldade para argumentar a favor de formas alternativas de uso da tecnologia, em parte, porque os objetivos e estratégias de gestão, muitas vezes, foram construídos para os novos sistemas e

foram reforçados por distribuições de organização de poder, tornando difícil alterar a tecnologia para atender às necessidades e interesses dos trabalhadores (KENSING; BLOMBERG, 1998). Apesar disso o Design Participativo continua a existir como um compromisso com a participação dos trabalhadores em projetos e um esforço para equilibrar as relações de poder entre usuários e especialistas técnicos, e entre trabalhadores e gestores (KENSING; BLOMBERG, 1998).

Dentro deste universo Muller, Wildman e White (1994), descrevem cinco técnicas de Design Participativo por meio de jogos. Segundo eles o uso de jogos facilita o trabalho em grupo, pois fornecem um ambiente familiar e descontraído, em que as diversas partes envolvidas podem combinar seus conhecimentos afim de desenvolver soluções e atender as necessidades uns dos outros. As cinco técnicas são descritas nas Seções 2.3.1 *Interface Metaphors Game*, 2.3.2 *CARD Game*, 2.3.3 *Icon Design Game*, 2.3.4 *Pictive* e 2.3.5 *Interface Theater*.

#### 2.3.1. *Interface Metaphors Game*

Nesta técnica os participantes são divididos em pequenos grupos, e a cada grupo é atribuída a tarefa de desenvolver rapidamente um projeto conceitual de alto nível<sup>4</sup>. Este jogo ajuda a desenvolver um modelo metafórico sistemático para um domínio de sistema complexo (aquele que possui muitas variáveis e interações possíveis), com o objetivo de fornecer um modelo mental potencial para tornar mais fácil para os usuários finais entenderem o sistema. Uma equipe explora um ou mais domínios de metáfora e tenta corresponder os seus atributos para o problema de *design* atual, usando um jogo de cartas e um jogo de tabuleiro para estruturar seu trabalho (MULLER; HASLWANTER; DAYTON, 1997).

Os objetos correspondentes a esse jogo são as cartas e o tabuleiro. Cartas de jogo: cartões do molde formatado, cada cartão possui o domínio de metáfora (ou domínio de sistema), pelo menos um atributo do referido domínio, e algumas notas sobre como o atributo relaciona-se com outros atributos. Tabuleiro: são espaços usados para organizar e inter-relacionar as cartas de jogo (MULLER; HASLWANTER; DAYTON, 1997).

O processo da atividade pode ser descrito da seguinte maneira: (1) A equipe explora um ou mais domínios, anotando os atributos de cada domínio em cartões, um cartão por atributo. (2) A equipe explora o sistema ou domínio de trabalho, anotando seus atributos em cartões, um

---

<sup>4</sup> Segundo Muller, Haslwanter e Dayton (1997), entende-se um projeto de alto nível como sendo uma abstração do projeto maior, ou seja, uma representação superficial do projeto que desejasse entregar.

cartão por atributo. (3) A equipe explora potenciais partidas de atributos com outros atributos do sistema, alinhando o cartão no tabuleiro (MULLER; HASLWANTER; DAYTON, 1997).

Uma comparação dos desenhos resultantes irá ilustrar tanto (a) o impacto do modelo mental dos *designers* sobre o projeto, como (b) a importância de uma correspondência adequada entre as características dos dados, tarefa do usuário e da metáfora usada para representá-los (MULLER; WILDMAN; WHITE, 1994).

### 2.3.2. *CARD Game*

Os participantes são divididos em pequenos grupos, em que cada um desenvolverá um fluxo de trabalho para o sistema, para tanto é utilizada a técnica *CARD (Collaborative Analysis of Requirements Design)*, em que são usados cartões para representar as tarefas (MULLER; WILDMAN; WHITE, 1994). Os participantes usam um baralho de cartas para o *layout* ou fluxos de tarefas críticas. (MULLER; HASLWANTER; DAYTON, 1997). Cada cartão pode representar um evento dentro do sistema, um evento local de trabalho que ocorre fora do sistema, uma ação do usuário, ou ainda um estado cognitivo motivacional de um usuário. (MULLER; WILDMAN; WHITE, 1994).

Os cartões representando componentes de trabalho, muitas vezes são organizados em uma hierarquia de classes de componentes, para tanto usa-se uma folha de papel de fundo, sobre a qual os cartões podem ser ligados, e na qual o fluxo de setas, pontos, critérios de decisão, podem ser desenhados (MULLER; HASLWANTER; DAYTON, 1997).

O modelo de processo desta atividade pode ser descrito como: (1) Introduções que incluem experiência de cada participante, contribuição para o trabalho compartilhado, e organização (2) Trabalhar em conjunto para explorar o domínio de tarefas (na forma de análise, concepção, ou avaliação). (3) Breve explicação passo a passo das realizações do grupo durante a sessão (MULLER; HASLWANTER; DAYTON, 1997).

Como resultado tem-se as representações dos fluxos de tarefas, gravado em sequências de cartas, além de anotações e setas de fluxo de tarefas, pontos de ramificação, critérios de decisão, e assim por diante, registrados na folha de fundo (MULLER; HASLWANTER; DAYTON, 1997).

### 2.3.3. *Icon Design Game*

Divididos em grupos de até 20 pessoas, um dos participantes (o *sketcher*) desenha ícones informais, enquanto outros participantes tentam adivinhar o conceito que o *sketcher* está

tentando expressar. Os desenhos tornam-se os primeiros projetos para o desenvolvimento dos ícones. Ao término de cada rodada, as equipes reveem os esboços para identificar as características que proporcionaram as melhores pistas. O jogo pode ser jogado cooperativamente (com uma única equipe) ou de forma competitiva (com várias equipes). Os materiais utilizados nesta atividade são folhas de papel e canetas (MULLER; HASLWANTER; DAYTON, 1997).

O processo pode ser descrito mais especificamente da seguinte forma: (1) O *sketcher* seleciona um conceito para tentar se comunicar com a equipe. (2) O desenhista então desenha imagens do conceito. (3) A equipe tenta adivinhar o conceito que o *sketcher* está tentando expressar. (4) Opcionalmente, um observador toma notas em desenhos dos pontos que parecem ser particularmente eficazes ou confusos. (5) Os melhores desenhos são selecionados para o desenvolvimento (MULLER; HASLWANTER; DAYTON, 1997). O resultado final são esboços de ícones, para o desenvolvimento (MULLER; WILDMAN; WHITE, 1994).

#### 2.3.4. PICTIVE

A técnica *PICTIVE* (*Plastic Interface for Collaborative Technology Initiatives through Video Exploration*) é aplicada em pequenos grupos de até oito pessoas, a dois subproblemas dentro do domínio do projeto (um gráfico, e outro textual). Os participantes projetam aspectos relevantes da *interface* usando papel e lápis para um tipo de "prototipagem rápida" sem usar um ambiente de prototipação (MULLER; WILDMAN; WHITE, 1994).

Para realização da atividade são utilizados materiais de escritório comuns (canetas coloridas, tesouras, *post-it*), e materiais pré-impressos personalizados (por exemplo, componentes de *interface* que estejam em conformidade com um guia, ou o desenvolvimento de um estilo ambiente em particular, ou ainda ícones do domínio de trabalho) (MULLER; HASLWANTER; DAYTON, 1997).

O modelo do processo pode ser descrito da seguinte forma: (1) Introduções que incluem experiência de cada participante e contribuição para o trabalho compartilhado. (2) Trabalhar em conjunto para explorar o domínio de tarefas (na forma de análise, concepção, ou avaliação). (3) Breve explicação passo a passo das realizações do grupo durante a sessão (MULLER; HASLWANTER; DAYTON, 1997).

Desta atividade obtêm-se 2 resultados: artefatos de papel que representam a aparência do sistema, e artefatos de papel que podem ser utilizadas para reconstruir as ideias do grupo sobre

a dinâmica do sistema (MULLER; HASLWANTER; DAYTON, 1997). O foco deste exercício é sobre a participação relativamente igual de todos os membros da equipe na atividade de *design*, e as contribuições separadas e únicas que a perícia de cada participante pode fazer para o esforço de *design* colaborativo (MULLER; WILDMAN; WHITE, 1994).

### 2.3.5. *Interface Theater*

Nesta técnica todos os participantes interagem em conjunto, o projeto será promulgado sob a forma de uma peça, usando pessoas para levar os componentes de *script* do sistema (por exemplo, Maria a opção do Menu, João o *DialogBox*, Ana o Teclado) (MULLER; WILDMAN; WHITE, 1994). Trabalhando com adereços teatrais em escala humana, a equipe de *design* age com a aparência e dinâmica da *interface* do usuário e seu sistema. Os atores são guiados por "*scripts* orientados a objetos" que descreve a funcionalidade de componentes de *interface*. O público dos utilizadores finais e outros interessados critica a aparência e as ações, transformando o *design*. A trupe de atores reencena a *interface* até que eles e o público estejam satisfeitos (MULLER; HASLWANTER; DAYTON, 1997).

O processo de desenvolvimento desta atividade pode ser descrito da seguinte forma: (0) Antes do teatro, a equipe de *design* desenvolve os adereços teatrais, tais como caixas de diálogo, imagens de cursor, e outros componentes de *interface*. A equipe de projeto também desenvolve roteiros orientados a objeto. Eles podem, assim, escrever um cenário para começar o drama. (1) Guiados por três personagens de processo (o agente público, o crítico, e o Espírito; consulte os passos 2-5), os caracteres (interpretado por membros da equipe de projeto) se apresentam para o público dos utilizadores finais e outras partes interessadas. (2) O agente público trabalha com o público para controlar o caráter cursor e o personagem teclado como entrada para o sistema. Seguindo seus roteiros, os personagens de cursor e teclado enviam mensagens para outros papéis. (3) Os outros papéis respondem com métodos e suas próprias mensagens. (4) O crítico trabalha com o público para criticar o projeto. (5) O Espírito tenta destacar questões de projeto e de trabalho, para acompanhar as necessidades de todos. (6) A peça é reapresentada, com alterações, até que todos estejam satisfeitos (MULLER; HASLWANTER; DAYTON, 1997).

## 2.4. TRABALHOS CORRELATOS

Como último fator de contextualização desta pesquisa são apresentados nesta seção os trabalhos correlatos encontrados na revisão de literatura.

Entre as temáticas encontradas, tem-se Larraza-Mendiluze et al. (2016) que trabalhou com ensino de programação por meio de jogos. Em sua pesquisa Larraza-Mendiluze et al. (2016) descreve um ambiente de aprendizagem baseado em projetos em que o desenho de um jogo é a base do projeto, a fim de incentivar os alunos a se envolverem mais com os conceitos de transmissão de informações em um computador. Os meios eletrônicos, como os computadores lidam com dados, que são conceitos abstratos e difíceis para os calouros de graduação entenderem, no entanto, é importante que os calouros assimilem estes conceitos se eles vão ser ensinados sobre os sistemas operacionais que trabalham sobre essa camada de arquitetura (LARRAZA-MENDILUZE et al., 2016). Como resultados para o ambiente desenvolvido os autores notaram uma melhora por parte dos estudantes na compreensão de conceitos antes abstratos (LARRAZA-MENDILUZE et al., 2016).

Já o autor Wang (2011) descreve uma avaliação de que se um projeto de desenvolvimento de jogos poderia ser usado para ensinar arquitetura de *software*. Neste projeto, estudantes universitários tiveram de construir e projetar um tipo de arquitetura de *software*, avaliar e implementar uma aplicação baseada na arquitetura, e testar essa implementação (WANG, 2011). A principal conclusão é que os projetos de desenvolvimento de jogos com êxito podem ser usados para ensinar arquitetura de *software* (WANG, 2011).

