



**UNIVERSIDADE ESTADUAL DA PARAÍBA – UEPB
CENTRO DE CIÊNCIAS EXATAS E SOCIAIS APLICADAS – CCEA
CAMPUS VII – GOVERNADOR ANTÔNIO MARIZ
CURSO DE LICENCIATURA EM COMPUTAÇÃO**

CLECIA LEITE RODRIGUES

**AVALIAÇÃO DE *SOFTWARE* EDUCACIONAL COM BASE NOS CONCEITOS DO
DESIGN EMOCIONAL**

**PATOS – PB
2016**

CLECIA LEITE RODRIGUES

AVALIAÇÃO DE *SOFTWARE* EDUCACIONAL COM BASE NOS CONCEITOS DO
DESIGN EMOCIONAL

Trabalho de Conclusão de Curso apresentado à
Coordenação do curso de Licenciatura Plena
em Computação da Universidade Estadual da
Paraíba – UEPB, como requisito parcial para
obtenção do título de licenciado.

Orientadora: MSc^o Angélica Felix Medeiros

PATOS – PB
2016

R696a Rodrigues, Clecia Leite

Avaliação de Software Educacional com base nos conceitos do Design Emocional [manuscrito] / Clecia Leite Rodrigues. - 2016.

69 p.

Digitado.

Trabalho de Conclusão de Curso (Graduação em Computação) - Universidade Estadual da Paraíba, Centro de Ciências Exatas e Sociais Aplicadas, 2016.

"Orientação: Profa. Ma. Angélica Felix Medeiros, CCEA".

1. Software Educacional. 2. Design Emocional. 3. Avaliação de software. I. Título.

21. ed. CDD 005.3

Clecia Leite Rodrigues

**AVALIAÇÃO DE SOFTWARE EDUCACIONAL COM BASE NOS CONCEITOS DO
DESIGN EMOCIONAL**

Trabalho de Conclusão de Curso apresentado ao
Curso de Licenciatura em Computação da
Universidade Estadual da Paraíba, em
cumprimento à exigência para obtenção do grau
de Licenciado em Computação

Aprovado em 21 de outubro de 2016

BANCA EXAMINADORA

Angélica Felix Medeiros
Angélica Felix Medeiros
(Orientadora)

Nádia Farias dos Santos
Nádia Farias dos Santos
(Examinadora)

Alanna Camilla Coelho Monteiro
Alanna Camilla Coelho Monteiro
(Examinadora)

DEDICATÓRIA

Dedico este trabalho primeiramente a Deus, por ter me dado força e sabedoria para concluir este trabalho, se não fosse ele eu não teria conseguido.

Aos meus pais por ter sido essenciais na minha formação, me proporcionando amor, carinho e dedicação. Dedico este trabalho também a minha Avó Joana Leite (*in memoriam*), pelo amor, carinho e por seus valiosos conselhos.

AGRADECIMENTOS

A Deus, em primeiro lugar, por ter proporcionado essa vitória, me dando força, coragem, saúde e muita dedicação.

A esta Universidade, pela oportunidade de formação.

A minha orientadora, Angélica Félix, que sempre esteve presente em todo meu trabalho, por ter aceitado me orientar e pela dedicação que sempre teve. Agradeço também por sempre está disponível para tirar as dúvidas que iam aparecendo no decorrer do trabalho.

Aos meus professores, por ter transmito conhecimento e me proporcionado aprendizagem.

Aos meus pais, Josefa Leite e José Rodrigues, por todos ensinamentos, pela educação que me ofereceram, por ter sempre acreditado no meu potencial, por sempre está do meu lado. Agradeço também pelo orgulho que ele de mim.

A minha Avó Joana Leite (*in memorian*), por ter sido uma segunda mãe e por ter contribuído para que este sonho se tornasse realidade. Sei que ela torce por mim onde quer que esteja.

Ao meu namorado, Mateus Alves, por compreender meus momentos de ausência, agradeço a ele por sempre me apoiar e me fazer acreditar que eu sempre fui capaz.

Ao meu irmão João Rodrigues e minha cunhada Najara Alves, pelo amor e a paciência que sempre tiveram comigo.

Aos meus familiares, que sempre me apoiaram e entenderam que eu tive que me ausentar em algumas datas importantes durante todo o curso e mais ainda nessa reta final.

Aos meus colegas de Sala, pelo carinho, companheirismo e amizade, agradeço também aos colegas de outros períodos, principalmente a Jaziel Moreira pelas vezes que me ajudou.

Agradeço principalmente a Verlândia Rodrigues, minha amiga de sala e companheira de todos os trabalhos, ela me ajudou em todos os momentos dos cursos, sempre dando conselhos e fazendo-me acreditar que sou capaz. Agradeço também a minha amiga de sala que se tornou amiga de trabalho, Rafaela Marinho, pelo carinho e amizade.

Aos Alunos e professores do Coriolano de Medeiros por ter me recebido muito bem e por toda atenção.

Agradeço a todos, que de uma forma ou de outra contribuíram para a realização deste trabalho.

“(...) utilidade e usabilidade são importantes, mas sem diversão, prazer, alegria e entusiasmo, e até ansiedade e raiva, medo e fúria, nossas vidas seriam incompletas.” NORMAN, 2008,p.28.

RESUMO

O Design Emocional é uma área que vem sendo estudada pelos designers e que deve ser levada em consideração na hora de desenvolver um produto, inclusive produtos educacionais. Antes de incluir qualquer *software* no ambiente escolar se faz necessário uma avaliação, pois da mesma forma que um *software* pode contribuir positivamente ele também pode trazer efeitos negativos, para isso existem várias técnicas e métodos utilizados na avaliação de *softwares* educacionais. É interessante que a avaliação consiga ir além de aspectos funcionais para ser possível avaliar diferentes nuances, como, por exemplo, os sentimentos dos usuários ao utilizar determinado produto. Sendo assim, esse trabalho tem como objetivo a avaliação de um *software* educacional levando em consideração alguns conceitos do Design Emocional. A pesquisa apresenta caráter exploratório e foi realizada a partir de um estudo de caso com a turma do projeto de extensão promovido pela Universidade Estadual da Paraíba (UEPB) que tinha como objetivo ensinar programação e jogos aos alunos do ensino fundamental das escolas públicas. Os dados dessa pesquisa foram obtidos por meio de questionário aplicado aos alunos que participaram do projeto e a abordagem adotada para a análise dos resultados é quantitativa e qualitativa. Como resultado, percebeu-se que a maioria dos sentimentos despertados pelo uso do *software* foram positivos, quando o produto desperta sentimentos positivos nos usuários eles conseguem desenvolver habilidades e a criatividade. O uso dos *emoticons* para expressar os sentimentos das crianças ao utilizar o *software* teve resultados interessantes, uma vez que facilitou a expressão dos sentimentos além de aproximá-las da realidade em que vivem.

Palavras-chave: *Software* Educacional. Design Emocional. Avaliação de *Software*.

ABSTRACT

The Emotional Design is an area that has been studied by the designers and that should be taken into account when developing a product, including educational products. Before you add any software in the school environment if an assessment is necessary, because in the same way that software can contribute positively it can also have negative effects, for this there are various techniques and methods used in the evaluation of educational software. Interestingly, the evaluation can go beyond functional aspects to be able to evaluate different shades, such as, for example, feelings of users using particular product. Thus, this study aims the evaluation of an educational software taking into consideration some concepts of Emotional Design. The research presents exploratory and was carried out from a case study of the extension project of the class sponsored by the Universidade Estadual da Paraíba (UEPB) which aimed to teach programming and games for elementary school students from public schools. The data from this study were obtained through a questionnaire applied to the students who participated in the project and the approach to the analysis of the results is quantitative and qualitative. As a result, we realized that most of the feelings aroused by the software use were positive when the product arouses positive feelings in users they can develop skills and creativity. The use of *emoticons* to express the feelings of children when using the software had interesting results since it facilitated the expression of feelings as well as bringing them closer to the reality they live.

Keywords: Educational Software. Emotional Design. Software Evaluation.

LISTA DE FIGURAS

| | |
|--|----|
| Figura 1 - Interface do LOGO (KTurtle)..... | 21 |
| Figura 2 - Interface do Scratch..... | 22 |
| Figura 3 - Interface do Alice..... | 23 |
| Figura 4 – Tela principal do KTurtle..... | 24 |
| Figura 5 - Interface do Code Monster..... | 25 |
| Figura 6- Interface do Squeak eToys..... | 26 |
| Figura 7 - Interface do FazGame..... | 27 |
| Figura 8 - Tela inicial do Kodu Game Lab..... | 28 |
| Figura 9 - Modelo de qualidade interna e externo segundo a ISO/IEC 9126-1..... | 30 |
| Figura 10 – Os níveis cerebrais do design emocional..... | 33 |
| Figura 11 - <i>Emoticons</i> utilizados no questionário..... | 38 |
| Figura 12 – Ícones instalação Kodu..... | 39 |
| Figura 13 – Tela inicial Kodu..... | 40 |
| Figura 14 – Cenário vazio..... | 41 |
| Figura 15 – Menu..... | 42 |
| Figura 16 – Ferramenta de objetos..... | 42 |
| Figura 17 – Modificação de objetos..... | 43 |
| Figura 18 – Programação do objeto..... | 43 |
| Figura 19 – Adicionar opção “Quando”..... | 44 |
| Figura 20 – Adicionar opção “Faz”..... | 44 |
| Figura 21 – Excluir expressão..... | 45 |
| Figura 22 – Salvar mundo..... | 45 |
| Figura 23 – Alterar definições de objeto..... | 46 |
| Figura 24 – Resposta sinalizada do questionário..... | 48 |

LISTA DE QUADROS

| | |
|---|-----------|
| Quadro 1 – Comparativo Entre as Principais Ferramentas | 29 |
| Quadro 2 – Questões exploradas no questionário da pesquisa | 37 |
| Quadro 3 – <i>Emoticons</i> para representar as respostas possíveis da questão 05..... | 50 |
| Quadro 4 – <i>Emoticons</i> para representar as respostas possíveis da questão 06..... | 51 |
| Quadro 5 – <i>Emoticons</i> para representar as respostas possíveis da questão 07..... | 52 |
| Quadro 6 – <i>Emoticons</i> para representar as respostas possíveis da questão 08..... | 53 |
| Quadro 7 – <i>Emoticons</i> para apontar a experiência de utilizar o jogo | 54 |
| Quadro 8 – Respostas possíveis para a questão 10..... | 55 |
| Quadro 9 – <i>Emoticons</i> para representar as respostas possíveis da questão 11..... | 56 |

LISTA DE GRÁFICOS

| | |
|--|----|
| Gráfico 1 – Quais os tipos de jogos que você mais utiliza..... | 49 |
| Gráfico 2 – O Kodu Game Lab é fácil de mexer..... | 51 |
| Gráfico 3 – Você demorou aprender a utilizar o Kodu Game Lab | 52 |
| Gráfico 4 – Você conseguiu criar o jogo queria..... | 53 |
| Gráfico 5 – O que você sentiu quando conseguiu criar o jogo que desejava | 54 |
| Gráfico 6 – Como você aponta a experiência de utilizar o jogo | 55 |
| Gráfico 7 – Teve funcionalidades que você não conseguiu utilizar..... | 56 |
| Gráfico 8 – Aponte os <i>emoticons</i> que estão relacionados às emoções/sentimentos ao utilizar os recursos de navegação (menus, ícones, links e botões)..... | 57 |

SUMÁRIO

| | |
|--|-----------|
| 1 INTRODUÇÃO | 13 |
| 1.1 OBJETIVOS | 16 |
| 1.1.1 Objetivo geral | 16 |
| 1.1.2 Objetivos específicos | 16 |
| 1.2 ORGANIZAÇÃO DO TRABALHO | 16 |
| 2.1 ENSINO DE PROGRAMAÇÃO PARA CRIANÇAS | 17 |
| 2.1.1 Linguagens de Programação para Educação | 19 |
| 2.1.1.1 LOGO | 20 |
| 2.1.1.2 Scratch | 21 |
| 2.1.2 Ferramentas de programação para crianças | 22 |
| 2.1.2.1 Alice | 23 |
| 2.1.2.2 Kturtle | 24 |
| 2.1.2.3 Code Monster | 25 |
| 2.1.2.4 Squeak eToys | 25 |
| 2.1.2.5 FazGame | 27 |
| 2.1.2.6 Kodu Game Lab | 28 |
| 2.2 AVALIAÇÃO DE <i>SOFTWARE</i> | 30 |
| 2.2.1 Avaliação de <i>software</i> através do design emocional | 32 |
| 2.2.3 Avaliação de <i>software</i> educacional | 34 |
| 3 METODOLOGIA..... | 36 |
| 3.1 DESCRIÇÃO DO <i>SOFTWARE</i> EDUCACIONAL KODU GAME LAB | 38 |
| 3.1.1. Instalação | 39 |
| 3.1.2. Funcionalidades | 39 |
| 3.1.3 Criação de Jogo | 40 |
| 3.2 ESTUDO DE CASO | 46 |
| 4 RESULTADOS E DISCUSSÕES | 48 |
| 4.1 Análise dos Resultados | 48 |
| 5 CONSIDERAÇÕES FINAIS..... | 59 |
| REFERÊNCIAS..... | 61 |
| APENDICE A | 66 |

1 INTRODUÇÃO

As tecnologias estão presentes em todos os meios, seja no ambiente educacional ou não, com este avanço na utilização das tecnologias, o modo como às pessoas vivem na sociedade foi se modificando. De acordo com Kensky (2006), as tecnologias interferem em nosso modo de pensar, agir, de relacionar-se socialmente e alcançar conhecimentos.