Os autores Werner, Derner e Campe (2014) relatam os resultados de um estudo da relação de programação de jogos de computador para a aprendizagem computacional no qual foram analisados 231 jogos programados por crianças de 11 e 12 anos que participaram de oficinas de programação durante ou depois da escola. Os resultados sugerem que os jogos dos estudantes exibiram: programas constituídos por sequências de construções de programação simples e o uso de padrões de desenvolvimento que não os foram ensinados (WERNER; DENNER; CAMPE, 2014). Os estudos de caso de jogos dos alunos foram usados para ilustrar como a variação no uso e integração de desenvolvimento de programação, padrões e mecânica de jogo podem ser usados para demonstrar evidências de aprendizado computacional (WERNER; DENNER; CAMPE, 2014). O estudo contribui para a compreensão de como atividades de programação de jogos podem promover o desenvolvimento do raciocínio computacional (WERNER; DENNER; CAMPE, 2014).

Repenning (2015) desenvolveu um modelo de currículo baseado em uma estratégia para integrar e ensino de ciências da computação no currículo escolar regular de estudantes do ensino básico e fundamental. A proposta do autor incluiu as oportunidades para estudantes de projetar e programar jogos, o ensino da engenharia por trás dos computadores e simulações matemáticas

(REPENNING et al., 2015). Uma abordagem foi desenvolvida para medir e correlacionar as habilidades de pensamento computacionais relevantes para *design* de jogo e simulações (REPENNING et al., 2015). Os resultados de um estudo, com mais de 10.000 alunos, demonstraram que a rápida adoção deste modelo por docentes de várias disciplinas, gerou um aumento da motivação dos estudantes, altos níveis de participação das mulheres, e o aumento do interesse dos estudantes (REPENNING et al., 2015).

Duas outras pesquisas (GESTWICKI; MCNELLY, 2016; HAO; LEUNG, 2013) trabalharam o desenvolvimento de modelos de aplicações computacionais com fins de facilitar o processo de aprendizagem. Gestwicki e McNely (2016) desenvolveram um modelo de espaço acadêmico para graduação. Este modelo traz docentes, alunos e parceiros da comunidade para investigar em conjunto uma questão acadêmica, e sua colaboração produz um resultado original que representa o seu inquérito (GESTWICKI; MCNELLY, 2016). Segundo o estudo, esse espaço acadêmico integra a prática ágil de desenvolvimento de *software*, a pedagogia orientada para o projeto, e as teorias de cognição socioculturais (GESTWICKI; MCNELLY, 2016).

Hao e Leung (2013) propuseram um quadro de aprendizagem social alternativa com duas contribuições importantes da seguinte forma: em primeiro lugar, introduzir um mecanismo de observação para reduzir a quantidade de comunicação necessária entre os agentes, e em segundo lugar, propõe-se que a estratégia de aprendizagem dos agentes deve ser baseada na técnica de reforço de aprendizagem, em vez de aprendizagem evolutiva. Desta forma, cada estudante mantém explicitamente o registro de seu estado atual em sua estratégia de aprendizagem, e aprende a sua política ótima para cada estado independente, com isso, o desempenho da aprendizagem, segundo os autores, é muito mais estável e adequado ao aprendizado (HAO; LEUNG, 2013).

O autor Boyd et al. (2015), trabalhou com tecnologias assistivas e jogos colaborativos buscando o desenvolvimento de habilidades sociais para crianças com transtorno do espectro do autismo. Por meio de um estudo empírico do uso de um jogo colaborativo para *iPad*, foi explorado como as tecnologias podem ser utilizadas para apoiar as relações sociais, mesmo sem a intervenção dos adultos (BOYD, 2015). Os resultados mostram que *design* específicos podem incentivar três níveis de relação social: inclusão, parceria e amizade (BOYD, 2015).

A modelagem virtual de ambientes físicos com fim de aprendizado foi um ponto abordado por alguns autores (ANTONIOU et al., 2013; COENEN; MOSTMANS; NAESSENS, 2013; FROSCHAUER et al., 2013; RUBINO et al., 2015; BELLOTTI et al., 2012). O enfoque destas



pesquisas estava no desenvolvimento de ambientes virtuais que possibilitassem o ensino de pessoas por meio da navegação nas aplicações desenvolvidas, os trabalhos têm como objetivo o ensino de história, e as aplicações foram projetadas para representar museus, patrimônios históricos, galerias de arte, e outras.

Froschauer et al. (2013), Antoniou et al. (2013) e Bellotti et al. (2012) desenvolveram ferramentas destinadas a modelar ambientes reais em ambientes virtuais 3D, em que, por meio de jogos e atividades os usuários pudessem andar por uma exposição e conhecer a história dos objetos expostos de maneira interativa. Os autores buscaram pelo uso dessa abordagem lúdica o aumento da motivação para aprender, e também expandir o interesse em história da arte e no patrimônio cultural em geral (FROSCHAUER et al., 2013; ANTONIOU et al., 2013; BELLOTTI et al., 2012). As três pesquisas encontraram resultados semelhantes quanto ao nível de estímulo que os artefatos apresentaram, os participantes da pesquisa se sentiram estimulados a continuar o *tour* pelo jogo, e gostaram da liberdade para interagir com o que desejavam (FROSCHAUER et al., 2013; ANTONIOU et al., 2013; BELLOTTI et al., 2012).

Os autores Coenen, Mostmans e Naessens (2013) e Rubino et al. (2015) buscaram implementar aplicações para serem usadas durante visitas a ambientes físicos, o objetivo era fornecer um meio de aprendizagem durante uma visita a uma exposição em um museu, por exemplo. O ponto principal levantado é que para experiência, por exemplo em museus, não é necessário um caminho pré-definido, com isso a aplicação estimula o visitante a olhar para os elementos do patrimônio cultural de uma maneira diferente, permitindo a construção de narrativas pessoais, e a criação de uma exposição pessoal (COENEN; MOSTMANS; NAESSENS, 2013; RUBINO et al., 2015). Os autores notaram que os usuários da aplicação despendiam mais tempo observando e aprendendo sobre determinados marcos, do que outros visitantes que não estavam utilizando esse recurso (COENEN; MOSTMANS; NAESSENS, 2013; RUBINO et al., 2015).

Por fim, ainda tratando do escopo de ensino de história, usando programas digitais, os autores Huang e Huang (2013) desenvolveram um jogo relacionado com o cultura e história de vida dos povos indígenas de Taiwan. O desenvolvimento do jogo compreendeu as camadas de dados, lógica e apresentação. Durante a camada de dados, foi utilizada taxonomia para extrair os componentes culturais da tribo *Atayal* de Taiwan; durante a camada de lógica, foi empregada uma análise das características culturais e comparações de metas de educação históricas para confirmar estrutura da narrativa do jogo; e na camada de apresentação, foram convertidos componentes culturais em elementos de *interface* do jogo (HUANG; HUANG, 2013).

Os resultados da avaliação mostraram que jogo desenvolvido obteve benefícios positivos no aumento da motivação de aprendizagem das crianças em idade escolar e no aumento do conhecimento sobre a vida indígena da região (HUANG; HUANG, 2013). A pesquisa verificou que a integração dos métodos interdisciplinares e antropólogos com professores de história, anciãos tribais e *designers* de jogos permitiu uma modelagem fiel e completa do ambiente proposto, o que não teria sido possível sem um grande nível de dados para serem usados na modelagem do artefato (HUANG; HUANG, 2013).

### 3. METODOLOGIA

No presente capítulo são abordados os assuntos referentes a metodologia do trabalho proposto. Na Seção 3.1 é apresentada a caracterização da pesquisa; na Seção 3.2 a descrita a obtenção do referencial teórico; na Seção 3.3 é definido o público-alvo da pesquisa; na Seção 3.4 são definidas as técnicas e métodos de Design Participativo que serão utilizadas; por fim na Seção 3.5 são descritos os passos estabelecidos.

#### 3.1. CARACTERIZAÇÃO DA PESQUISA

Do ponto de vista de sua natureza, esta pesquisa pode ser classificada como pesquisa aplicada, pois objetiva gerar conhecimentos para aplicações práticas dirigidas à solução de problemas específicos (LAKATOS; MARCONI, 2005).

Quanto a forma de abordagem utilizada, esta pesquisa é classificada como qualitativa, pois considera que existe uma relação entre o mundo e sujeito que não pode ser traduzida em números (LAKATOS; MARCONI, 2005).

Em se tratado dos objetivos, pode-se classificar este trabalho como pesquisa ação, pois é concebida em associação com uma ação, os pesquisadores e participantes da situação ou problema estão envolvidos de modo cooperativa ou participativo (LAKATOS; MARCONI, 2005).

#### 3.2. REFERENCIAL TEÓRICO

Nessa seção são descritos os processos de obtenção da pesquisa bibliográfica e da revisão de literatura do presente trabalho.

##### 3.2.1. Pesquisa Bibliográfica

A pesquisa bibliográfica deste trabalho foi dividida em três temas de pesquisa: Jogos Digitais na Educação, Ensino de Computação e Design Participativo. A pesquisa por referências que sustentam esses temas foi feita no repositório de periódicos da Capes e com o auxílio do *Google Scholar*, que atua como um buscador de artigos em diversas bases de periódicos.

Inicialmente as palavras-chave foram procuradas nos meios de busca, para fins de escolha foi dada prioridade a artigos científicos, que foram selecionados em ordem de apresentação, para cada artigo foi lido o resumo e escolhidos os que se encaixavam dentro do tema da pesquisa.

As palavras-chave usadas para jogos digitais no ensino foram: “Jogos digitais no ensino”, “Jogos computacionais no ensino”, “Aprendizado com jogos digitais”, “*Teaching with computer games*”, “Gamificação”, “*Gamefication*”. Para ensino de computação foram: “Ensino de computação”, “Aprendizado de Computação”, “Introduzir programação no ensino”, “*Teaching computing*”, “*Computing learning*”, “*Computing in school*”. Por fim as palavras chaves usadas para pesquisa no campo de Design Participativo foram: “Desing Participativo” e “*Participatory Design*”.

### 3.2.2. Revisão de Literatura

A pesquisa por trabalhos correlatos a esta pesquisa foi realizada no repositório de periódicos ACM DL (*Association for Computing Machinery Digital Library*), utilizando a palavra-chave “*“Collaborative model” game “children programming”*”. Optou-se por este repositório já que ele concentra artigos da área de computação apresentados em eventos e também publicados em periódicos. Alguns filtros foram definidos, como: pesquisa por artigos acadêmicos e o ano de publicação dos artigos variando de 2011 a 2016.

Foram encontrados 300 artigos correspondentes. Para cada artigo apresentado na busca foi lido o resumo para ver se havia correspondência com o tema proposto. Após a leitura dos resumos, foram selecionados 33 artigos que possuíam alguma relação com o tema do presente trabalho. Os artigos foram então lidos, e mais 20 artigos foram eliminados, restando 13 artigos que apresentam relação com o tema. Os resultados encontrados foram descritos na Seção 2.4 de Trabalhos Correlatos.

### 3.3. DEFINIÇÃO DO PÚBLICO-ALVO

O público-alvo da presente pesquisa consiste em estudantes do ensino básico e fundamental, com faixa etária de 5 a 9 anos.

O público foi proveniente de escolas parceiras do grupo PET-CoCE (Programa de Educação Tutorial - Computando Culturas em Equidade) da Universidade Tecnológica Federal do Paraná (UTFPR), devido ao autor e a orientadora do presente trabalho fazerem parte do grupo.

O grupo PET-CoCE desenvolve diversas atividades de extensão e inclusão com a comunidade interna à universidade e com a comunidade externa, incluindo crianças e adolescentes de diversas escolas e institutos de Curitiba.

### 3.4. DEFINIÇÃO DAS TÉCNICAS E MÉTODOS DE DESIGN PARTICIPATIVO

No Capítulo 2, Seção 2.3, são descritas cinco técnicas de Design Participativo que fazem uso de jogos, que segundo Muller, Wildman e White (1994) proporcionam um ambiente familiar e relaxante para aplicação de atividades. Por se tratar de uma pesquisa realizada com um público-alvo muito jovem, com faixa etária de 5 a 9 anos, estas técnicas são boas alternativas para serem usadas nas atividades referentes a esta pesquisa.

Cada técnica gera diferentes resultados, e a escolha das técnicas foi determinada por esse fator. Para serem aplicadas em um primeiro momento foram escolhidas as técnicas que resultam em um esboço preliminar, no caso as técnicas *PICTIVE*, *Interface Metaphors Game* e *CARD Game*. Após, caso houvesse necessidade, seriam usadas as técnicas *Icon Design Game* e *Interface Theater*, que possibilitariam um refinamento do modelo.

### 3.5. PASSOS ESTABELECIDOS

A Figura 3 apresenta o fluxo de atividades estabelecidos para o desenvolvimento do projeto.



*Figura 3 - Fluxo de atividades (Fonte: Autoria Própria)*

Inicialmente, as atividades a serem realizadas foram definidas. Então, as técnicas escolhidas foram aplicadas e os resultados obtidos, os quais foram analisados e usados para projetar o modelo do jogo.

Vale ressaltar que as atividades estabelecidas foram planejadas para serem realizadas em dois encontros, porém apenas o primeiro encontro foi realizado, visto que não verificada a necessidade do segundo encontro. O primeiro encontro durou 1,5 hora. O detalhamento deste encontro é apresentado no Capítulo 4.

Com o projeto colaborativo finalizado foi possível realizar o desenvolvimento do artefato, que posteriormente foi validado. Com todos os resultados adquiridos foi então redigida a conclusão do trabalho.

## 4. AS ATIVIDADES

Neste capítulo são descritos todos os aspectos referentes as atividades propostas, com a seguinte divisão: na Seção 4.1 é descrito o planejamento das atividades, na Seção 4.2 é apresentada a aplicação das atividades, e por fim a Seção 4.3 apresenta uma análise dos resultados obtidos.

### 4.1. PLANEJAMENTO

No capítulo 2, Seção 2.3, do presente trabalho foram descritas cinco técnicas de Design Participativo que poderiam ser usadas para o desenvolvimento do artefato proposto nesta pesquisa, visto isso, o planejamento das atividades foi realizado com base no resultado que cada técnica oferece, de maneira a criar uma ordem de desenvolvimento para o modelo.

Para este trabalho uma atividade caracteriza a aplicação de uma das técnicas de Design Participativo. Uma atividade envolve, acolhimento dos participantes, apresentação do objetivo da técnica de Design Participativo, desenvolvimento da técnica propriamente dita, socialização e discussão com o grupo sobre os resultados obtidos.

A realização das atividades foi dividida em duas partes, sendo que na primeira foi planejada a aplicação de duas técnicas (*PICTIVE* e *Interface Metaphors Game + CARD Game*), e caso houvesse a necessidade de um maior detalhamento nos resultados da primeira etapa, seria realizada uma segunda parte com outras duas técnicas (*Icon Design Game* e *Interface Theater*) que permitiriam uma complementação do projeto colaborativo do artefato. Esta atividade ocorreu no dia 07 de outubro, com duração de 1,5 horas.

A técnica escolhida para primeiro ser aplicada foi a *PICTIVE* (*Plastic Interface for Collaborative Technology Initiatives through Video Exploration*), descrita na Seção 2.3.4. Essa técnica foi escolhida porque resulta em projeto preliminar de alto nível em papel, que permite uma visão geral de um sistema de forma simples e superficial, porém com todos os componentes já inclusos (MULLER; HASLWANTER; DAYTON, 1997). O tempo de aplicação da *PICTIVE* foi de 30 minutos.

Após a aplicação da técnica *PICTIVE*, vislumbrou-se a possibilidade de unir duas técnicas que possibilitam o refinamento do projeto gerado na etapa anterior, sendo estas a técnica *Interface Metaphors Game* (Seção 2.3.1) e a *CARD Game* (Seção 2.3.2). Como ambas trabalham com cartões, isso possibilitou utilizar um baralho que unisse as duas técnicas,

permitindo um maior domínio de atuação, e reduzindo o tempo de aplicação das técnicas. Elas foram desenvolvidas no mesmo encontro do dia 07 de outubro e a aplicação durou 40 minutos.

Porém antes de aplicar as técnicas foi necessário explicar para os participantes qual o objetivo conceitual que se buscava atingir, ou seja, apresentar programação para crianças. Como este não é um conteúdo usualmente de domínio infantil, optou-se por realizar uma apresentação com artefato lúdico que desse condições de tratar temas como: comando, instrução, “finitude”, erro, lateralidade. Assim, o projeto chamado Roboqueda (descrito na Seção 4.1.1) desenvolvido na UTFPR surgiu como uma boa forma de demonstrar de maneira prática o que pretendia-se ter em um meio digital.

#### 4.1.1. Roboqueda

O Roboqueda (Figura 4) é um projeto desenvolvido pelo grupo PET-CoCE da UTFPR, ele consiste em um brinquedo que tem por objetivo estimular o ensino de programação e o desenvolvimento do raciocínio lógico matemático. A ideia do projeto é trabalhar habilidades de raciocínio lógico para possibilitar melhorias em outras áreas educacionais (matemática, língua portuguesa, ciências), por meio de conceitos de instrução, comando, lateralidade, “finitude”, trabalho em grupo (GERON; SDROIEVSKI; AMARAL, 2014).



*Figura 4 - Roboqueda (Fonte: Grupo PET-CoCE)*



O Roboquedo é constituído de uma série de artefatos, sendo estes: uma mesa com setas, um mapa e um robô em forma de tartaruga. Na mesa são inseridas peças em forma de setas, que comandam o robô, o instruindo a ir na direção especificada. O mapa consiste um caminho que o robô deve percorrer (GERON; SDROIEVSKI; AMARAL, 2014). O artefato foi utilizado antes das técnicas com a finalidade de tornar lúdica a exemplificação do objetivo do jogo colaborativo que seria modelado durante as técnicas.

#### 4.2. O QUE ACONTECEU

As atividades que envolviam as técnicas participativas ocorreram no dia 07 de outubro, e contaram com a participação de seis crianças voluntárias com idades entre 7 e 11 anos, sendo cinco meninos e uma menina. As crianças foram contatadas por meio do grupo PET-CoCE, e são discentes de oficinas ofertadas pelo grupo. Inicialmente as técnicas seriam realizadas com oito participantes, porém no dia da atividade duas meninas não puderam comparecer.

Como planejado, inicialmente os participantes tiveram contato com o Roboquedo (Figura 5), descrito na Seção 4.1.1. Um instrutor explicou o funcionamento da ferramenta e qual sua finalidade e objetivo a ser atingido, e em grupo os participantes puderam interagir com o Roboquedo durante o período de 10 minutos. Durante a interação não houveram problemas, o objetivo foi entendido por todos, e o grupo conseguiu concluir a atividade de maneira rápida e tranquila.



*Figura 5 - Interação com o Roboquedo (Fonte: Autoria Própria)*

Após o término da interação com o Roboquedo, os estudantes foram conduzidos para outra sala, e foram instruídos de maneira independente a se dividirem em dois grupos, cada um com 3 alunos. Para fins de análise os grupos foram denominados de Grupo A (três meninos com idade de 11 anos cada) e Grupo B (dois meninos com idades de 11 e 7 anos e uma menina com 11 anos). Após os grupos estarem divididos os participantes foram instruídos a realização da primeira técnica participativa, que é descrita na Seção 4.2.1.

#### 4.2.1. *PICTIVE*

Os dois grupos foram separados em duas mesas, cada qual com folhas sulfite no tamanho A3, diversos materiais de escritório (canetas, lápis, borracha, tesoura...), e alguns moldes de componentes de sistema digitais impressos em uma impressora 3D.

Um instrutor explicou que cada grupo teria 15 minutos para discutir entre os integrantes e desenhar como imaginavam que seria um jogo para ensinar sequência de passos, para crianças com idade inferior à deles. Os grupos poderiam fazer da maneira como desejassem, não precisando se basear no Roboquedo.

Em cada mesa um instrutor ficou tomando notas sobre os debates realizados no grupo, os pontos de conflito, as principais dificuldades e os pontos principais do modelo. Após o término do tempo, cada grupo apresentou o desenho que desenvolveram para os demais participantes e instrutores. A Figura 6 apresenta um conjunto de imagens da aplicação da técnica.

Os pontos mais relevantes observados na aplicação dessa técnica foram:

- Ambos os grupos decidiram por usar a tartaruga como personagem do jogo;
- Ambos os grupos usaram setas direcionais;
- Foi possível constatar a influência que elementos de outros jogos clássicos tiveram em ambos os grupos na hora de escolher quais elementos usar;
- O Grupo A inseriu elementos como vida, customização, e um caminho variável;
- O Grupo B discutiu a importância das cores no modelo;
- O Grupo B inseriu um mapa e um caminho.



*Figura 6 - Desenvolvimento da Técnica PICTIVE (Fonte: Autoria Própria)*

Também foram levantados alguns comentários sobre cada grupo. O grupo A trabalhou de forma unida e sem conflitos, os participantes citaram diversas vezes jogos conhecidos por eles como base para o que estava sendo desenvolvido, mas houveram momentos que inseriram funcionalidades que não sabiam explicar o motivo. O grupo B não trabalhou de maneira unida e ocorreram muitos conflitos, o grupo demorou para conseguir chegar a um consenso, houveram muitas ideias e sugestões, mas por fim o grupo conseguiu realizar a atividade.

Após o término da aplicação desta técnica os grupos foram instruídos a se reunirem novamente com os desenhos para a próxima técnica descrita no tópico 4.2.2.

#### 4.2.2. *Inteface Metaphors Game + CARD Game*

Após a apresentação dos desenhos feitos na técnica anterior deu-se início a atividade *Interface Metaphors Game + CARD Game*. Cada grupo contou com um instrutor, e com a ajuda deste, cada equipe explorou domínios do modelo proposto. Cada instrutor anotou os dados referentes a interação do grupo que estava acompanhando.

Os grupos também realizaram modificações em seus desenhos, resultante da exploração dos domínios durante a técnica. Os desenhos resultantes e os pontos anotadas são descritos na Seção 4.2.3.

#### 4.2.3. Resultados das Aplicações das Técnicas

Os resultados obtidos com as técnicas de forma geral são descritos neste tópico, para fins de análise os resultados de cada grupo são descritos separados. Como dito anteriormente os grupos foram divididos em Grupo A e Grupo B para facilitar na descrição.

Para o protótipo do grupo A, apresentado na Figura 7, tem-se os seguintes tópicos:

- O agente do jogo seria uma tartaruga, que teria que seguir por um caminho desviando de obstáculos e juntando moedas;
- A tartaruga pode acelerar e desacelerar sua velocidade, ir para o lado direito e esquerdo, e subir e descer dentro da água;
- A tartaruga possui uma barra de vida e outra de oxigênio, que indicam a vida restante e quantidade de tempo que ela ainda pode permanecer embaixo da água;
- Existe um limite de tempo para terminar o percurso;
- Existe uma loja em que o usuário pode personalizar a tartaruga;
- O jogo deveria ter cores fortes e chamativas;
- Se a tartaruga “morrer” deve recomeçar o percurso do início.

Durante a interação foi possível levantar alguns pontos interessante sobre o grupo A: ocorreram momentos em que os praticantes perderam o objetivo e não sabiam explicar a razão de alguns elementos, como por exemplo a presença do jacaré. Outro ponto, foram elementos como vida, tempo, moedas e loja, os quais os participantes do grupo afirmaram terem retirado de outros jogos.

Quanto aos processos dentro do jogo os participantes do grupo A descreveram que o jogador deveria percorrer um caminho coletando as moedas, para tanto deveria usar as setas. Com as moedas o jogador poderia personalizar a tartaruga, como mudar a cor ou forma do casco.