A tecnologia está a favor da sociedade, mas o que se percebe é ao mau uso da mesma. Um exemplo disso é a inserção da tecnologia em sala de aula sem planejar os objetivos que devem ser alcançados e os métodos utilizados para tal.

A programação é uma nova forma de fazer com que as crianças aprendam a fazer com o computador o que elas querem de forma divertida e atraente. Diferentemente de outras disciplinas, a programação é uma ferramenta que as crianças conseguem usar e ver algo pronto, podendo assim mostrar a outras pessoas o que foi feito. Segundo Deters et al. (2008), disciplinas que envolvem a programação e o ensino de algoritmos são consideradas desafiadoras pelos alunos, já que elas requerem o desenvolvimento de estratégias de resolução de problemas com base lógico matemática.

O ensino de programação para crianças vem sendo algo bastante discutido entre profissionais da educação e profissionais da área de computação. O principal ponto de debate é esclarecer se programar ajuda ou não no desenvolvimento escolar da criança, ou ainda se existe a real necessidade deste ensino.

Segundo Rushkoff (2012) aprender a programar hoje em dia é tão importante quanto foi aprender a ler e a escrever no século XX. Da mesma forma que existem especialistas que são a favor do ensino de programação para crianças existem também outros especialistas que são contra como, por exemplo, Linus Torvalds, o criador do sistema operacional Linux, Segundo Love (2014) a programação não pode ser comparada com aprender a ler e escrever ou fazer conceitos básicos de matemática, ela é algo mais complexo, não pode achar que a maioria das pessoas irá programar. Para Setzr (2002), a introdução precoce de programação prejudica a infância e a juventude, e que ainda podem causar desastres mentais.

Neste contexto destaca-se que a educação é à base de qualquer sociedade, ela deve ser adequada ao seu tempo. Segundo a organização Code Club Brasil¹ (2015) “aprender a programar não é útil apenas se você quiser ser um programador no futuro. Programar ajuda em outras habilidades, como resolver problemas, desenvolve o raciocínio lógico e contribui

¹ Code Club Brasil: Disponível: <http://codeclubbrasil.org/oquee/>

em outras matérias como ciências e matemática”. Ensinar programação as crianças, estará aumentando a qualidade da educação e assim fazendo com que elas aprimorem outras áreas.

Existem pesquisas que apontam que alunos que aprenderam a programar tiveram suas notas crescentes. Segundo Barros (2013), um projeto realizado pelo núcleo Universidade Estadual Paulista mostrou que o uso da tecnologia na educação melhora em 32% o rendimento dos alunos em matemática e física.

Segundo Pausch e Kelleher (2005), desde o início dos anos 1960, pesquisadores tentam desenvolver linguagens de programação e ambientes com a intenção de tornar a programação acessível para um número maior de pessoas. Neste sentido, existem várias ferramentas de programação que foram desenvolvidas para auxiliar o ensino da programação para crianças. Grande parte das abordagens apresenta uma linguagem de programação visual e tem por objetivo apoiar os seus usuários no aprendizado de conceitos de programação mais divertida e estimulante (Cooper, Danne Pausch 2000; Maloney et al.2010). Pois, quando a aprendizagem é prazerosa e motivadora o aluno aprende com mais facilidade.

Existem vários ambientes de ensino de programação para crianças, entre eles o Scratch², (Mit Media Lab, 2012), Kodu Game Lab³ (Microsoft Research, 2014), Alice⁴ (Carnegie Mellon Universit, 2013), FazGame⁵ (2015), KTurtle⁶(2011).

O Scratch é uma das linguagens de programação mais conhecidas, ela é uma linguagem simples que pode ser usada por qualquer pessoa sem precisar conhecer outras linguagens de programação, ele serve principalmente para iniciantes em programação. Existem também outras linguagens de programação que são simples de utilizar em sala e que também servem como alternativa no processo de ensino e aprendizagem de programação. Como por exemplo, o Kodu Game Lab – que permite que os usuários sem nenhum conhecimento em programação consiga desenvolver seus próprios jogos.

Outro exemplo é o KTurtle, um ambiente educativo de programação que diferente do Scratch, utiliza uma linguagem de programação. Ele é um *software* que além de ensinar programação ensina também matemática e geometria, fazendo a tradução de comandos no idioma do programador. O Alice também é um *software* educativo de programação, ele faz uso da realidade aumentada para ensinar programação orientada a objetos. Para fazer uso desse ambiente não precisa ter experiência em programação.

² Scratch: Disponível: <https://scratch.mit.edu/about/>

³ Kodu Game Lab: Disponível: <https://www.kodugamelab.com/>

⁴ Alice: Disponível: <http://www.alice.org/index.php>

⁵ FazGame: Disponível: <https://www.fazgame.com.br/>

⁶ KTurtle: Disponível: <http://wiki.ubuntu-br.org/KTurtle?action=info>

A aplicação de um *software* no ambiente escolar é bastante complexa e envolve diversos fatores e estratégias, segundo Meira (1998) mais importante que o software, em si, é o modo como ele será utilizado, pois nenhum software é, em termos absolutos, um bom software. Isso se evidencia ainda mais quando a área de aplicação é estranha aos alunos e professores, como acontece com o ensino de programação através de ferramentas e *softwares*. Nesse contexto, uma importante etapa dessa aplicação está relacionada à avaliação do *software* e para tanto, existem várias abordagens e técnicas de avaliação voltadas para eficácia e eficiência.

Dentre estas, a usabilidade de *software* já foi bastante discutida ao longo dos anos, a norma ISO 9241-11 (1998) define usabilidade como a “capacidade de um produto ser usado por usuários específicos para atingir objetivos específicos com eficácia, eficiência e satisfação em um contexto específico de uso”.

Outra vertente da avaliação de *software* que vem sendo estudada é o Design Emocional (DE), pois “Não reagimos às qualidades físicas das coisas, mas ao que elas significam para nós.” (KRIPPENDORFF, 2001 p.89 apud MONT’ALVÃO & DAMÁZIO, 2008, p. 8). Com isso o design emocional tenta extrair para a avaliação os sentimentos despertados ao utilizar determinado produto e avaliar essa interação a partir disto. “A maior parte dos objetos que nos rodeiam acaba por ser capa de desencadear emoções, fortes ou fracas, boas ou más, conscientemente ou não” (DAMÁSIO, 2004 p.62/63). Ou seja, tornar o produto amigável, agradável e que chame a atenção do usuário. Quando o usuário se sente atraído pelo *software* ele consegue interagir melhor com o produto.

Com o estudo do emocional é possível entender as respostas dos usuários ao utilizar determinada ferramenta bem como sua reação diante do contato com a ferramenta. Um dos principais objetivos do design emocional é desenvolver produtos que não causem desconfortos no usuário e que ele não fique aflito ao utilizar, já que isso pode implicar tanto na qualidade do produto como no aprendizado do usuário.

Diante deste contexto, o presente trabalho pretende fazer um estudo sobre o design emocional, bem como avaliar um *software* de programação levando em consideração o emocional do usuário. Para tanto os objetivos da pesquisa foram traçados e estão explanados a seguir.

1.1 OBJETIVOS

1.1.1 Objetivo geral

O objetivo geral desse trabalho é avaliar um *software* educacional voltado para o ensino de programação para crianças com base nos conceitos do Design Emocional.

1.1.2 Objetivos específicos

- Realizar o levantamento bibliográfico sobre os tipos de linguagens e *softwares* de programação para crianças e os tipos de avaliações de *softwares* educacionais;
- Identificar os principais/mais utilizados *softwares* educacionais que objetivam o ensino de programação para crianças;
- Descrever o *software* educacional que será avaliado;
- Definir os critérios de avaliação que serão utilizados para analisar o *software*;
- Discutir os resultados da avaliação analisando o impacto do design emocional na utilização do *software*.

1.2 ORGANIZAÇÃO DO TRABALHO

Este trabalho está dividido em cinco capítulos, o capítulo 1 refere-se a esta parte introdutória do trabalho, onde delimita o tema a ser discutido no trabalho, expõem os objetivos e apresenta à justificativa.

O capítulo 2 apresenta o embasamento teórico, contendo todo o estudo acerca do tema, serão mostrados alguns tipos de linguagens e ferramentas para ensino de programação para as crianças e alguns tipos de avaliações dando ênfase ao design emocional.

O capítulo 3 apresenta a metodologia do trabalho, a descrição do *software* bem como a sua funcionalidade e o modelo de critérios usados na sua avaliação.

No capítulo 4 são apresentados os resultados alcançados e a análise dos resultados.

Por último, o capítulo 5, onde serão apresentadas as conclusões do trabalho, as contribuições, limitações a respeito do tema e sugestões para trabalhos futuros.

2 ENSINO DE PROGRAMAÇÃO PARA CRIANÇAS E AVALIAÇÃO DE *SOFTWARES*

No referencial teórico serão apresentados alguns temas relacionados à área de pesquisa deste trabalho para alcançar os objetivos propostos nesta pesquisa. Levando em consideração alguns temas como, o ensino de programação para crianças, avaliação de *software* educacional e design emocional.

2.1 ENSINO DE PROGRAMAÇÃO PARA CRIANÇAS

A programação está presente em diversas atividades do dia a dia, até mesmo em uma simples porta. O tema programação para crianças tem sido discutido cada vez mais nos debates e nos programas de televisão. Entender a programação e aprender a programar ajuda a resolver diversos problemas que aparecem em algumas situações do dia a dia de forma estruturada e organizada.

Vários nomes da tecnologia tem defendido a ideia de ensinar programação para crianças, Bill Gates e Mark Zuckerberg defendem a ideia de introduzir a programação no ensino fundamental, para eles e outros estudiosos todos devem ter pelos menos algumas noções básicas de como programar. Em um discurso Barack Obama, presidente dos Estados Unidos, chama atenção ao falar de programação “Não compre, apenas, um videogame, desenvolva um. Não baixe um aplicativo, ajude a projetar um. Não jogue no seu celular, programe-o.” (OBAMA, 2013).

Aprender programação não é uma tarefa fácil e quando se refere ao ensino de programação para crianças os desafios e obstáculos são muitos. Precisa-se ter um cuidado especial ao se tratar desse público.

A ideia de inserir a programação nas escolas surgiu nos Estados Unidos, e hoje já existem algumas escolas no Brasil, principalmente as particulares, que já inseriram a programação como uma disciplina da grade curricular.

É muito importante para o desenvolvimento das crianças a criação de projetos que incentive o ensino de programação. Existem alguns projetos no Brasil e fora do Brasil que tem por objetivo o ensino de programação para crianças, entre eles está o Code Club, ele foi criado pela designer Clare Sutcliffe e a desenvolvedora Linda Sandvik. “A filosofia do CODEClub é diversão, criatividade e aprendizagem pela descoberta. È importante que as

crianças curtam o tempo que elas passam no CODEClub e que não de sintam como se estivessem em mais uma aula da escola.” (Code Club Brasil, 2015). Existem outras organizações similares que incentivam ensino da programação, como o Code.org que foi criado pelo Iraniano Hadi Partovi, formado em ciência da computação pela universidade de Havard e investidor de empresas como o Facebook⁷ e o Dropbox⁸. O Code.org⁹ visa ensinar qualquer pessoa e principalmente as crianças a programar, ele possui um site onde os interessados podem aprender os primeiros passos para a programação de computadores com a ajuda de tutoriais.

O Google¹⁰ lançou dia 27 de junho desse ano o Project Blocks¹¹, uma plataforma que tem como objetivo ensinar programação para crianças, a plataforma é resultado de uma pesquisa colaborativa entre o Google, IDEO¹² e o professor Brasileiro Paulo Blikstein.

Falar em ensino de programação a crianças é um assunto de grande importância, só que para essa inserção acontecer de forma eficiente será preciso um planejamento. Já que a programação não é uma tarefa fácil e requer várias habilidades. Vários estudiosos no assunto não defendem a ideia de que as crianças devem aprender a programar, muitos acreditam que a programação é uma tarefa complexa que deixam de lado uma fase importante na vida das pessoas que a fase que a criança brinca com brinquedos que não envolvem a tecnologia. Segundo Setzer “Ao usar o computador, a criança é obrigada a exercer um tipo de pensamento que deveria empregar somente em idade mais avançada. Com isso podemos dizer que os computadores roubam das crianças sua necessária infantilidade. Elas são obrigadas a pensar e usar uma linguagem que deveria ser dominada apenas por adultos” (Setzer, 2002).

Saeli (2011) apresenta cinco dificuldades que pode provocar distúrbios ao ensinar e aprender programação:

- Entender as vantagens de utilizar a programação e os benefícios que ela traz;
- Compreender o funcionamento da máquina;
- Aprender uma nova notação, que pode provocar problemas relacionados aos aspectos de sintaxe e semântica das linguagens de programação.
- Entender o uso correto das estruturas de programação;

⁷ Facebook: uma das maiores redes sociais

⁸⁸ Dropbox: serviço de armazenamento em nuvem

⁹ Code.org: disponível: <https://code.org/>

¹⁰ Google: é uma empresa multinacional americana de serviços online e *software*

¹¹ Project Blocks: disponível: <https://projectbloks.withgoogle.com/>

¹² IDEO: é uma empresa internacional de design e consultoria em inovação,

- Attingir proficiência nas habilidades necessárias à implementação;

O ensino de programação é visto como algo bem complexo, segundo alguns autores esse nível de dificuldade pode causar vários problemas relacionados à aprendizagem. Sales e Dantas (2010) citam os principais problemas relacionados ao ensino e aprendizagem de programação:

1. Alto grau de abstração;
2. Metodologias tradicionais de ensino que visam à aprendizagem de conceitos dinâmicos utilizando abordagens e materiais estáticos;
3. Existência de alunos com backgrounds diversificados, levando a ritmos de aprendizagem diferenciados;
4. Dificuldade de acompanhamento individualizado do aluno;
5. Necessidade de um estudo baseado na prática;
6. Linguagens de programação com sintaxes complexas e carência de representações visuais de algoritmos dificultando a adequada compreensão dos conceitos de programação envolvidos.

Quando se pensa em criança e ensino a ideia é tentar ensinar de uma forma que as crianças gostem de fazer e tenham prazer em fazer determinada atividade e assim se sintam motivadas, as crianças que nasceram na era digital ditas como crianças digitais estão muito focadas nas tecnologias, elas têm grande facilidade de lidar com as ferramentas computacionais. Não existe outra maneira melhor da criança aprender do que o ato de brincar, ensinar a criança a aprender brincando é algo que as fazem interagir melhor.

Com a ideia de ensinar programação para crianças existem uma gama de ferramentas e linguagens de programação que servem como auxílio de ensino fazendo com que as crianças se sintam motivadas e tenha uma aprendizagem divertida e prazerosa. Tais ferramentas e linguagens estarão explanadas nas seções abaixo.

2.1.1 Linguagens de Programação para Educação

Existem várias linguagens de programação, elas variam das mais simples as mais complexas. As linguagens de programação para crianças devem ser escolhidas com mais atenção. Segundo Papert as linguagens de programação devem apresentar três critérios, o

primeiro (Low floor) é que as linguagens de programação devem ser fáceis de começar a utilizar, o segundo (High ceiling) é que eles devem dar a oportunidades de criar atividades mais complexas ao decorrer do tempo, e o último (Wide walls), não menos importante, elas devem suportar diferentes projetos para que os diversos tipos de projetos sejam aceitos independente da complexidade.

Encontram-se várias Linguagens de programação para educação, com ambientes de fácil entendimento, como é o caso do Logo (MIT, 1967), Scratch (Mit Media Lab, 2012), dentre outras.

2.1.1.1 LOGO

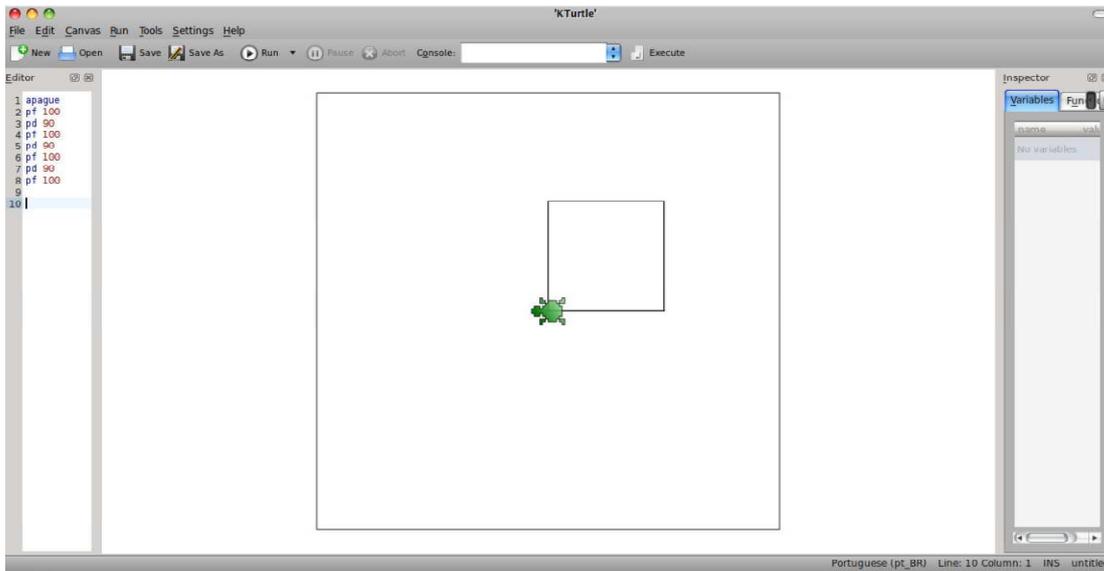
O Logo é uma linguagem de programação desenvolvida pelo matemático e educador Seymour Papert na década de 60, a linguagem logo é baseada na teoria de Jean Piaget e é totalmente voltada para a área da educação, ela é indicada para as crianças por ser uma linguagem simples e de fácil entendimento. O objetivo de Papert ao desenvolver a Logo foi dá oportunidades para que as crianças aprendam com prazer a programar para assim potencializar a aprendizagem (PAPERT, 1997; SOUZA, 2007).

Sua proposta é colocar a criança para controlar uma tartaruga robótica e ao mesmo tempo em que esse controle está acontecendo será possível ver essa representação na tela do computador.

As crianças dão os comandos e a tartaruga é controlada de acordo com os comandos inseridos pela criança. Por meio dessa linguagem as crianças conseguem interagir com o computador para realizar várias ações e com isso podem fazer vários desenhos.

Logo é a linguagem que abriu espaço para as pessoas que não tinham acesso a programação, incluindo as crianças. Ela é uma linguagem de fácil interação, que possui vários comandos básicos, como por exemplo, andar para frente e andar para trás. Na figura 1 a tela do Logo está representada, mostrando os passos que a tartaruga dá a partir dos comandos.

Figura 1: Interface LOGO (KTurtle)



Fonte: <<https://bustillo.files.wordpress.com/2012/11/KTurtle.pdf>> Acesso em abr. 2016

A ideia do Logo é que a criança aprenda programação de uma maneira divertida, onde ao mesmo tempo em que se diverte controlando a tartaruga na tela do computador, ela consegue aprender programação e ainda desenvolver conhecimentos de outras áreas.

Segundo Papert a linguagem Logo é “um instrumento capaz de mudar a escola como um todo, para mudar atitudes sobre o modo como se fala sobre a aprendizagem e mesmo em relação às pessoas” (PAPERT, 1988, p. 27).

2.1.1.2 Scratch

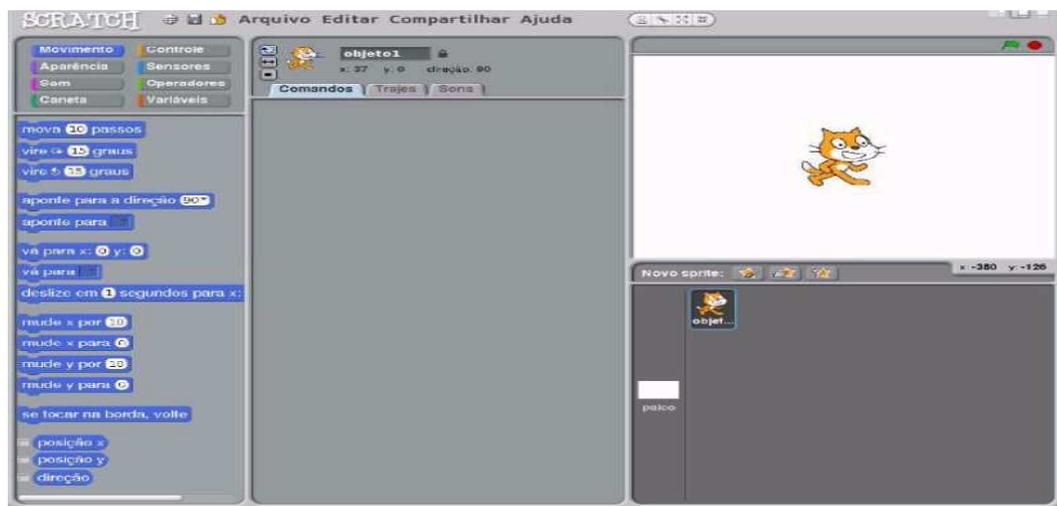
Scratch é uma linguagem e um ambiente de programação gráfica desenvolvida pelo Massachusetts Institute Technology (MIT), ela permite criar animações, fazer histórias animadas e interativas, simulações ou jogos e fazer compartilhamento dessas criações na web.

Ela foi desenvolvida para estimular o desenvolvimento de lógica e programação com as crianças. Segundo Trajano (2010) o Scratch é uma linguagem que serve como base para ajudar os usuários entender e utilizar outras linguagens e ferramentas de programação que sejam mais complexas.

“Embora o Scratch tenha sido projetado principalmente para crianças de 8 a 16 anos, ele é usado por pessoas de todas as idades, inclusive crianças mais jovens, com a ajuda de seus pais.” (SCRATCH, 2015).

O Scratch possui vários elementos e dentro desses elementos contém blocos de comandos, ao abrir a ferramenta, o próprio Scratch cria o primeiro objeto, um gato, ele pode ser movimentado na tela a partir de alguns comandos desta linguagem. Como por exemplo, o comando “mova 10 passos”, para executar este comando é só clicar duas vezes com o botão esquerdo do mouse. O ideal nessa criação são os trajés, quanto mais traje mais bonito fica a animação. A ferramenta consiste em apenas mover blocos, sem precisar digitar os comandos. A figura 2 mostra a interface do Scratch com elementos e ações em blocos.

Figura 2: Interface do Scratch



Fonte: <<https://nuestratecnologia.files.wordpress.com/2011/09/manual-scratch.pdf>> Acesso em abr. 2016.

O Scratch utiliza uma linguagem visual que é simples e intuitiva baseada em blocos de montar que pode ser trabalhada com pessoas de todas as idades, ela desenvolve o raciocínio lógico podendo assim ajudar as crianças no processo de aprendizagem de outras atividades.

O Scratch ajuda os jovens a aprenderem de maneira criativa, refletir de maneira sistemática e trabalhar de forma colaborativa e habilidades essenciais para a vida no século 21 (Scratch, 2014).