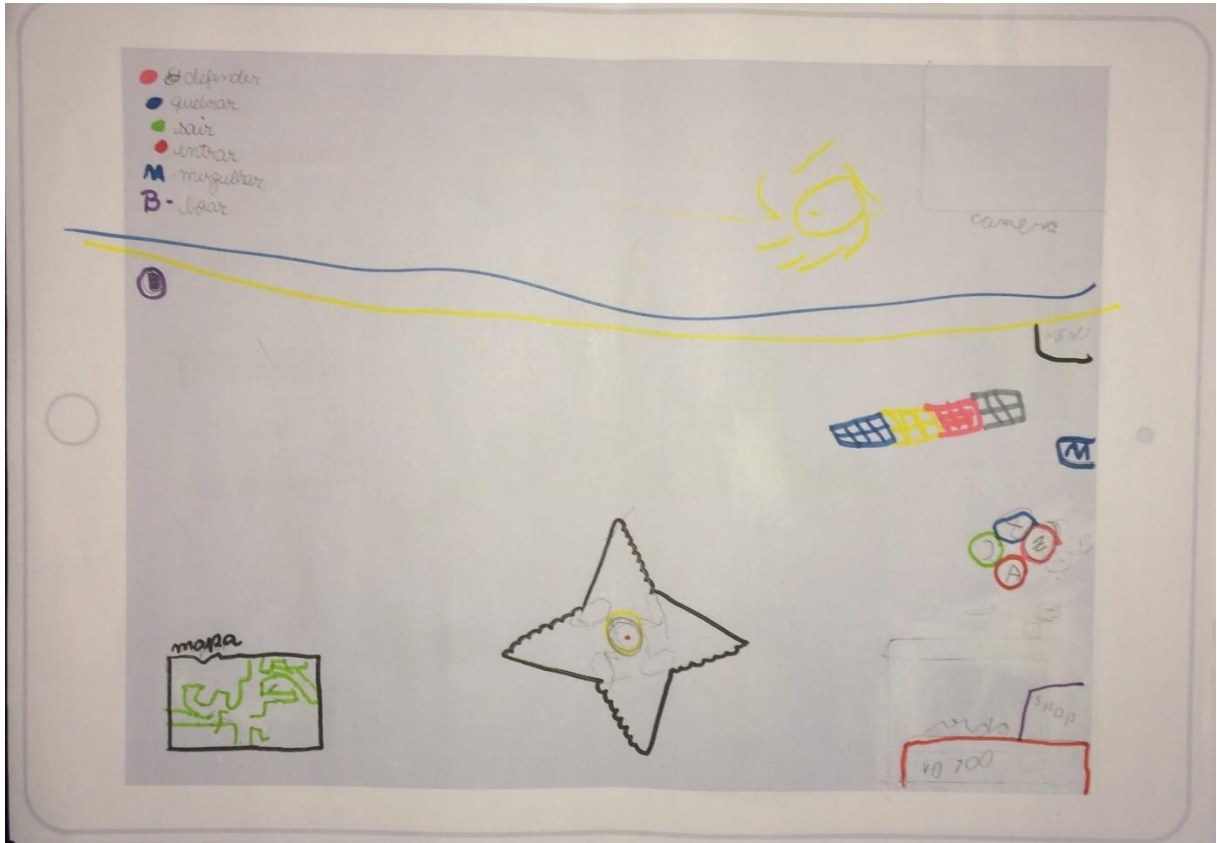
*Figura 7 - Desenho Grupo A (Fonte: Autoria Própria)*

Para o protótipo do Grupo B, apresentado na Figura 8, tem-se os seguintes tópicos:

- O agente do jogo seria uma tartaruga, que teria que passar por diversas fases, onde cada fase é um caminho diferente;
- A tartaruga pode ir para cima, baixo, esquerda e direita;
- O usuário digita os comandos nas setas, e pressiona um botão no centro das setas para executar a sequência de comandos;
- Existe um limite de tempo para terminar o percurso;
- Caso a tartaruga “morra” o personagem deve recomençar o jogo da primeira fase, independente da fase que tenha morrido.

O grupo B demorou para conseguir entrar em um consenso, foram diversas ideias, que não estavam conectas. O grupo frisou diversas vezes a importância das cores do jogo. Para eles também era importante a existência de uma grande variedade de fases, que ficariam mais difíceis a medida que o jogador avançasse no jogo.

Após discutirem bastante, o grupo optou por desenvolver um jogo composto de várias fases, cada uma com um caminho diferente dentro de um labirinto. O jogador deveria se movimentar usando as setas, e teria um tempo para resolver o caminho.



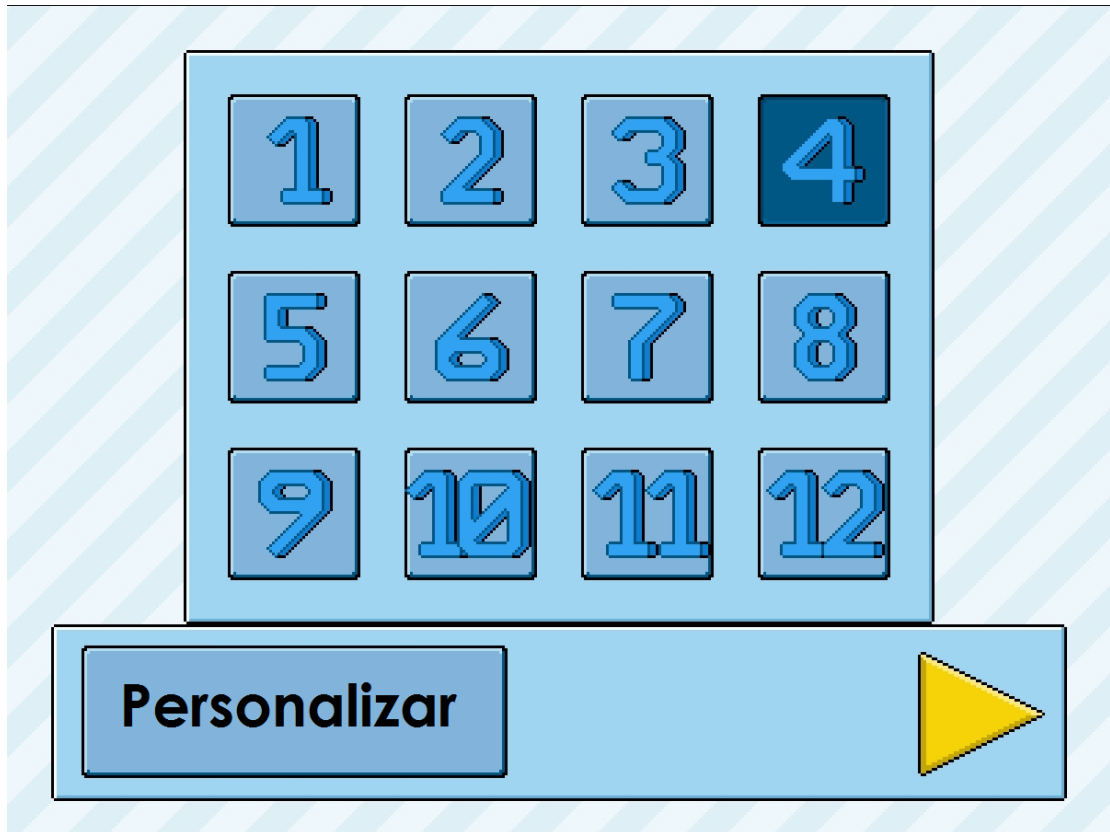
*Figura 8 - Desenho Grupo B (Fonte: Autoria Própria)*

Com a análise dos resultados obtidos foi possível desenvolver um *mockup* preliminar do jogo proposto, descrito no tópico 4.3

#### 4.3. ANÁLISE

A partir dos desenhos obtidos nas técnicas participativas foi possível criar um *mockup* de jogo com os principais componentes levantados. A Figura 9 e a Figura 10 representam os desenhos resultantes obtidos. Este tipo de análise é realizada para tornar legível os objetivos expostos nos desenhos realizados nas técnicas. Os *mockups* também foram utilizados como um resumo dos resultados obtidos pelos dois grupos.

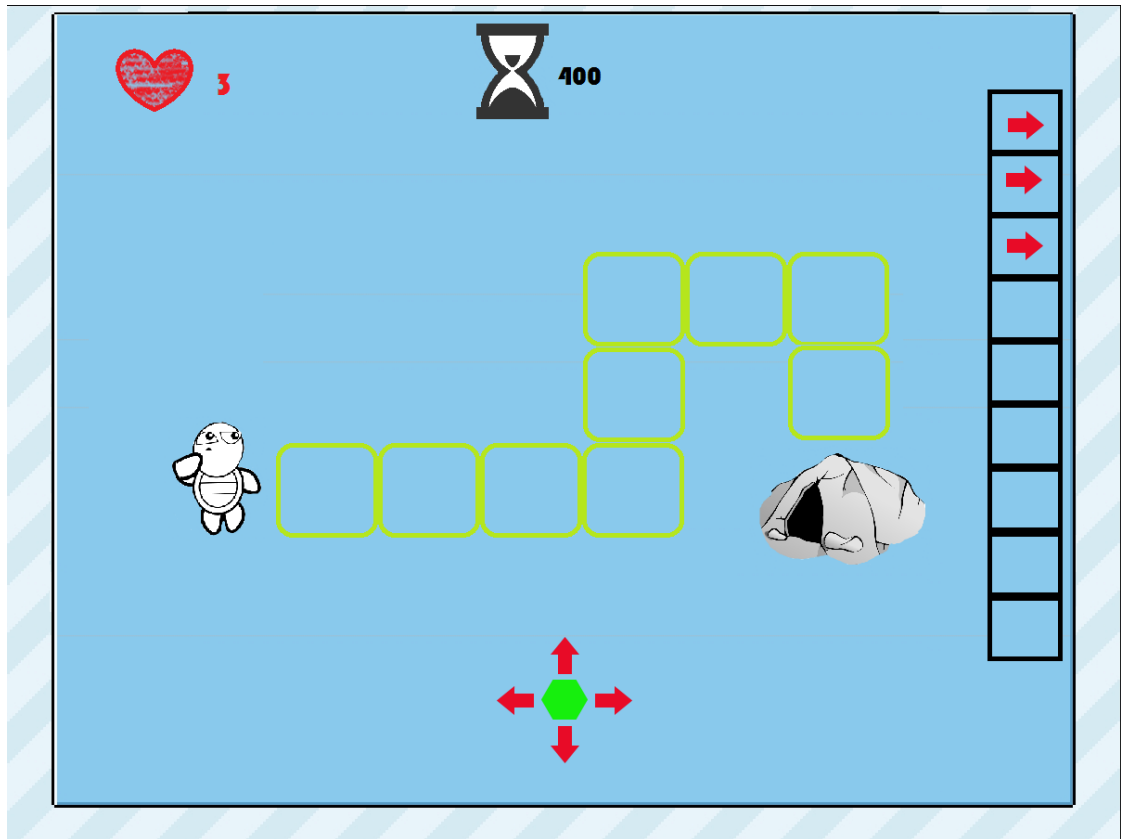
Na Figura 9 está representado o menu de fases, baseado nas sugestões do Grupo B. Outro elemento nessa tela é o botão de personalizar sugerido pelo Grupo B, em que o jogador pode alterar a cor e outros elementos da tartaruga.



*Figura 9 - Mockup preliminar, Fases (Fonte: Autoria Própria)*

Na Figura 10 está apresentado o jogo. Vale destacar que os participantes assistiram e contribuíram com o desenvolvimento do *mockup*. É possível observar alguns pontos principais:

- Caminho a ser percorrido;
- Setas de direção;
- Botão para execução (sugestão do Grupo B);
- Temporizador (sugestão do Grupo A);
- Número de vidas (sugestão do Grupo A);
- Barra para comandos.



*Figura 10 - Mockup preliminar, Jogo (Fonte: Autoria Própria)*

Realizando uma análise dos resultados obtidos com as técnicas e o objetivo de apresentar programação para crianças, foi possível notar que os participantes consideraram o propósito de ensinar o conceito de algoritmo. O conceito mais claro para os dois grupos foi o de lateralidade, além de assimilarem o conceito, implementaram por meio do percurso da tartaruga. Houve especial preocupação em como o jogador deveria percorrer o caminho inserindo as setas em ordem, antes do personagem andar.