2.1.2 Ferramentas de programação para crianças

Além das linguagens de programação, muitas ferramentas são utilizadas no contexto de ensinar programação, pois facilitam o ensino de programação para crianças. Nesta seção algumas ferramentas mais utilizadas serão explanadas

2.1.2.1 Alice

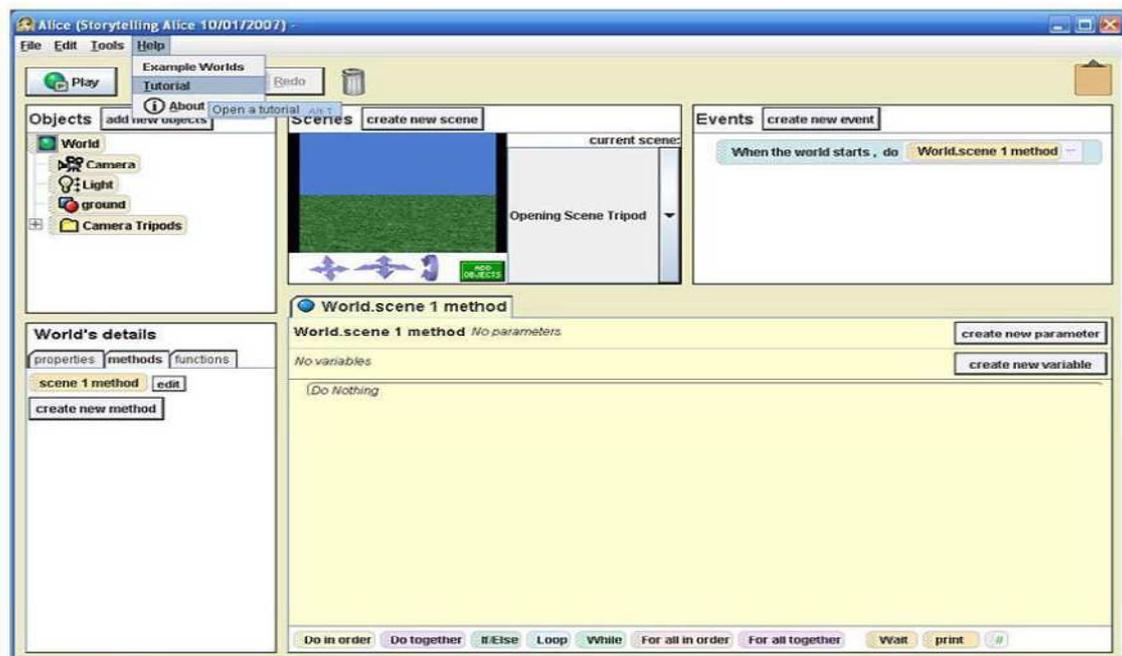
Alice é um ambiente de programação 3D, criado nos Estados Unidos por pesquisadores da Universidade Norte Americana Carnegie Mellon . O Alice torna mais fácil a criação de uma animação para contar uma história, jogar um jogo interativo, ou criar um vídeo e depois compartilhar na web.

A linguagem do Alice é visual, ou seja, não necessita digitar os códigos, a edição ocorre por meio da técnica de arraste e solte, esse tipo de linguagem elimina as chances de erros de sintaxe, já que esses erros podem desmotivar os usuários iniciantes.

Alice é uma ferramenta livre e gratuita que ensina programação orientada a objetos usando a realidade virtual. Dentre os seus objetos, este ambiente de programação possui um diferencial, uma câmera, que permite exibir o mundo virtual na tela do computador, oportunizando visualizar o que está acontecendo no mundo virtual quando o programa está em execução (SHELLY et al., 2007).

Ela possui uma interface (Figura 3) interativa, que corresponde a outras linguagens de programação como, por exemplo, Java, C++ Builder, Delphi.

Figura 3: Interface do Alice



Fonte: <<http://www.alice.org/index.php>> Acesso em abr. 2016.

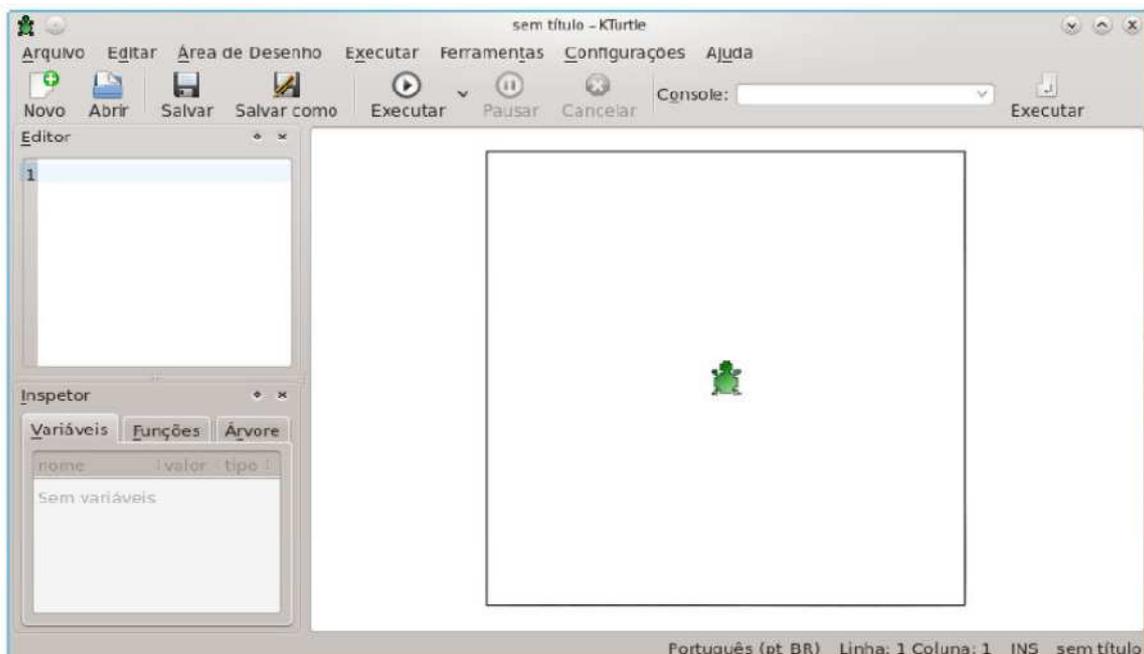
Essa ferramenta foi desenvolvida para facilitar o teste de novas tecnologias e tornou-se uma ferramenta didática em cursos de nível superior da área de computação e até mesmo para o ensino médio (ZOTOVICI & MENEZES, 2010).

2.1.2.2 KTurtle

O KTurtle é um ambiente de programação que usa a linguagem TurtleScrip, uma linguagem baseada no Logo. O KTurtle permite que o usuário programe uma tartaruga, com os diversos comandos, (para frente, para trás, use lápis, usinada, para direita, para esquerda), que o KTurtle oferece será possível criar simulações, apresentações, gráficos, textos e jogos.

De acordo com o manual, o KTurtle tem como objetivo tornar a programação fácil e acessível quanto possível, sendo assim uma linguagem simples que se torna adequada para ensinar algumas bases de programação e geometria para as crianças de forma divertida. A figura 4 abaixo mostra a tela principal do KTurtle.

Figura 4: Tela principal do KTurtle



Fonte : <<https://bustillolog.files.wordpress.com/2012/11/KTurtle.pdf>> Acesso em abr. 2016

Diferente de outras linguagens estudadas nesse trabalho, o KTurtle é um ambiente de programação que faz uso de linguagem textual, esse tipo de linguagem aproxima mais o usuário da realidade do programador, já que ele precisa digitar os códigos.

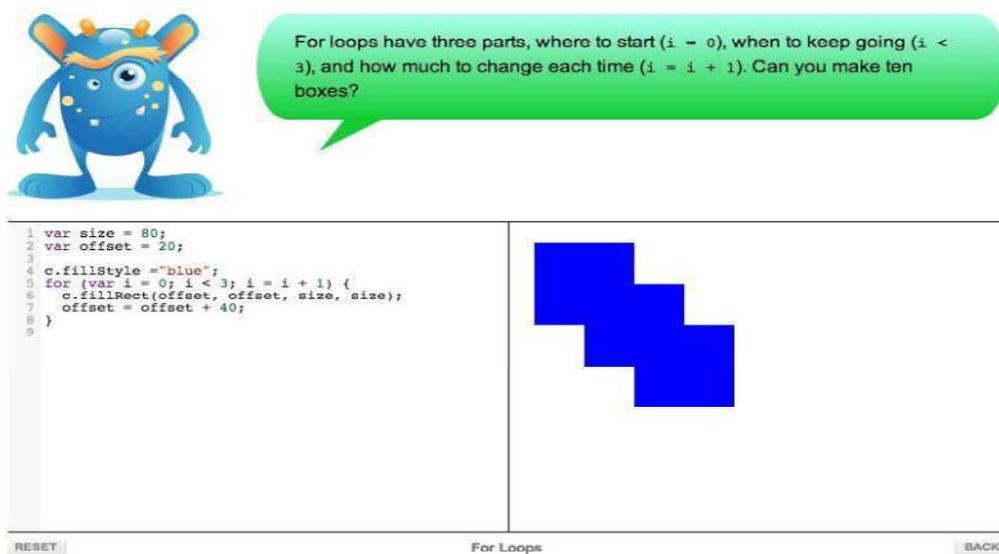
Torna-se evidente que o *software* Kturtle é uma ótima forma de ensinar programação para as crianças, Segundo Papert “a tartaruga liga a ideia de ângulo à navegação, atividade positiva e firmemente enraizada extracurricular de muitas crianças” (Papert, 1985, p. 87). O Kturtle é usado em muitas escolas públicas e já vem pré-instalado no sistema operacional Linux.

2.1.2.3 Code Monster

O Code Monster foi desenvolvido pela Empresa Ventures Geeky Inc. situada em Washington nos Estados Unidos, assim como outras ferramentas apresentadas nesse capítulo também, O Code Monster usa a linguagem de programação orientada a objetos, essa ferramenta utiliza a linguagem Javascript e pode ser acessada online diretamente do navegador.

Ela é uma ferramenta indicada para ensinar programação para as crianças, por ser considerada uma ferramenta dinâmica e divertida e isso acontece através de animações.

Figura 05: Interface do Code Monster



Fonte: <http://www.crunchzilla.com/code-monster>

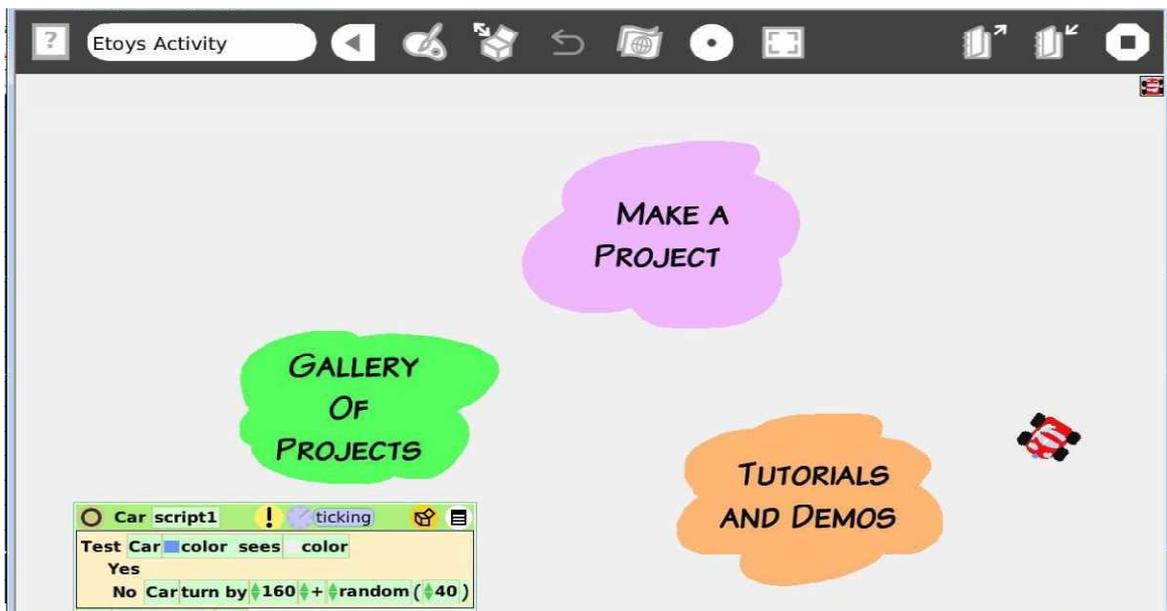
2.1.2.4 Squeak eToys

Squeak eToys é um ambiente de programação desenvolvido por Alan Kay e sua equipe, este ambiente usa recursos simples de programação permitindo que as crianças

programem suas próprias brincadeiras e consigam criar histórias interativas podendo assim ser compartilhadas.

A Linguagem de programação do Squeak eToys é uma linguagem orientada a objetos. Ele é um tipo de ferramenta que explora a criatividade da criança e desenvolve outras habilidades, já que as crianças têm a oportunidade de conhecer e usar a linguagem do computador podendo simular tanto ambientes em reais como virtuais. Na figura 6 está apresentada ilustrada a interface do Squeak eToys.

Figura 6: Interface do Squeak eToys



Fonte: <http://www.squeakland.org/tutorials/>

Fraga e Gewerc (2005) caracterizam o Squeak, definindo-o como:

- Ambiente pedagógico;
- Multimídia;
- Meio de comunicação;
- Ambiente de programação.

Para Allen-Conn & Rose (2003), o Squeak eToys, brinquedos digitais, oferece um leque sem fim de possibilidades e experimentação em todas as áreas de conhecimento, essencialmente visível na simulação e representação de modelos.

2.1.2.5 FazGame

O FazGame é um *software* educacional que foi desenvolvido no Brasil com o intuito de ensinar a criação de jogos para as crianças, ele pode ser utilizado por escola, alunos e usuários independentes.

O FazGame é uma ferramenta que não requer nenhum conhecimento de programação podendo dá oportunidade as crianças que nunca tiveram contato com linguagens de programação.

As crianças criam seus próprios jogos contando suas próprias histórias por meio da narração, como mostrado na figura 7. No *software* é possível às crianças escolherem seus próprios cenários, personagens e objetos, podendo assim fazer uso de vários cenários, objetos e personagens.

Com o uso desse *software* a criança está fazendo uso dos recursos computacionais, desenvolvendo seus próprios jogos e podendo ainda aprender conteúdo de qualquer disciplina.

Para desenvolver o jogo a criança deverá conhecer o assunto que se deseja narrar, é preciso que aja criatividade e uma reflexão do assunto, já que o jogo tem que ter um desenvolvimento lógico.

Figura 7: Interface do FazGame



Fonte: <<https://www.fazgame.com.br/>> Acesso em Maio de 2016.

Os jogos são criados e divulgados no portal do Fazgame permitindo que pais, professores e alunos tenham acesso aos jogos criados. Criar um game é mais divertido e produtivo do que usar um game feito, pois ao mesmo tempo em que a criança desenvolve um jogo ela desenvolve outras habilidades, como estimular a criatividade, o pensamento crítico e desenvolve o raciocínio lógico podendo assim ajudar em outras disciplinas.

2.1.2.6 Kodu Game Lab

Kodu é uma linguagem e um ambiente de programação visual desenvolvido pelo laboratório FUSE (Future Social Experiences) Labs, mantido pela Microsoft que serve, especialmente, para criar jogos do gosto do usuário, ela é uma linguagem destinada a todos em especial para as crianças, possui uma interface que usa uma linguagem simples, permitido assim que qualquer pessoa, mesmo sem conhecimento algum em programação, consiga criar seus jogos.

O Kodu possui uma série de ícones, onde a criança tem a possibilidade de criar mundos e todos os elementos que deseja e estabelecer as ações inerentes às personagens. Segundo o manual da Microsoft a lógica do Kodu é sempre uma, a “Quando-Faz” que significa que o usuário irá definir sempre a ação e a condição para que cada personagem possa executar o procedimento que é solicitado.

Na figura 8 abaixo será possível visualizar a tela inicial do Kodu Game Lab com todas as opções (Retornar, novo mundo, carregar mundo, comunidade, opções, ajuda, abandonar kodu) que o *Software* oferece.

Figura 8: Tela inicial do Kodu Game Lab



Fonte: Próprio autor

O Kodu é uma linguagem baseada totalmente em ícones, tornando-se assim uma aprendizagem direta e intuitiva. O ambiente do Kodu funciona no PC Windows e no Xbox.

Existem essas e outras ferramentas com o intuito de ensinar programação para crianças, elas diferem por algumas características, mas todas tem o intuito de ensinar de forma que a criança aprenda de uma maneira divertida e prazerosa, podendo desenvolver habilidades e tenha a oportunidade não só de usar o que já tem pronto, mas sim de produzir. Segundo Papert “as crianças programem a máquina em vez de serem programada por ela” (PAPERT, 2001, p. 39).

O Kodu vem sendo bastante utilizado em projetos de extensão, como por exemplo, na Universidade Estadual da Paraíba (UEPB), que utilizou o Kodu Game Lab no projeto que objetivava a inclusão digital no ensino fundamental.

No quadro 1 abaixo, segue o comparativo das principais ferramentas de programação para crianças apresentadas nesta seção.

Quadro 1: Comparativo entre as principais Ferramentas

| Ferramentas | Idioma Português | Gratuito | Interação Visual | Plataforma |
|---------------|------------------|----------|------------------|-------------------------|
| Alice | | X | X | Windows |
| KTurtle | | X | | Linux, Windows |
| Code Mosnter | | X | X | Web |
| Squeak eToys | X | X | X | Windows, MacOS, Linux |
| FazGame | X | X | | Windows, Linux, Mac OSX |
| Kodu Game Lab | X | X | X | Windows, XBox 360 |

Fonte: Próprio autor.

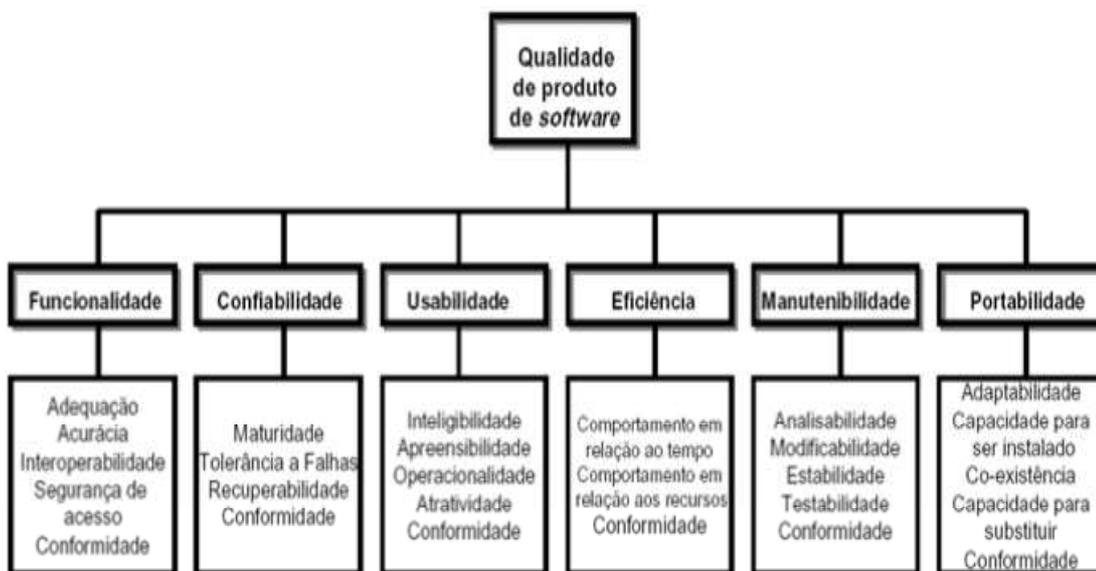
Neste sentido, este trabalho opta pelo Kodu Game Lab para avaliar. Como citado, a avaliação é um fator importante na utilização do *software*. Para tanto, os conceitos e estratégias em torno da avaliação serão explanados a seguir.

2.2 AVALIAÇÃO DE SOFTWARE

Avaliação de *software* é um assunto bastante discutido, mas que possui complexidade. Para um *software* ser considerado um *software* de qualidade ele precisa ser antes de tudo avaliado, essa avaliação vai identificar se esse *software* tem a qualidade para satisfazer as necessidades do usuário.

A ISO 9126 apresenta um modelo de qualidade interno e externo que contem seis características que são subdivididas em subcaracterísticas. Na figura 9 está representado o modelo de qualidade da ISO 9126

Figura 9: Modelo de qualidade interna e externo segundo a ISO/IEC 9126-1



Fonte: ISO 9126 (2009)

Ao se avaliar um *software* é importante observar como o *software* pode ser usado para determinados fins que se deseja alcançar com seu uso.

A cada dia que passa os *softwares* precisam oferecer maior qualidade para os clientes, as exigências por parte dos consumidores são muitas e só tende a aumentar. Essas exigências se devem ao fato da grande existência de *softwares* no mercado, ou seja, da grande concorrência. A cada dia que passa as tecnologias inovam mais e trazem características para o contexto social, e é preciso se adaptar para não ficar desatualizado.

Mais importante do que desenvolver um *software* é a sua avaliação, de nada adianta criar um *software* para determinada atividade se ele não atende as necessidades dos usuários

finais. É necessário pensar na avaliação antes do desenvolvimento, levando em consideração alguns critérios, como público alvo do *software* e a sua finalidade.

Avaliar um *software* não é uma tarefa simples, principalmente quando é destinado para crianças, é preciso levar em consideração vários critérios e normas. São vários os problemas encontrados nas avaliações de *softwares*, e é pensado nesses problemas que existe uma gama de métodos para a sua avaliação.

Existem vários tipos de avaliação, que vão da parte técnica do *software*, computacionais até os conteúdos abordados nele. Avaliar um *software* é tão importante quanto avaliar um livro didático. Ao escolher o *software* é preciso levar em consideração alguns critérios no que diz respeito à sua avaliação, e para isso existem vários critérios como: funcionalidade, compatibilidade, usabilidade, validade, eficiência, portabilidade, conformidade, público alvo, conteúdos abordados, faixa etária, interface agradável. Esses critérios envolvem tantos aspectos técnicos como pedagógicos que vão desde a criação até a aplicação aos usuários finais.

Os desenvolvedores dos *softwares* devem conhecer bem esses tipos de avaliação de qualidade de *softwares* e seus princípios. Um desses princípios é a usabilidade que segundo a ISO/IEC 9126, “Usabilidade refere-se à capacidade de uma aplicação ser compreendida, aprendida, utilizada e atrativa para o usuário, em condições específicas de utilização.”.

Usabilidade diz respeito a facilidade de usar determinado produto, se o produto não respeitar os critérios de uma boa usabilidade ele não será fácil de usar e conseqüentemente irá desestimular o usuário fazendo com que ele não tenha interesse em utilizar. Pois, de nada adianta ter um *software* educativo de excelente conteúdo, se sua utilização é tão complexa que desestimula o usuário.

A usabilidade deve ser uma preocupação constante em qualquer produto seja ele um *software* educacional ou não, se um *software* não for fácil de usar ele irá afetar a qualidade do produto e conseqüentemente a aprendizagem do usuário.

Encontram-se vários métodos de avaliação de qualidade de *software* educacional, entre eles a avaliação Heurística, Método de Reeves, a Técnica de Mucchielli e Ticese Revisão de Guidelines, Inspeção de Consistência, esses métodos são utilizados para avaliar diversos critérios.

Além dos critérios citados acima, existem outras questões a serem levadas em conta. Segundo Memória (2005), além de uma boa usabilidade existem outras questões que devem

ser pensadas na hora de avaliar um produto, como é o caso das emoções dos usuários. Com isso torna-se necessário estudar aspectos relacionados ao design emocional.

2.2.1 Avaliação de *software* através do design emocional

O Design Emocional é mais uma ferramenta que vem sendo estudado pelos designs de interface, para eles este estudo é tão importante quanto outros. Estudiosos da área percebem que a influência das emoções dos usuários é algo que pode interferir na qualidade do produto. Um dos objetivos do design emocional em produtos é que eles não causem efeitos indesejados e minimizem as frustrações nos usuários ao fazerem seu uso.

A emoção afeta os sentimentos e se esses sentimentos são positivos o usuário ficará mais motivado para utilizar o produto. Existem diversos autores que estudam o design emocional, os que estão em destaque são Jordan (1999), Desmet (2002) e Norman (2004). No entanto, Norman é considerado, por muitos autores, o pai do Design emocional.

Segundo Norman (2008), nós somos resultados de três níveis diferentes de estrutura cerebral:

Nível visceral: Também conhecido como automático e pré-programado porque é o primeiro impacto que o produto causa no usuário. Segundo Norman (2008) esse nível é mais encontrado em propaganda, artesanato popular e produtos infantis.

Nível Comportamental: Está relacionado diretamente ao uso do produto e aos seus aspectos funcionais.