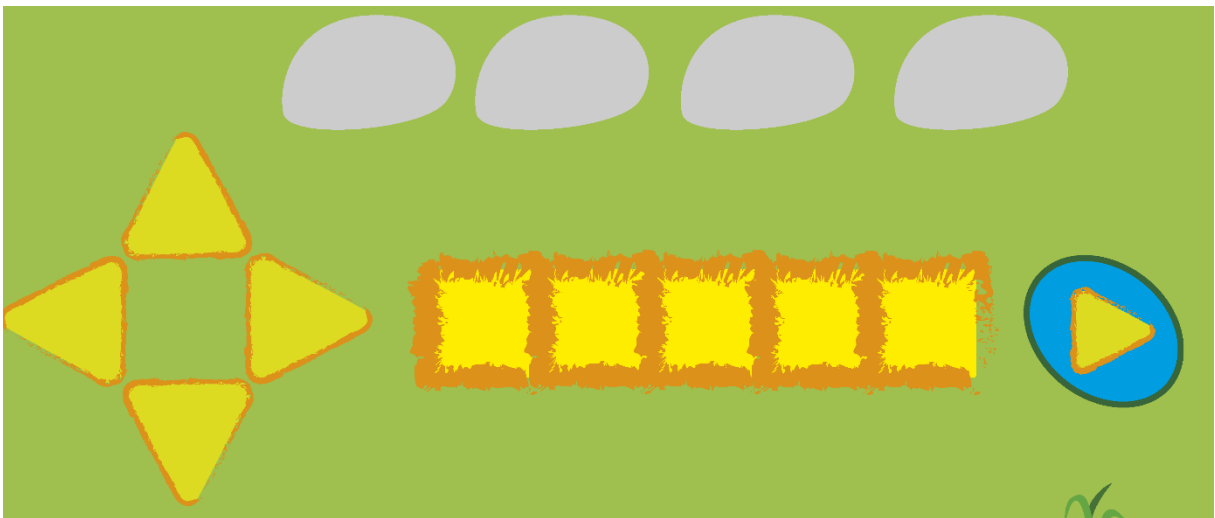


## 5. ARTEFATO

Neste capítulo são abordados os aspectos de desenvolvimento do artefato proposto, o projeto do jogo na Seção 5.1, o desenvolvimento na Seção 5.2 e a relação entre artefato gerado e técnicas na Seção 5.3.

### 5.1. PROJETO

Partindo da análise realizada na Seção 4.3, foi possível desenvolver o roteiro de um jogo. O resultado foi o “Jogo da Tartaruga”, um jogo em que o usuário deve percorrer um caminho para levar uma tartaruga de volta ao seu lar. Para tanto o jogador deve atribuir comandos com as direções (setas) que possibilitem que a tartaruga faça todo o percurso. Os comandos são inseridos em uma lista e posteriormente são executados, caso o caminho esteja correto o jogador irá para próxima fase, caso contrário perde uma vida e deve tentar novamente. Os elementos citados estão representados no Figura 11. Caso o jogador perca todas as vidas, deve voltar a fase inicial, independente da fase em que tenha morrido. O jogador vence caso consiga chegar à fase final.



*Figura 11 - Elementos: Caminho, Setas, Lista e Botão Executar (Fonte: Autoria Própria)*

Um ponto do projeto a ser salientado foi o uso de textos e mensagens escritos em letras de forma no jogo, visto que a primeira alfabetização é realizada com este tipo de letra, para após ensinar o uso de letras cursivas.

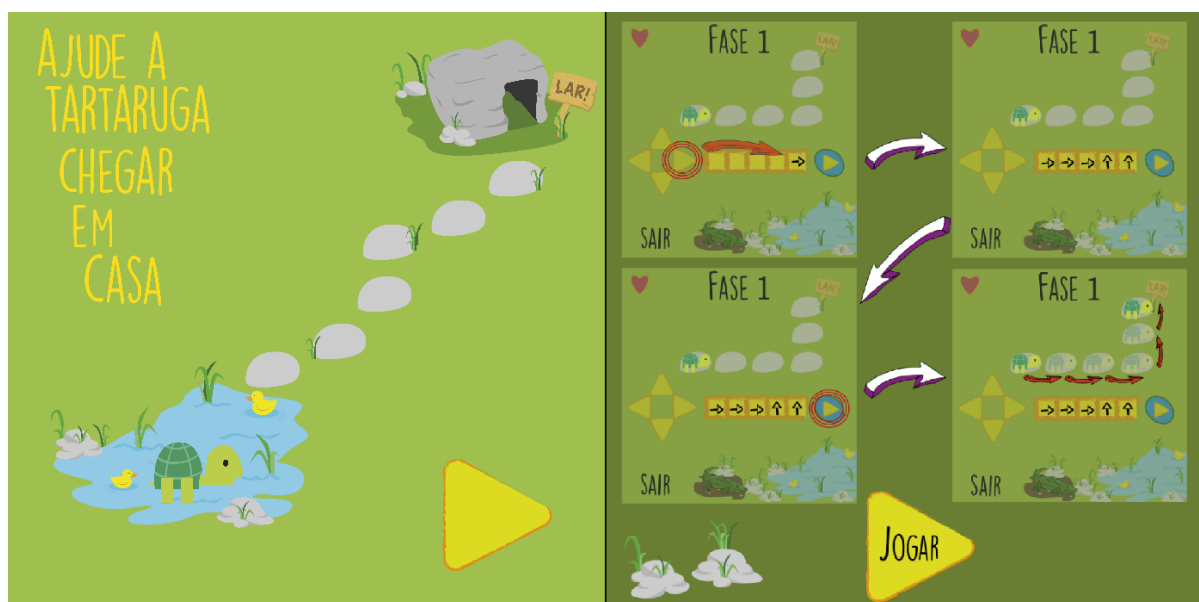
Com o roteiro, descrito acima, pronto, foi possível desenvolver a interface do jogo, que é composta basicamente de uma tela inicial, uma tela de créditos, um menu de fases (das diversas

fases do jogo) e de uma tela para vitória e outra para derrota. A relação de telas e respectivos fluxos são apresentados na Figura 12.



**Figura 12 - Telas do jogo (Fonte: Autoria Própria)**

Também foram desenhadas duas telas de tutorial de jogo (Figura 13), nas quais, no caso de ser um primeiro acesso ao programa, são apresentados ao jogador os comandos básicos do jogo.



**Figura 13 - Telas de tutorial (Fonte: Autoria Própria)**

Com o roteiro e interface prontos, faltou tornar o jogo funcional. O planejado era desenvolver o artefato utilizando a linguagem Scratch, porém com o projeto pronto foram constatadas limitações que o Scratch apresenta para o desenvolvimento do jogo, como:

- Controlar uma variável de vida em diversas fases;
- Trabalhar com *layers* de maneira dinâmica;
- Trabalhar com um *array* de comandos.

Como alternativa ao Scratch optou-se por usar a plataforma para jogos Construct 2, que supre todas as limitações apresentadas. O Construct está descrito na Seção 5.1.1.

### 5.1.1. Construct 2

O Construct 2 (Figura 14) é uma plataforma de desenvolvimento de jogos baseada em HTML5, desenvolvida e distribuída pela empresa Scirra. O desenvolvimento é baseado em blocos, não requerendo codificação (SCIRRA, 2016).

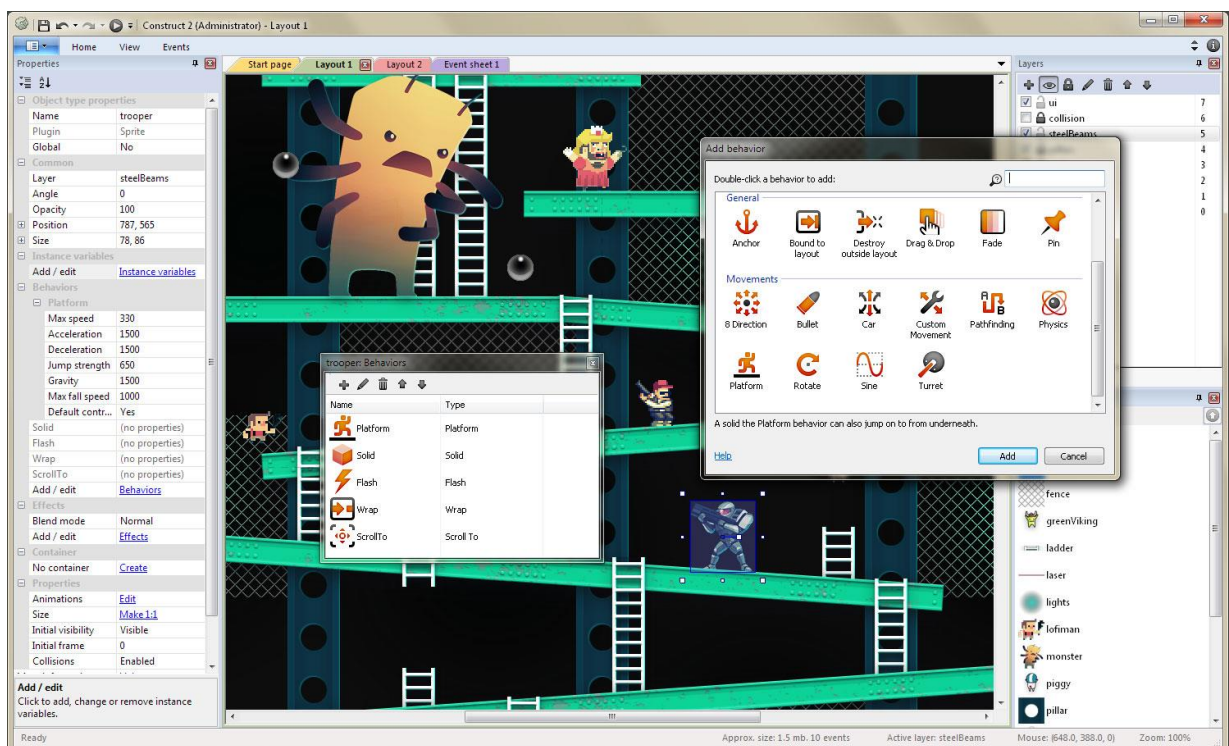
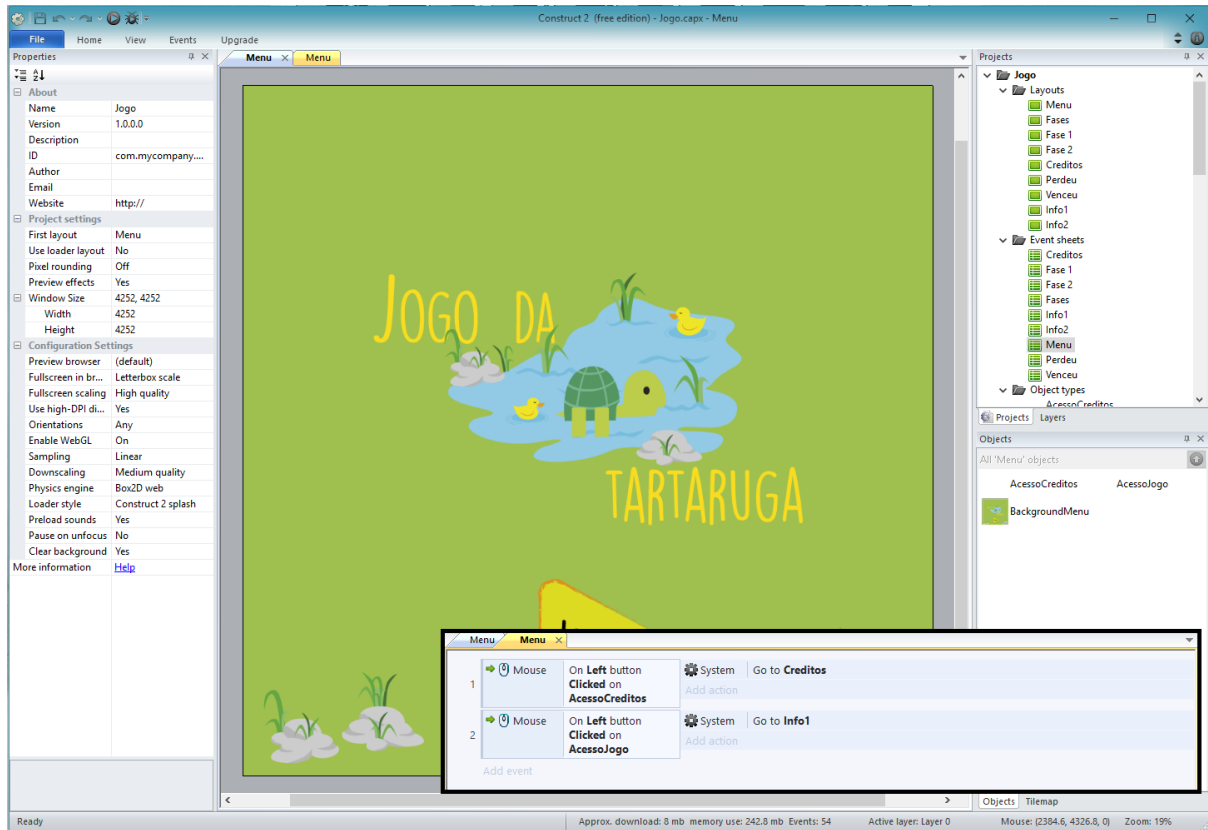


Figura 14 - Construct 2 (Fonte: <https://www.scirra.com/>)

Os jogos desenvolvidos com Construct 2 possuem três camadas de trabalho. A primeira chamada de *Layouts*, consiste nas telas do jogo, a parte visual do desenvolvimento. Nessa camada são inseridos todos os objetos que fazem parte do programa. A segunda camada é chamada de *Object types*, no qual todos os objetos inseridos na camada anterior ficam

armazenados. Por fim, tem-se a camada *Event sheets*, é nessa camada que todos os eventos são registrados por meio de blocos relacionados a objetos inseridos nos *layouts* (SCIRRA, 2016). Na Figura 15 é possível ver um exemplo de *Layout* ligado a um *Event sheet*.