Norman (2008) afirma que para um design ser bom ele precisa apresentar quatro componentes:

Função: O produto tem que atender as necessidades do usuário.

Compreensibilidade: O usuário tem que saber qual a finalidade do produto, não tem como o usuário utilizar o produto sem saber qual é a sua serventia. De acordo com Norman (2008), quando não entendemos como determinado objeto funciona é provável que gere várias emoções negativas.

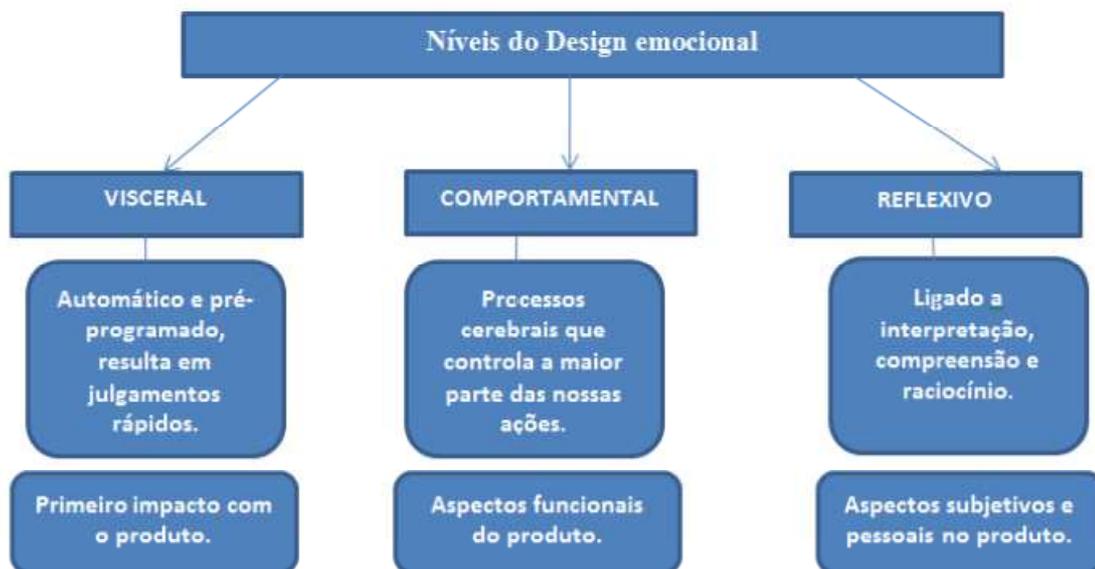
Usabilidade: O produto deve ser fácil de usar, se o produto não for fácil de usar o usuário se sentirá desmotivado.

Sensação física: O usuário deve usar o produto, deve ter contato físico.

Nível Reflexivo: Está relacionado à compreensão do usuário em relação ao seu meio, De acordo com Norman (2008), este nível está relacionado às lembranças e significados que os produtos causam.

Os três níveis do design emocional de Norman (2008) são representados abaixo na figura 10.

Figura 10: Os níveis cerebrais do design emocional



Fonte: Próprio autor adaptado de Norman (2008).

Enquanto a usabilidade trata da facilidade de uso, o design emocional se preocupa em saber a satisfação do usuário em usar determinado produto, ou seja, em vez de “penso e faço” ele está mais preocupado “sinto e faço”. Segundo Norman (2008), o modo como pensamos mudam a partir das nossas emoções. Ainda segundo o autor as nossas emoções servem para ajudar a termos um comportamento apropriado.

Segundo Memória (2005), existem algumas normas e boas práticas que podem ser aplicadas em interfaces para servir como respostas a três questões dos usuários: São elas:

1. Onde estou.
2. Onde Estive.
3. Onde posso ir.

Segundo Norman (2008), as pessoas cometem erros na utilização de alguns produtos mal elaborados e com esse erro os usuários acreditam que é culpa deles, ou seja, que eles não sabem fazer uso do produto, mas se isso acontece como muitas pessoas podem ser problemas de design mal feitos e mal elaborados. Com esses tipos de produtos os usuários não têm culpa, eles são apenas vítimas de produtos mal desenvolvidos.

Segundo Berkun (2001), as pessoas levam certo tempo para aprender algo, mas quando aprendem, conseguem fazer sem precisar pensar em como está realizando determinada atividade, como por exemplo, aprender a ler e a andar de bicicleta. Isso acontece da mesma forma com os *softwares*, quando o usuário aprende, ele não precisa estar pensando como realizar determinado processo.

Existem vários tipos de avaliações e métodos, mas o foco desse trabalho é analisar as respostas das crianças com base no design emocional ao utilizar o *software* em estudo.

2.2.3 Avaliação de *software* educacional

A avaliação de um *software* educacional é necessária já que eles vão ser utilizados dentro da sala de aula para fins educativos tendo como objetivo auxiliar os professores nas aulas tornando assim o entendimento dos assuntos mais fácil, rápido e prazeroso. De nada adianta inserir um *software* em sala de aula e achar que ele irá auxiliar o professor só pelo fato de ser um *software* educacional. Para Santos e Amaral (2012), existem várias metodologias voltadas para avaliação de *softwares* educacionais e essas metodologias procuram sempre serem eficazes no ponto de vista educacional. Ainda segundo eles o foco da avaliação de *softwares* educacionais está ligado diretamente à melhora de resultados educacionais.

Um *software* é dito como educacional quando o seu objetivo é auxiliar o processo de ensino-aprendizagem, e neles existem assuntos educacionais. “Um *software* educativo é um programa que é usado para alguma finalidade educacional, mas não, necessariamente, que foi concebido para tal, como é o caso da planilha eletrônica” (Cristovão e Nobre, 2011, p.127).

O *software* ele não veio para substituir o professor e sim para auxiliar nesse processo de ensino-aprendizagem, fazendo com que essa aprendizagem se torne prazeroso e assim cause motivação nos alunos.

Avaliar um *software* já é complicado e quando esse *software* é voltado para o ensino Quando um *software* é avaliado não se deve levar em conta apenas aspectos técnicos no que

diz respeito à engenharia de *software*, mas também a parte pedagógica, ele deve ser avaliado também por profissionais da educação.

Se os *softwares* educacionais não forem bem avaliados não irão satisfazer as necessidades dos alunos, pelo contrário, podem dificultar ainda mais o seu aprendizado. Não adianta o professor pensar em qual *software* e inserir em sala de aula é preciso pensar, antes de tudo, como esse *software* será utilizado, qual proposta será adota e qual a finalidade dessa proposta.

Levando em consideração as ferramentas de programação apresentadas no presente trabalho, torna-se necessário uma avaliação criteriosa, principalmente quando estas ferramentas são destinadas para o ensino a crianças. Dentre vários tipos de avaliação de *software* educacional é possível destacar o Design Emocional.

3 METODOLOGIA

O percurso metodológico de uma pesquisa é de fundamental importância para estabelecer quais passos serão necessários para que os objetivos definidos para o trabalho sejam alcançados. Neste sentido, os passos desta pesquisa foram especificados com o intuito de possibilitar a avaliação do *software* Kodu Game Lab com base nos conceitos do design emocional.

Inicialmente, foi realizada uma pesquisa exploratória com o objetivo de realizar um levantamento bibliográfico em torno dos temas relacionados a esta pesquisa. Em seguida, foi definida como estudo de caso uma turma do projeto de extensão promovido pela Universidade Estadual da Paraíba (UEPB) campus VII intitulado como “Inclusão digital e informática educativa na rede pública” sob a orientação do professor Wellington Candeia de Araújo.

O objetivo deste projeto em questão é “Ensinar programação e Jogos aos alunos do Ensino Fundamental das escolas Públicas”, Esse projeto teve duração de três meses, mas ainda pretende dá continuidade com outras turmas.

O projeto teve uma parceria entre a Escola Estadual do Ensino Fundamental Coriolano de Medeiros e o Telecentro Cajueiro ambos localizados na Cidade de Patos-PB. A escolha da escola aconteceu pelo fato de ser a mais próxima do Telecentro. Para a escolha dos participantes, teve inicialmente a aplicação de um teste de lógica com todos os alunos da turma e os que se saíram melhor no teste de lógica foram selecionados para participar do projeto, a seleção foi necessária pelo fato de existir poucos computadores no Telecentro.

Após selecionar os alunos, foi necessário conversar com os pais dos alunos para saber se eles poderiam participar do projeto, o projeto acontecia uma vez na semana em horário oposto as aulas. A partir disto, foram selecionados dez alunos de uma turma do 7º ano.

Com a conclusão das aulas, foi aplicado um questionário aos alunos participantes com o intuito de avaliar o *software* com base na análise de sentimentos através de *emoticons* como proposto pelo design emocional, tal abordagem facilita a análise das experiências dos alunos em relação ao uso do *Software*.

O questionário é um dos principais métodos para colher opinião dos usuários, no questionário será possível identificar como o usuário se sente ao utilizar determinado produto. Os tipos de questionários e entrevistas a serem utilizados variam em diversos aspectos (NICOLACI-DA-COSTA, 2001; PREECE et al., 2002).

O questionário aplicado foi composto de doze questões, sendo onze objetivas e uma discursiva, e as questões foram definidas com o objetivo de extrair informações sobre a experiência de utilização da ferramenta, as questões estão apontadas no Quadro 2, e encontra-se na íntegra no Apêndice A.

Quadro 2: Questões exploradas no questionário da pesquisa.

| Questão | Pergunta |
|---------|---|
| 1. | Qual seu gênero? |
| 2. | Você gosta de jogar videogames? |
| 3. | Você faz uso de jogos com frequência? |
| 4. | Quais os tipos de jogos que você mais utiliza? |
| 5. | O Kodu Game Lab é fácil de mexer? |
| 6. | Você demorou aprender a utilizar o Kodu Game Lab? |
| 7. | Você conseguiu criar o jogo que queria? |
| 8. | O que você sentiu quando conseguiu criar o jogo que desejava? (Aponte os <i>emoticons</i>). |
| 9. | Como você aponta a experiência de utilizar o Jogo? |
| 10. | Teve funcionalidades que você não conseguiu utilizar? |
| 11. | Aponte os <i>emoticons</i> que estão relacionados às suas emoções/sentimentos ao utilizar os recursos de navegação (menus, ícones, links e botões). |
| 12. | O que você achou da experiência de utilizar este questionário? |

Fonte: Próprio autor

Para elaboração dessas questões, vários critérios foram levados em consideração, dentre eles: o *software* utilizado, objetivo de avaliação, público alvo, etc. E como aplicação do mesmo foi a partir do projeto de extensão citado anteriormente, algumas dessas variáveis foram definidas com base no projeto.

A escolha do *software* para avaliação além dos fatores destacados na seção que explora o *software* Kodu Game Lab, como facilidade de uso, ser gratuito, atender os requisitos do sistema operacional Linux. Ainda foi reforçado por ser o *software* utilizado no projeto o que facilitou a aplicação do questionário. O primeiro critério levado em consideração foi à escolha do *software*, cuja opção teve base em um *software* que objetivasse o ensino de programação para crianças, já que é um assunto bem comentado nos debates atuais.

O segundo critério foi o público alvo, o público alvo escolhido são crianças entre 09 e 11 anos que estudam em escola da rede pública de Patos. A faixa etária foi baseada no público do Kodu Game Lab pelo fato do *software* ser direcionado também para esse público.

Estas escolas deveriam possuir laboratórios de informática, em bom estado de funcionamento, que atendesse aos requisitos da ferramenta. Entendendo as barreiras que são encontradas para conseguir acessar os laboratórios das escolas públicas, foi necessário fazer uma parceria com o Telecentro.

Para as questões objetivas eram apontados diversos *emoticons* para que o usuário pudesse expor suas experiências, sentimentos e/ou opiniões em cada questionamento. A lista de *emoticons* utilizados está ilustrado na figura 19 abaixo.

Figura 11: *Emoticons* utilizados no questionário

| | | | | | | |
|---|---|---|---|---|---|---|
|  |  |  |  |  |  |  |
| Tédio, Chato | Nervoso | Insatisfeito | Raiva | Aliviado | Aborrecido, irritado | Confuso |
|  |  |  |  |  |  |  |
| Triste | Exausto | Pouco feliz | Bem | Apaixonante | Feliz | Muito Feliz |

Fonte: Próprio autor com base no *emojipedia*¹³

Para a análise dos resultados, serão utilizadas as abordagens quantitativa e qualitativa. Segundo Minayo e Souza (2005), a abordagem quantitativa é usada na apresentação de resultados que podem ser contados e esse tipo de abordagem é importante para avaliar objetivos bastante específicos e estabelecer relações significativas entre variáveis. Mas, a qualitativa também se mostra extremamente importante. Segundo Siena (2007), esse tipo de pesquisa é considerada descritiva a cerca da organização das informações obtidas durante o estudo.

3.1 DESCRIÇÃO DO *SOFTWARE* EDUCACIONAL KODU GAME LAB

O Kodu Game Lab foi desenvolvido pelo laboratório de pesquisas FUSE (Future Social Experiences) Labs, mantido pela Microsoft, ele encontra-se disponível grátis para

¹³ *Emojipedia*: disponível: <http://emojipedia.org/>

download¹⁴. O *software* Kodu tem como objetivo ajudar crianças e jovens a iniciar uma programação orientada a objetos sem precisar de conhecimento das linguagens tradicionais, sendo assim será possível à criação de jogos apenas com uma programação simples e intuitiva. O *software* teve sua primeira versão lançada em 2010, ele está disponível grátis para Windows e paga para Xbox.

3.1.1. Instalação

Antes de instalar o Kodu é preciso instalar os *plugins*. NET e XNA Framework, para identificar se o *software* foi realmente instalado é só verificar se os dois ícones, Kodu Game Lab e Configure Kodu Game Lab, apresentados na figura abaixo estão na área de trabalho do computador.

Figura12: Ícones instalação Kodu



Fonte: Próprio autor

3.1.2. Funcionalidades

Após a instalação e a verificação do *software*, é só clicar duas vezes no ícone do Kodu Game Lab e já será possível utilizar o *software*. A figura 13 apresenta a tela inicial do Kodu Game Lab.

¹⁴ <http://fuse.microsoft.com/project/kodu.aspx>

Figura 13: Tela inicial



Fonte: Próprio autor

Na tela inicial são apresentadas sete funcionalidades, todas bem intuitivas e de fácil compreensão:

- Retomar: Permite que o usuário retome seu jogo;
- Novo mundo: Criar novo jogo;
- Carregar mundo: Carregar o jogo;
- Comunidade: Na comunidade do Kodu o aluno tem a oportunidade de publicar seus jogos e ter acesso aos jogos de outras pessoas;
- Opções: Em opções o usuário tem acesso às configurações do jogo, bem como ao idioma do *software*;
- Ajuda: É direcionado ao tutorial do jogo;
- Abandonar Kodu: Sair do *software*.

3.1.3 Criação de Jogo

Clicando na opção novo mundo, o Kodu abre e exibe um cenário vazio, sem água, montanhas, árvores ou veículos, em forma de quadrado.

Figura 14: Cenário vazio

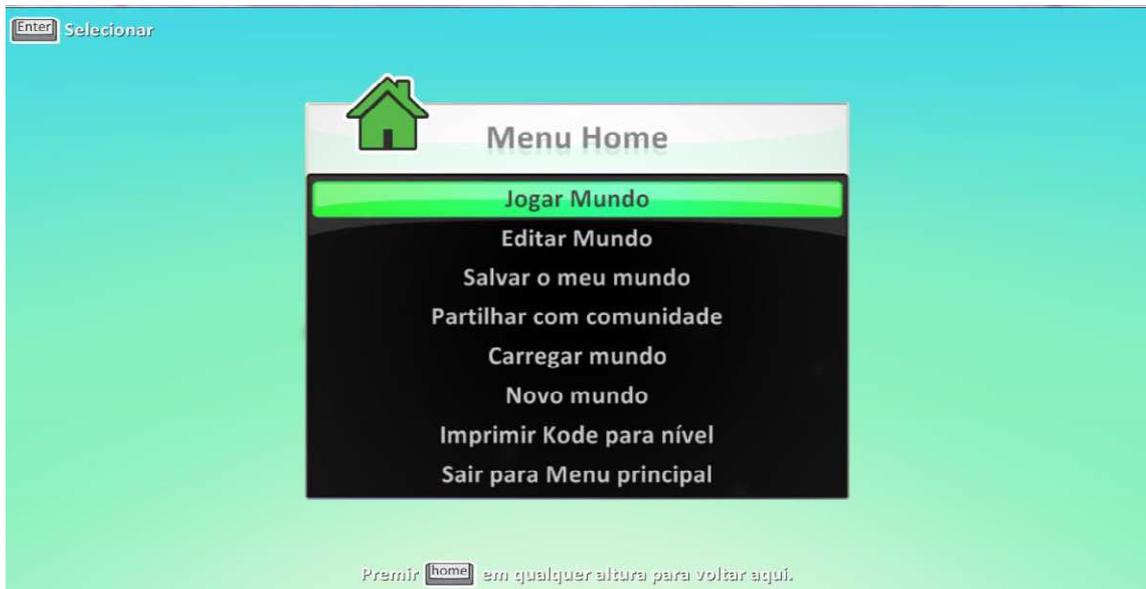


Fonte: Próprio autor

O Kodu sempre irá oferecer uma lista de indicações no canto superior esquerdo da tela, indicando quais as teclas responsáveis por cada ação possível. As teclas mais utilizadas são: Botão direito, botão esquerdo e Esc.

Na parte inferior estão todas as ferramentas do Kodu, clicando no ícone da casinha o usuário será redirecionado para o Menu (figura 15). As demais opções servem para ver o funcionamento do jogo, escolha do tipo de solo do cenário, formato do pincel que irá modificar o tipo de solo, tipos de solos disponíveis, criar relevos no solo como morros e vales, aplainar relevos, criar picos montanhosos, criar lagos nas áreas rebaixadas do relevo, construir um caminho ou uma área cercada no cenário, tipo de material a ser colocado no caminho criado e se não gostar do resultado é só desfazer.

Figura 15: Menu



Fonte: Próprio autor

Depois de criar o cenário é só incluir os personagens e objetos, na figura abaixo estão os personagens e objetos que podem ser adicionados ao cenário. O Kodu não permite que o usuário crie seus próprios personagens, ou seja, só será possível utilizar os que o jogo oferece.

Figura 16: Ferramenta de objetos



Fonte: Próprio autor

Cada personagem usado para criar jogos que é colocado no cenário também é chamado de objetos, cada um desses objetos pode ser programado para realizar um conjunto de funções.

Para configurar as características e as ações do personagem incluído, é só pressionar o botão direito do mouse sobre ele. A partir daqui é que começa a programação, sendo essa programação rápida e simples.

Para programar as ações do personagem é só clicar na opção: programar, como mostra a figura 17 abaixo.

Figura 17: Modificação de objetos



Fonte: Próprio autor

Após escolher a opção “programar” poderá agora programar o objeto individualmente. Essa programação é simples e pode ser visualizada na figura 18 abaixo.

Figura 18: Programação do objeto

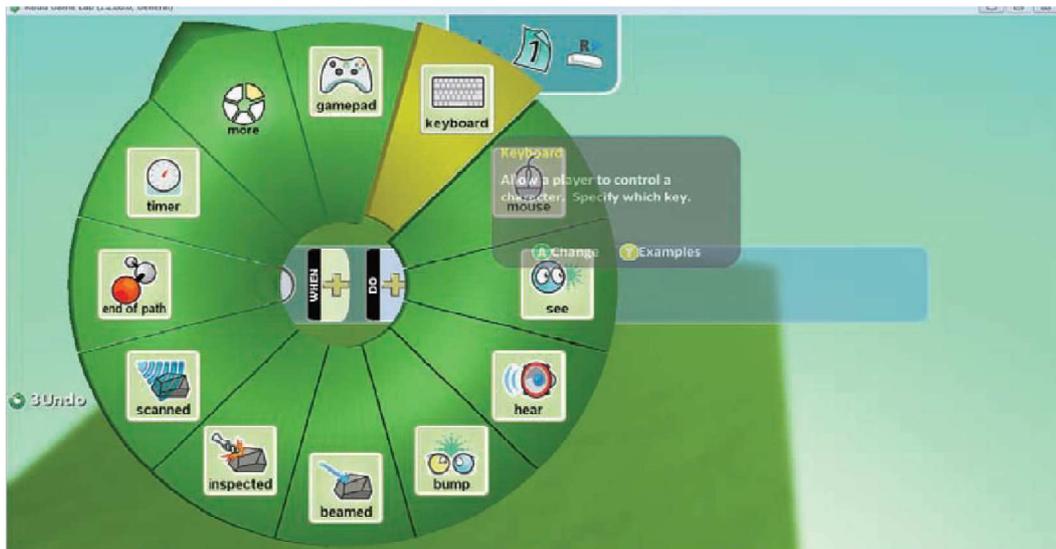


Fonte: Próprio autor

Na linha de programação existem dois ícones “Quando” e “Faz”. Em outras palavras isso significa que quando um evento acontecer, o personagem ou objeto deverá fazer algo. Para iniciar a programação é só clicar “+” com o botão esquerdo do mouse.

Ao clicar no botão “+” a frente do “Quando”, aparecem várias opções como, por exemplo, quando usar o teclado faça algo. Ver figura 19 abaixo.

Figura 19: Adicionar opção “Quando”



Fonte: Próprio autor

Agora é necessário determinar o que o personagem vai fazer quando determinado evento acontece, para escolher é só clicar no botão “+” a frente do “Faz”.

Figura 20: Adicionar opção “Faz”



Fonte: Próprio autor

Nas figuras 19 e 20 foi possível mostrar a programação do objeto que significa dizer que quando o usuário clicar no teclado o personagem irá fazer movimentos.

Para excluir qualquer expressão, (figura 21) é preciso selecionar “Corta Mosaico” com o botão direito do mouse.

Figura 21 Excluir expressão



Fonte: Próprio autor

Depois de concluir o jogo é só clicar na opção “Play”, para salvar o novo mundo é só clicar no botão “salvar o meu mundo” figura 22. O Kodu permite que o jogo seja modificado e renomeado a qualquer momento.

Figura 22: Salvar mundo



Fonte: Próprio autor

Cada objeto tem suas características, para acessar essas características no Kodu é só selecionar a opção “alterar definições”. Selecionando esta opção, a lista de rolagem que contém as configurações do objeto aparece, como mostra a figura 23.

Figura 23: Alterar definições do objeto



Fonte: Próprio autor

No Kodu também é possível criar a atividade com o cenário já pronto, ele também permite que explore outros jogos criados dentro dele, onde esses jogos foram criados por pessoas que também fazem uso do jogo. O Kodu também permite que os usuários compartilhem seus jogos com outros usuários.

3.2 ESTUDO DE CASO

Os questionários foram aplicados na Escola Estadual do Ensino Fundamental Coriolano Medeiros da cidade de Patos-PB, participaram da aplicação dos questionários 10 alunos, de onze anos, que fazem parte do 7º ano do ensino fundamental. O questionário contém doze perguntas, sendo onze objetivas e uma discursiva.

O uso de *emoticons* para expressar emoções foi a estratégia adotada para a elaboração do questionário. Os *emoticons* são símbolos que servem para expressar as emoções, eles são usados para expressar os sentimentos e são usados com frequência em Chats. Para Martins (2007: 92), “Sempre existirá a necessidade de símbolos de integração mundial”. Esses símbolos que ele se refere são os *emoticons*.

Às vezes a escrita não deixa bem claro o que a mensagem quer transmitir, é por isso que muitas vezes as palavras acabam sendo interpretadas de forma diferente, com o uso de *emoticons* será possível deixar bem claro, de forma resumida, as expressões emocionais.

A imagem é uma linguagem de fácil entendimento para as crianças, fazer uso de *emoticons* com crianças é uma forma de fazer com que elas se sintam mais atraídas e motivadas, já que é uma linguagem de fácil compreensão que torna a leitura bem mais atraente e menos cansativa, além disso, faz com que ela se aproxime do seu mundo.

Sendo assim, o uso de *emoticons* em questionários faz com que as crianças consigam fazer uma ligação entre as palavras e os símbolos, facilitando assim expressar seus sentimentos diante do que foi perguntando com base na utilização da ferramenta.

As crianças irão responder os questionários para expressar seus sentimentos ao utilizar o *software* e para isso terão o auxílio dos *emoticons* para traduzir.

4 RESULTADOS E DISCUSSÕES

Na próxima seção, apresentam-se os resultados alcançados com base no objetivo do trabalho e a análise dos dados obtidos através da aplicação dos questionários. Os dados coletados foram expostos em gráficos.