**Figura 15 - Exemplo de Layout e Event Sheet (Fonte: <https://www.scirra.com/>)**

Conforme já mencionado, a codificação não é requerida para trabalhar com Construct 2 já que o desenvolvimento é baseado em blocos. Cada bloco está relacionado a um evento no sistema ou a objeto no jogo. Os eventos consistem de condições que testam se alguns critérios foram satisfeitos, se todas as condições forem verdadeiras, as ações dos eventos serão executadas, e após executar as ações, quaisquer sub-eventos também serão executados, e estes por sua vez poderão testar mais condições, e assim por diante (SCIRRA, 2016).

O Construct 2 possui uma versão gratuita para testes e uma paga, as duas se diferem na medida que a versão gratuita possui uma série de limitações em relação a paga. Algumas limitações da versão gratuita são: número de eventos limitados a 100, número de *layers* limitado a 4, não existe acesso ao *debugger*, e não é possível construir jogos *multiplayer* (SCIRRA, 2016).

## 5.2. DESENVOLVIMENTO

O desenvolvimento das telas que não correspondem a fases do jogo foi relativamente simples e consistiu apenas em blocos de comando que indicavam a troca de tela caso um botão fosse pressionado.

A maior complexidade ficou nas telas de fases do jogo (percurso da tartaruga), nas quais foram usados 40 eventos para todas as funcionalidades de uma fase do jogo. O maior número de eventos foi utilizado na manipulação do vetor que guarda as direções que o jogador apontou para então serem executadas ao apertar o botão de executar. A Figura 16 apresenta todos os blocos de eventos utilizados para a fase 1 do jogo.

Na Figura 16, o primeiro elemento que está em um bloco verde representa uma variável que guarda o número de vidas do jogador. A linha 1 é um evento para o botão sair da fase, caso esse evento seja registrado o sistema retorna a tela das fases. Na linha 2 é realizada a inicialização do vetor que guardará as posições inseridas pelo jogador. Na linha 3 está sendo impresso na tela da fase o número de vidas do jogador. Da linha 4 a linha 23 está sendo realizado uma série de comparações para imprimir na tela o valor presente no vetor de direções.

Na linha 24 da Figura 16, caso o jogador chegue na linha de chegada do caminho, o sistema altera para tela de vitória. Na linha 25 o sistema, caso o número de vidas do jogador chegue a zero, altera para tela de derrota. Da linha 26 a linha 29, estão sendo registrados os clicks do jogador nas setas dentro do vetor de direções. Da linha 30 a linha 40, ao jogador pressionar o botão de executar, o sistema executa as direções presentes no vetor ao mesmo tempo que realiza uma série de comparações para verificar se a direção seguida está correta, caso esteja incorreta o sistema reduz uma vida do jogador.

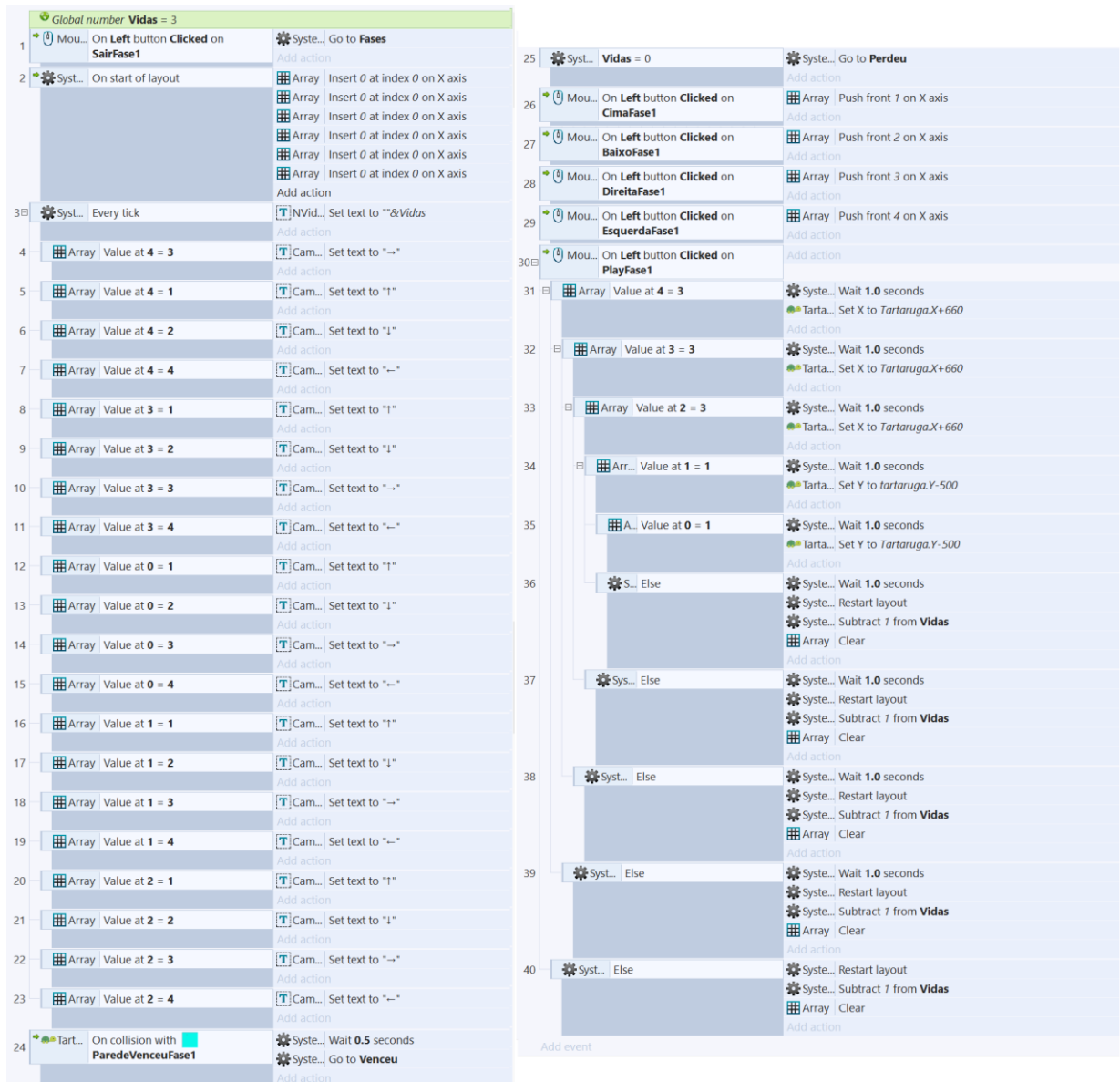


Figura 16 - Event sheet Fase 1 (Fonte: Autoria Própria)

### 5.3. ARTEFATO X TÉCNICAS

É possível traçar relações entre as técnicas que foram desenvolvidas e descritas no Capítulo 4 e o artefato gerado. Utilizando a denominação de grupos do Capítulo 4, o menu de fases, apresentado na Figura 17, foi sugerido pelo grupo B, os participantes indicaram que o jogo deveria possuir um menu com diversas fases (item 1, Figura 17), porém essas fases deveriam ser jogadas em ordem sequencial (item 2, Figura 17) com a dificuldade sendo incrementada a cada nova fase.



Figura 17 - Componentes do Menu de Fases (Fonte: Autoria Própria)

A tela de percurso da tartaruga, representada pela “Fase 1” na Figura 18, possui componentes retirados das atividades realizadas pelos dois grupos. Analisando cada item apresentado na Figura 18, pode-se concluir:

- Item 1: Consiste no caminho que o personagem deve percorrer para atingir o objetivo. Este *desing*, foi retirado do modelo resultante do grupo B, no qual eles sugeriram um caminho ondulado para o personagem percorrer. Porém como um dos conceitos que pretende-se ensinar com o jogo é a lateralidade, optou-se pelo desenvolvimento de um caminho linear com ângulos retos. Esse item se originou da técnica *PICTIVE*.
- Item 2: Corresponde a setas de direção. Esse componente foi indicado tanto pelo grupo A como o B, e é a forma que o jogador tem para movimentar o personagem no plano, para cima, baixo, esquerda e direita. Esse item se originou da técnica *PICTIVE*.
- Item 3: É a barra que exibe a direções inseridas pelo usuário. Esse componente foi inserido por conta, pois havia a necessidade de apresentar ao jogador qual o caminho ele digitou. Esse item se originou da técnica *Interface Metaphors Game + CARD Game*.

- Item 4: É o botão de executar, após inserir todas as direções na barra, o jogador aperta esse botão para executar os movimentos indicados. Esse botão foi sugerido pelo grupo B, onde o jogador usaria um botão para executar todos os comandos. Esse item se originou da técnica *Interface Metaphors Game + CARD Game*.
- Item 5: Corresponde ao indicador do número de vidas que o jogador possui. Este *desing*, foi retirado do modelo resultante do grupo A, que definiram que o jogador teria um número limite de vidas para percorrer todas as fases, caso o jogador erre o caminho ele perde uma vida, e caso perdesse todas a vidas teria de recomeçar o jogo da primeira fase novamente. Esse item se originou da técnica *PICTIVE*.



Figura 18 - Componentes do Jogo Fase 1 (Fonte: Autoria Própria)

O objetivo do jogo é o ensino de conceitos básicos de programação como comando, instrução, finitude, erro e lateralidade, visto isso algumas modificações foram realizadas quanto aos projetos resultantes das técnicas participativas. Porém buscou-se sempre manter a base do jogo nos projetos gerados pelas duas equipes, sem fugir do *design* apresentado por eles durante as técnicas de Design Participativo.



## 6. VALIDAÇÃO PRÉVIA

Conforme o último objetivo específico, inicialmente a intenção era realizar a validação do artefato com o público-alvo para o qual ele foi desenvolvido, porém notou-se que antes de validar o artefato com crianças seria importante ter o parecer de pessoas da área de educação infantil e de profissionais atrelados a área de ensino de programação. Desta forma foram realizadas 5 entrevistas com 3 profissionais da área de educação e 2 na área de ensino de programação. As entrevistas foram realizadas de maneira individual durante os dias 1 e 2 de dezembro.

A validação foi feita de maneira qualitativa, por meio de uma entrevista guiada, presente no Apêndice A deste trabalho. A entrevista foi composta de 10 questões divididas em três grupos de interesse: legibilidade, jogabilidade e aprendizagem (SALEN; ZIMMERMAN, 2012). A legibilidade diz respeito a interface do jogo, jogabilidade refere-se à interação do usuário com o programa, e a aprendizagem aborda a questão do ensino. Esses conceitos emergiram dos processos das atividades participativas.

O processo da entrevista iniciou com os entrevistados testando o jogo, após isso, os dados de formação acadêmica e pessoais dos entrevistados foram coletados e as questões foram arguidas. A amostra entrevistada foi composta de duas professoras formadas em Pedagogia (denominados entrevistados A e B), uma estudante de mestrado formada em Licenciatura em Línguas Inglês (denominado entrevistado C), um graduando em Sistemas de Informação formado em Licenciatura em Ciências Biológicas (denominado entrevistado D), e um estudante de doutorado formado em Ciências da Computação e mestre em Engenharia de Computação (denominado entrevistado E).

Na Seção 6.1 estão os apontamentos quanto a legibilidade do jogo, na Seção 6.2 discutem-se as questões de jogabilidade, na Seção 6.3 refere-se as questões de aprendizagem e ensino, e por fim, na Seção 6.4 são apontadas as sugestões de mudança formuladas após a análise das entrevistas.

### 6.1. LEGIBILIDADE

Do ponto de vista da legibilidade do jogo, foram realizadas perguntas sobre a interface do jogo, a disposição dos menus e botões, e fluxo de telas, conforme o questionário do Apêndice A.

Quanto a interface do jogo, houve unanimidade em apontar que o jogo possui uma interface simples e de fácil entendimento, porém um pouco confusa na parte do tutorial, pois esta tela dá a impressão que o usuário já está na tela de jogo.

Os botões são de fácil compreensão, e não necessitam de linguagem escrita. O entrevistado D fez uma ressalva, que os botões deveriam ser mais destacados para diferenciá-los de outros elementos na tela.

O fluxo de telas é bem intuitivo, exceto pelo tutorial já citado anteriormente (situação destacada por todos os entrevistados), que pode causar confusão.

Em vista da legibilidade algumas sugestões de modificações e acréscimos foram feitas, entre elas: a substituição total da linguagem escrita por símbolos (indicação do entrevistado E), e a melhora do tutorial, usando talvez um tutorial interativo (indicação dos entrevistados D e E).

Em uma análise geral no quesito legibilidade, o jogo foi considerado positivo, os entrevistados conseguiram navegar tranquilamente pelas telas, e compreender os objetivos.

## 6.2. JOGABILIDADE

As questões 2.1, 2.2 e 2.3, referentes a jogabilidade abordaram os pontos de objetivos (reconhecer e executar um caminho), comandos e dificuldade do jogo.

Os objetivos do jogo ficaram claros para todos os entrevistados, porém os entrevistados B, D e E constataram que deveria ficar mais claro como atingir esse objetivo. Os comandos também ficaram bem claros, e a simbologia bem entendível.

Com respeito a dificuldade de jogo, todos os entrevistados concordaram que o jogo está simples, e será de fácil entendimento para os jogadores do público-alvo.

Quanto a modificações e acréscimos, foram sugeridos dois pontos importantes, primeiro bloquear o botão de executar até que todos os comandos de direção estejam inseridos (sugestão dos entrevistados A e E), e o segundo ponto importante a ser tratado é o *feedback* do erro, visto que o jogador não possui nenhuma indicação de onde errou (sugerido por todos os entrevistados).

### 6.3. APRENDIZAGEM

Quanto a aprendizagem questionou-se, nas perguntas 3.1 e 3.2, se por meio do jogo seria possível apresentar os conceitos de comando, instrução, “finitude”, erro, lateralidade e algoritmo.

Os entrevistados foram unânimes em afirmar que o jogo consegue apresentar esses conceitos. Os pontos mais presentes são o da lateralidade e do algoritmo, que são bem expressos quando trabalha a direção que o personagem deve percorrer, e que o jogador deve inserir todos as direções antes de executar o caminho.

### 6.4. SUGESTÕES DE MUDANÇA

A partir das entrevistas foi possível apontar uma série de alterações para tornar o jogo mais apropriado para o público alvo. As sugestões de mudança são:

- Destacar os botões para diferencia-los dos demais elementos do jogo;
- Fazer maior uso de símbolos ao invés de linguagem escrita;
- Bloqueio do botão executar até todos os comandos de direção estarem inseridos;
- Mudança do tutorial, de forma a explicar melhor a jogabilidade;
- Inserir o *feedback* do erro, para o jogador visualizar onde errou.

A sugestões fizeram referência a legibilidade e jogabilidade do jogo, visto que no quesito aprendizagem o jogo obteve avaliações positivas.

## 7. CONCLUSÕES

A computação possui presença no cotidiano das pessoas, visto isso, o ensino de programação para crianças atua como uma ferramenta de domínio e entendimento do mundo. Nesse contexto os jogos são uma forma lúdica de transmitir um conhecimento, de maneira dinâmica, além de estimular a criatividade e cooperação.

Durante a elaboração deste trabalho, ficaram claros alguns pontos em relação ao desenvolvimento de ferramentas voltadas ao ensino de programação. Um ponto relevante observado é o fato de que possuir uma ferramenta de ensino dita de última geração não é garantia do aprendizado do estudante. Nesse aspecto, encontrou-se um potencial que pode ser explorado e desenvolvido, porém que não é fácil de ser acessado.

Desta forma, o objetivo geral desta pesquisa, que envolvia apresentar programação para crianças utilizando técnicas participativas, foi atingido por meio do desenvolvimento de um jogo.

As técnicas participativas para apoiar o desenvolvimento do artefato proposto foram definidas em estudo de referencial bibliográfico, em que foram selecionadas práticas adequadas para produzir o artefato com base no público-alvo da pesquisa como crianças estudantes do ensino básico e fundamental, com faixa etária de 5 a 9 anos.

A realização de atividades com membros do público-alvo, usando técnicas de Design Participativo selecionadas para o desenvolvimento de jogos, resultou em dois modelos que com devidas análises permitiram propor um jogo baseado no que os participantes produziram.

O jogo proposto foi desenvolvido utilizando a ferramenta Construct 2, alternativa a linguagem de programação Scratch, que foi identificada como limitada para o escopo da pesquisa.

A intenção, como proposto nos objetivos específicos, era realizar a validação do artefato. Notou-se que antes de realizar tal validação com crianças, era necessário ter o parecer de profissionais da área de educação infantil e também de profissionais atrelados a área de ensino de programação. Desta forma, a validação foi realizada com essas pessoas.

Os resultados obtidos com a validação foram positivos, o jogo foi considerado adequado para o ensino de programação para crianças, algumas críticas foram levantadas nos quesitos de legibilidade e jogabilidade.

Durante o desenvolvimento desta pesquisa revelaram-se algumas dificuldades, entre elas é possível citar a busca por alguns conteúdos, e a validação com crianças da educação infantil.

### 7.1. TRABALHOS FUTUROS

Como trabalhos futuros podem ser citados: a realização das mudanças sugeridas na Seção 6.4 deste trabalho, ou seja, destacar os botões para diferencia-los dos demais elementos do jogo; fazer maior uso de símbolos ao invés de linguagem escrita; bloqueio do botão executar até todos os comandos de direção estarem inseridos; mudança do tutorial, de forma a explicar melhor a jogabilidade; e inserir o *feedback* do erro, para o jogador visualizar onde errou. Também se sugere a validação do artefato com o público-alvo, como era inicialmente pretendido no último objetivo específico.

Outra sugestão é estudar a inserção de outras estruturas de programação no jogo, como condicionais e pontos de intersecção. O jogo foi desenvolvido para trabalhar em interface Web, então seria interessante estudar sua migração para dispositivos móveis.

O exemplo mostrado durante as atividades foi o Roboqueda, então seria interessante realizar novamente as técnicas, porém adotando novos modelos, além do Roboqueda. E como forma de uma validação complementar, realizar testes de aprendizagem com grupos de controle.

## REFERÊNCIAS

- ADACHI, P. J. C.; WILLOUGHBY, T. More than just fun and games: The longitudinal relationships between strategic video games, self-reported problem solving skills, and academic grades. **Journal of Youth and Adolescence**, p. 1041–1052, jul. 2013.
- ALENCAR, G. A. et al. Utilizando o SCRATCH nas aulas de Lógica de Programação do Proeja: Um relato de experiência. **Nuevas Ideas en Informática Educativa TISE**, p. 542–545, 2014.
- ANTONIOU, A. et al. An approach for serious game development for cultural heritage: Case study for an archaeological site and museum. **Journal on Computing and Cultural Heritage (JOCCH)**, p. 19, 2013.
- ANTUNES, A. R. et al. **Computação Desplugada Aplicada para Pessoas com Altas Habilidades**. UTFPR, 2014. Disponível em: <<http://arcaz.dainf.ct.utfpr.edu.br/rea/files/original/8314d733b112fadf656aa7195ec7fdda.pdf>>
- AURELIANO, V. C. O.; TEDESCO, P. C. DE A. R. Avaliando o uso do Scratch como abordagem alternativa para o processo de ensino-aprendizagem de programação. **Congresso da Sociedade Brasileira de Computação**, v. 32, p. 1–10, 2012.
- BARCELOS, T. S.; SILVEIRA, I. F. Pensamento Computacional e Educação Matemática: Relações para o Ensino de Computação na Educação Básica. **XX Workshop sobre Educação em Computação**, p. 10, 2012.
- BELL, T. et al. Computer science unplugged: School students doing real computing without computers. **The New Zealand Journal of Applied Computing and Information Technology**, p. 20–29, 2009.
- BELL, T.; WITTEN, I. H.; FELLOWS, M. **Computer Science Unplugged. Ensinando Ciência da Computação sem o uso do computador**, 2011. Disponível em: <<http://csunplugged.org/wp-content/uploads/2014/12/CSUnpluggedTeachers-portuguese-brazil-feb-2011.pdf>>
- BELLOTTI, F. et al. A serious game model for cultural heritage. **Journal on Computing and Cultural Heritage (JOCCH)**, p. 27, 2012.
- BORGES, M. A. F. Avaliação de uma metodologia alternativa para a aprendizagem de programação. **VIII Workshop de Educação em Computação–WEI**, n. 8, p. 15, 2000.
- BOULTON, M. **Zamenhof: creator of Esperanto**. [s.l.] UPS Blackburn, 1980.
- BOYD, L. E. et al. Evaluating a Collaborative iPad Game’s Impact on Social Relationships for Children with Autism Spectrum Disorder. **ACM Transactions on Accessible Computing (TACCESS)**, p. 18, 2015.
- CAMARGO, L. S. DE A.; FAZANI, A. J. Explorando o Design Participativo como Prática de Desenvolvimento de Sistemas de Informação. **Revista de Ciência da Informação e Documentação**, v. 5, n. 1, p. 138–150, 2014.

- COENEN, T.; MOSTMANS, L.; NAESSENS, K. MuseUs: Case study of a pervasive cultural heritage serious game. **Journal on Computing and Cultural Heritage (JOCCH) - Special issue on serious games for cultural heritage**, p. 19, 2013.
- COLOMBO, C. DA S. O Ensino de Programação de Computadores baseado em Jogos. **Anais do Encontro Virtual de Documentação em Software Livre e Congresso Internacional de Linguagem e Tecnologia Online**, v. 4, n. 1, p. 5, 2015.
- SCIRRA LTD. **Construct 2**. 2016. Disponível em: <<https://www.scirra.com/construct2>>.
- COUTINHO, C. P.; LISBÔA, E. S. Sociedade da informação, do conhecimento e da aprendizagem: desafios para educação no século XXI. **Revista de Educação**, v. 18, n. 1, p. 5–22, 2011.
- DA SILVA, R. E.; MARTINS, S. W. Ensino de Ciência da Computação através do Desenvolvimento de Jogos. **RIBIE**, p. 1286–1295, 2004.
- DE OLIVEIRA, M. L. S. et al. Ensino de lógica de programação no ensino fundamental utilizando o Scratch: um relato de experiência. **XXXIV Congresso da Sociedade Brasileira de Computação**, n. 34, p. 1493–1502, 2014.
- DIMURO, G. P.; COSTA, A. C. DA R.; RODRIGUES, F. DE P. M. Uma experiência de ensino integrado dos fundamentos matemáticos da ciência da computação. **Revista Brasileira de Informática na Educação**, v. 7, n. 1, p. 65–75, 2000.
- DOS SANTOS, L. G. O.; JÚNIOR, E. R. DE O. Desenvolvendo lógica no ensino fundamental: uma experiência com Scratch. **Semex em Resumos**, v. 3, n. 3, 2015.
- EHN, P. **Work-oriented design of computer artifacts**. Hillsdale, NJ, USA: L. Erlbaum Associates Inc, 1988.
- FARDO, M. L. A gamificação aplicada em ambientes de aprendizagem. **RENOTE**, v. 11, n. 1, p. 9, 2013.
- FERNANDES, L. D.; FURQUIM, A. A.; BARANAUSKAS, M. C. C. Jogos no computador e a formação de recursos humanos na indústria. **VI Simpósio Brasileiro de Informática na Educação**, 1995.
- FIGUEIREDO, K. DA S. et al. Uma Abordagem Gamificada para o Ensino de Programação Orientada a Objetos. **SIBGRAPI**, p. 10, 2015.
- FRANÇA, R. S. DE; SILVA, W. C. DA; AMARAL, H. J. C. DO. Ensino de ciência da computação na educação básica: Experiências, desafios e possibilidades. **XX Workshop sobre Educação em Computação**, p. 4, 2012.
- FROSCHAUER, J. et al. Art history concepts at play with ThIATRO. **Journal on Computing and Cultural Heritage (JOCCH) - Special issue on serious games for cultural heritage**, p. 15, 2013.

- GERON, G.; SDROIEVSKI, N. M.; AMARAL, M. A. **Computando Culturas em Equidade em um Programa de Educação Tutorial**. 2014. Disponível em: <<http://arcaz.dainf.ct.utfpr.edu.br/rea/items/show/112>>. Acesso em: 11/10/2016.
- GESTWICKI, P.; MCNELLY, B. Interdisciplinary Projects in the Academic Studio. **ACM Transactions on Computing Education (TOCE)**, p. 24, 2016.
- GIBSON, J. P. Teaching Graph Algorithms to Children of All Ages. **Proceedings of the 17th ACM Annual Conference on Innovation and Technology in Computer Science Education**, v. 12, p. 34–39, 2012.
- HAO, J.; LEUNG, H.-F. Achieving Socially Optimal Outcomes in Multiagent Systems with Reinforcement Social Learning. **ACM Transactions on Autonomous and Adaptive Systems (TAAS)**, p. 23, 2013.
- HINTERHOLZ, O. Tepequém: uma nova Ferramenta para o Ensino de Algoritmos nos Cursos Superiores em Computação. **XVII Anais do Workshop sobre Educação em Informática**, p. 485–488, 2009.
- HUANG, C.-H.; HUANG, J. An annales school-based serious game creation framework for taiwanese indigenous cultural heritage. **Journal on Computing and Cultural Heritage (JOCCH) - Special issue on serious games for cultural heritage**, p. 31, 2013.
- JOHNSON, J. S.; NEWPORT, E. L. Critical period effects in second language learning: The influence of maturational state on the acquisition of English as a second language. **Cognitive Psychology**, p. 60–99, 1989.
- JÚNIOR, J. C. R. P. et al. Ensino de algoritmos e programação: uma experiência no nível médio. **XIII Workshop de Educação em Computação**, p. 2351–2362, 2005.
- JÚNIOR, J. C. R. P.; RAPKIEWICZ, C. E. O processo de ensino-aprendizagem de fundamentos de programação: Uma visão crítica da pesquisa no brasil. **Anais do XII Workshop sobre Educação em Computação**, v. 7, n. 1, p. 28, 2004.
- KENSING, F.; BLOMBERG, J. Participatory design: Issues and concerns. **Computer Supported Cooperative Work**, p. 167–185, 1998.
- LAKATOS, E. M.; MARCONI, M. DE A. **Metodologia Científica**. 6. ed. São Paulo: Atlas, 2005.
- LARRAZA-MENDILUZE, E. et al. Using a Real Bare Machine in a Project-Based Learning Environment for Teaching Computer Structure: An Analysis of the Implementation Following the Action Research Model. **ACM Transactions on Computing Education (TOCE)**, p. 17, 2016.
- MALONEY, J. et al. The Scratch Programming Language and Environment. **ACM Transactions on Computing Education**, v. 10, n. 4, p. 16, 2010.
- MARJI, M. **Aprenda a Programar com Scratch: Uma introdução visual à programação com jogos, arte, ciência e matemática**. São Paulo: Novatec, 2014.



- MARTINS, T. M. DE O. et al. A Gamificação de conteúdos escolares: uma experiência a partir da diversidade cultural brasileira. **X Seminário de Jogos Eletrônicos, Educação e Comunicação**, p. 10, 2014.
- MELO, A. M.; BARANAUSKAS, M. C. C. Design para a inclusão: desafios e proposta. **Proceedings of VII Brazilian symposium on Human factors in computing systems**, v. 6, p. 11–20, 2006.
- MÉLO, F. É. N. DE et al. Do Scratch ao Arduino: Uma Proposta para o Ensino Introdutório de Programação para Cursos Superiores de Tecnologia. **XXXIX Congresso Brasileiro de Educação em Engenharia**, p. 10, 2011.
- MONROY-HERNÁNDEZ, A. ScratchR: sharing user-generated programmable media. **Proceedings of the 6th international conference on Interaction design and children**, v. 7, p. 167–168, 2007.
- MORATORI, P. B. **Por que utilizar jogos educativos no processo de ensino aprendizagem**. Tese de Mestrado - Rio de Janeiro: UFRJ, 2003.
- MOREIRA, D. Elementos para um plano de melhoria do ensino universitário ao nível de instituição. **Revista IMES**, v. 9, p. 28–32, 1986.
- MOREIRA, M. P.; FAVERO, E. L. Um ambiente para ensino de programação com feedback automático de exercícios. **Workshop sobre Educação em Computação**, v. 17, p. 429–438, 2009.
- MOTA, M. P.; PEREIRA, L. W. K.; FAVERO, E. L. JavaTool: Uma Ferramenta para o Ensino de Programação. **XXVIII Congresso da Sociedade Brasileira de Computação**, p. 127–136, 2008.
- MULLER, M. J.; HASLWANTER, J. H.; DAYTON, T. **Participatory practices in the software lifecycle**. [s.l: s.n.]. v. 2, 1997.
- MULLER, M. J.; WILDMAN, D. M.; WHITE, E. A. Participatory design through games and other group exercises. **CHI '94 Conference Companion on Human Factors in Computing Systems**, p. 411–412, 1994.
- PAPERT, S. **Máquina das Crianças: Repensando a Escola na Era da Informática**. Porto Alegre: Artes Médicas, 1994.
- RAPKIEWICZ, C. E. et al. Estratégias pedagógicas no ensino de algoritmos e programação associadas ao uso de jogos educacionais. **RENOTE**, v. 4, n. 2, p. 11, 2006.
- REPENNING, A. et al. Scalable Game Design: A Strategy to Bring Systemic Computer Science Education to Schools through Game Design and Simulation Creation. **ACM Transactions on Computing Education (TOCE) - Special Issue II on Computer Science Education in K-12 Schools**, p. 31, 2015.
- RESNICK, M. Scratch: programming for all. **Communications of the ACM**, v. 52, n. 11, p. 60–67, 2009.

- RODRIGUES, M. C. Como Ensinar Programação? **ULBRA**, v. 1, n. 1, 2002.
- RONCA, A. C. C.; ESCOBAR, V. F. **Técnicas pedagógicas: domesticação ou desafio à participação**. 2. ed. Petrópolis: Vozes, 1980.
- RUBINO, I. et al. Integrating a Location-Based Mobile Game in the Museum Visit: Evaluating Visitors' Behaviour and Learning. **Journal on Computing and Cultural Heritage (JOCCH)**, p. 18, 2015.
- SAJANIEMI, J.; KUITTINEN, M. Program animation based on the roles of variables. **Proceedings of the 2003 ACM symposium on Software visualization**, p. 7–ff, 2003.
- SALEN, K.; ZIMMERMAN, E. **Regras do jogo: fundamentos do design de jogos**. São Paulo: Blucher, 2012. v. 1
- SANTOS, S. C. DOS. O processo de ensino-aprendizagem e a relação professor-aluno: aplicação dos “sete princípios para a boa prática na educação de ensino superior”. **Caderno de pesquisas em administração**, v. 8, n. 1, p. 69–82, 2001.
- SCAICO, P. D. et al. Relato da Utilização de uma Metodologia de Trabalho para o Ensino de Ciência da Computação no Ensino Médio. **Anais do Workshop de Informática na Escola**, v. 1, n. 1, p. 10, 2012a.
- SCAICO, P. D. et al. Um Relato de Experiências de Estagiários da Licenciatura em Computação com o Ensino de Computação para Crianças. **RENOTE**, v. 10, n. 3, p. 12, 2012b.
- SCAICO, P. D. et al. Ensino de Programação no Ensino Médio: Uma Abordagem Orientada ao Design com a linguagem Scratch. **Revista Brasileira de Informática na Educação**, v. 21, n. 2, p. 92, 2013.
- SEEHORN, D. et al. **CSTA K-12 Computer Science Standards**. Computer Science Teachers Association, 2011. Disponível em: <<http://www.education2020.ca/Content/K-12ModelCurrRevEd.pdf>>
- SEIXAS, L. DA R. et al. Gamificação como Estratégia no Engajamento de Estudantes do Ensino Fundamental. **Anais do Simpósio Brasileiro de Informática na Educação**, v. 25, n. 1, p. 559, 2014.
- SETTI, M. DE O. G. **O Processo de Discretização do Raciocínio Matemático na Tradução para o Raciocínio Computacional: Um Estudo de Caso no Ensino/Aprendizagem de Algoritmos**. Pós-Graduação - Curitiba: Universidade Federal do Paraná, 2009.
- SPINUZZI, C. A Scandinavian challenge, a US response: methodological assumptions in Scandinavian and US prototyping approaches. **Proceedings of the 20th annual international conference on Computer documentation**, p. 208–215, 2002.
- TAROUCO, L. M. R. et al. Jogos educacionais. **CINTED**, v. 2, n. 1, p. 7, 2004.
- VALENTE, J. A. et al. **O computador na sociedade do conhecimento**. Campinas: Unicamp/NIED, 1999.

WANG, A. I. Extensive Evaluation of Using a Game Project in a Software Architecture Course. **ACM Transactions on Computing Education (TOCE)**, p. 28, 2011.

WERNER, L.; DENNER, J.; CAMPE, S. Children Programming Games: A Strategy for Measuring Computational Learning. **ACM Transactions on Computing Education (TOCE)**, p. 22, 2014.

WING, J. M. Computational thinking. **Communications of the ACM**, v. 49, n. 3, p. 33–35, 2012.

APÊNDICE A – Entrevista Guiada

## QUESTIONÁRIO DE AVALIAÇÃO

Autores: Lucas Cyulik

Orientadora: Marília A. Amaral

Entrevistado: \_\_\_\_\_

Questionário de avaliação do jogo para ensino de programação para crianças desenvolvido como projeto de TCC do aluno de Sistemas de Informação da UTFPR (Universidade Tecnológica Federal do Paraná) Lucas Cyulik, orientado pela professora doutora Marília A. Amaral.

### 1. LEGIBILIDADE

- 1.1. O jogo possui uma interface de fácil entendimento?
- 1.2. Os botões nos menus são entendíveis?
- 1.3. O jogo apresenta um fluxo intuitivo?
- 1.4. Existe algum elemento que deveria ser modificado ou poderia ser acrescentado?

### 2. JOGABILIDADE

- 2.1. Os objetivos do jogo são claros?
- 2.2. Os comandos são claros?
- 2.3. A dificuldade de jogabilidade é condizente com a idade do público-alvo do jogo?
- 2.4. Algum aspecto de jogabilidade deveria ser modificado ou poderia ser acrescentado?

### 3. APRENDIZAGEM

- 3.1. O jogador terá condições de entender conceitos de comando, instrução, “finitude”, erro, lateralidade?
- 3.2. Segundo ASCENCIO (2002, p. 2): “Um **algoritmo** é a descrição de uma sequência de passos que deve ser seguida para a realização de uma tarefa.”. Você acredita que o jogo consegue apresentar o conceito de algoritmo citado?

(ASCENCIO, 2002) ASCENCIO, Ana Fernandes Gomes. *Lógica de Programação com Pascal*. São Paulo: Ed. Person Education do Brasil. 2002.