4.1 Análise dos Resultados

A primeira pergunta foi relacionada ao gênero dos alunos, 50% do público entrevistado era do gênero feminino e 50% era do gênero masculino, porém o gênero não afetará os resultados da pesquisa, já que tanto os meninos quanto as meninas estão aptos a utilizar a ferramenta. A idade dos alunos que participaram do questionário são onze anos, a idade é um fator importante, ela pode afetar os resultados da pesquisa, pois cada faixa etária possui uma maneira diferente de reagir perante um produto.

Na segunda questão o usuário foi questionado para saber se eles gostam de jogar videogames, 90% dos entrevistados responderam que sim e apenas 10% responderam que não.

O fato das crianças gostarem de utilizar os videogames influencia no seu emocional em relação aos jogos, se a criança gosta de jogar videogames ela terá um maior interesse em utilizar os *softwares* educacionais, pelo fato dela ter maior facilidade em utilizar.

Segundo Norman (2008) um objeto favorito é um símbolo que induz a pensamentos positivos em relação ao objeto, então se as crianças têm pensamentos positivos em relação aos jogos que elas utilizam em videogames isso fará com que elas tenham recordações boas ao fazerem uso dos *softwares* educacionais, essas lembranças facilitará a utilização de outros jogos já que o usuário tem alguma noção de como funciona o jogo. O design emocional é justamente para contribuir com essas relações entre o usuário e os objetos.

Para sinalizar as respostas possíveis do questionário foram usados os dois *emoticons* (Sim e Não) como mostra na figura 24

Figura 24: Resposta sinalizada do questionário

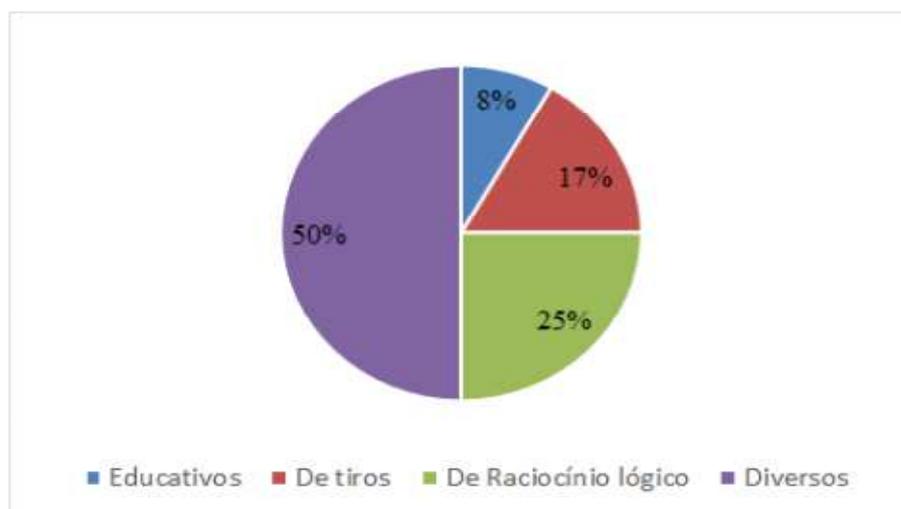


Fonte: Próprio autor com base nos *emoticons*

Na terceira questão foram questionados se eles têm acesso aos jogos com frequência, 90% dos entrevistados responderam que sim e 10% dos entrevistados responderam que não, os 90% que utilizam os jogos com frequência são justamente os que responderam na questão anterior que gostavam de jogar videogames. Pode-se concluir que quase todos utilizam os jogos com frequência. A utilização de jogos com frequência é muito importante, pois as crianças terão mais facilidade em aprender a utilizar outros jogos, elas estão inseridas nesse meio e isso facilitará o entendimento dos *softwares* educacionais.

Nessa questão entra o terceiro nível do design emocional, o nível reflexivo, que é quando o usuário é atraído por suas memórias e emoções (Norman, 2004). Se o usuário faz uso de jogos com frequência ele terá lembranças de como os jogos funcionam, facilitando assim, o seu entendimento em outros jogos.

Gráfico 1: Quais os tipos de jogos que você mais utiliza.



Fonte: Próprio autor

Ao analisar o gráfico, percebe-se que 50% dos entrevistados utilizam diversos jogos, esses jogos incluem jogos de competição, jogos interativos, jogos de esportes, entre outros. 25% dos entrevistados utilizam jogos de raciocínio lógico. O uso de jogos de raciocínio lógico estimula a criatividade da criança e a desafia a sair de obstáculos para solucionar problemas. 17% alegaram que utilizavam jogos de tiros.

Segundo GIL (2005) dentre os jogos mais procurados estão os de tiros, lutas e estratégias de guerra. Os jogos de estratégia são importantes, pois eles desenvolvem nas crianças a habilidade de solucionar problemas. E os jogos de tiros, que são os jogos de ação, ajudam as crianças a pensarem rapidamente e a desafiar a mente.

O que chamou mais atenção no gráfico foi referente à utilização dos jogos educativos, porque apenas 8% dos entrevistados afirmaram que utilizam jogos educativos. Isso é preocupante, já que os jogos educativos deveriam ser mais utilizados nas escolas. Com isso faz-se necessário a presença de um Licenciado em Computação, para incentivar o uso dos jogos educacionais nas diversas disciplinas. Falkemback (2013) caracteriza os jogos educativos como o envolvimento emocional, pois os jogos têm a capacidade envolver emocionalmente o participante.

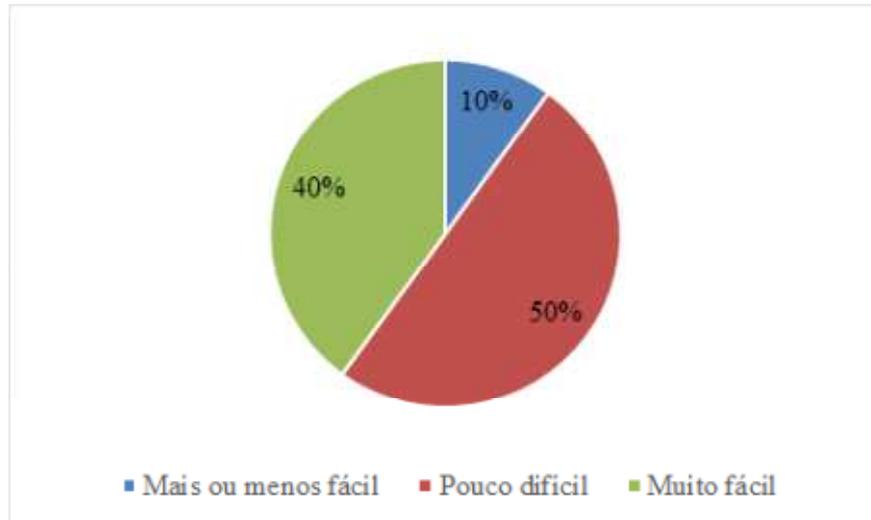
A quinta questão do questionário diz respeito à facilidade de mexer no *Software* em estudo, no Quadro 3 será possível visualizar as respostas possíveis.

Quadro 3: *Emoticons* para representar as respostas possíveis da questão 05

| | | | | |
|---|---|---|--|---|
|  |  |  |  |  |
| Muito Difícil | Difícil | Mais ou menos Fácil | Pouco Difícil | Muito Fácil |

Fonte: Próprio autor

Em relação à facilidade de mexer no Kodu Game Lab, os resultados apontados no Gráfico 2 mostram que 50% dos entrevistados consideraram pouco difícil de mexer, 40% muito fácil e 10% mais ou menos fácil. Ao analisar o gráfico, percebe-se que a maioria das crianças teve facilidade em utilizar o *software*. A facilidade de usar o produto é extremamente importante, pois quando o produto é difícil de usar o usuário poderá ficar frustrado e assim ter vários sentimentos negativos que podem assim interferir na utilização do produto e consequentemente na aprendizagem.

Gráfico 2: O Kodu Game Lab é fácil de mexer:

Fonte: Próprio autor

É no nível comportamental que Nornam (2008) trata dos aspectos funcionais dos produtos, ele aponta a usabilidade como um dos componentes de extrema importância para o uso em produtos.

Quando o produto é fácil de usar o usuário começa a ter prazer pelo produto, Segundo Jordan (2001), além dos produtos apresentarem um bom funcionamento e serem fáceis de usar, eles devem também proporcionar prazer, ser agradáveis e desencadear emoções positivas no usuário.

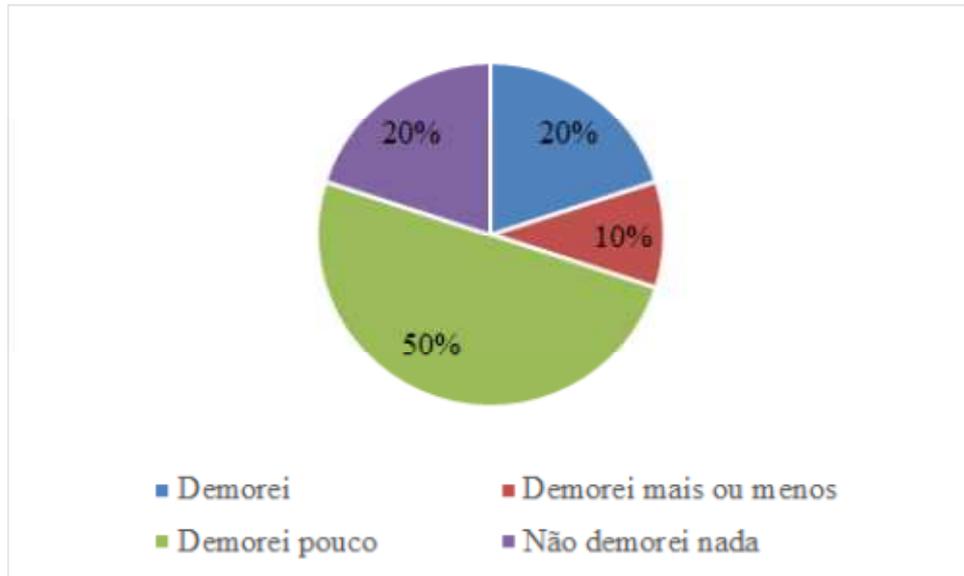
A Sexta pergunta tinha como objetivo verificar se os alunos demoraram aprender a utilizar o Kodu Game Lab, as possíveis respostas podem ser visualizadas no quadro 04.

Quadro 4: Emoticons para representar as respostas possíveis da questão 06

| | | | | |
|---|---|---|---|---|
|  |  |  |  |  |
| Demorei muito | Demorei | Demorei mais ou menos | Demorei pouco | Não demorei nada |

Fonte: Próprio autor

Com a análise do Gráfico 3 é possível perceber que 50% dos entrevistados responderam “Demorei pouco”, 20% responderam “Não demorei nada” e 20% responderam “Demorei”. Percebe-se que a maioria dos entrevistados não teve dificuldade em aprender a utilizar o Kodu Game Lab.

Gráfico 3: Você demorou aprender a utilizar o Kodu Game Lab:

Fonte: Próprio autor

Quando o usuário demora aprender a utilizar determinado produto, ele começa a ter sentimentos negativos, esses sentimentos fazem com que o usuário perca a vontade de utilizar, quando se fala em criança esses cuidados têm que ser dobrado. Segundo Norman (2008) quando o usuário tenta utilizar um determinado produto mais três vezes é bem provável que ele desista de continuar usando.

Na sétima pergunta os alunos foram questionados se eles conseguiram criar o jogo que desejava. Os *Emoticons* que representam as respostas podem ser visualizados no quadro 5.

Quadro 5: *Emoticos* para representar as respostas possíveis da questão 07

| | | | | |
|---|---|---|--|---|
|  |  |  |  |  |
| Não consegui nada | Cosegui pouco | Conseguí mais ou menos | Conseguí | Conseguí tudo |

Fonte: Próprio autor

Percebe-se no Gráfico 4 que 55%, mais da metade, das crianças conseguiram criar o jogo da forma que queria, mas ainda teve 9% que não conseguiu criar o jogo que desejava. 18% responderam “Conseguí mais ou menos” e 18% responderam “Conseguí tudo”. As respostas ficaram concentradas no “Conseguí”, Segundo Jordan, (1999b) quando o usuário consegue realizar o que deseja ele desperta prazeres psicológicos.

Gráfico 4: Você conseguiu criar o jogo que queria:

Fonte: Próprio autor

A questão oito é quanto aos sentimentos que despertaram na criança ao conseguir criar o jogo que desejava.

Quadro 6: Emoticons para representar as respostas possíveis da questão 08:

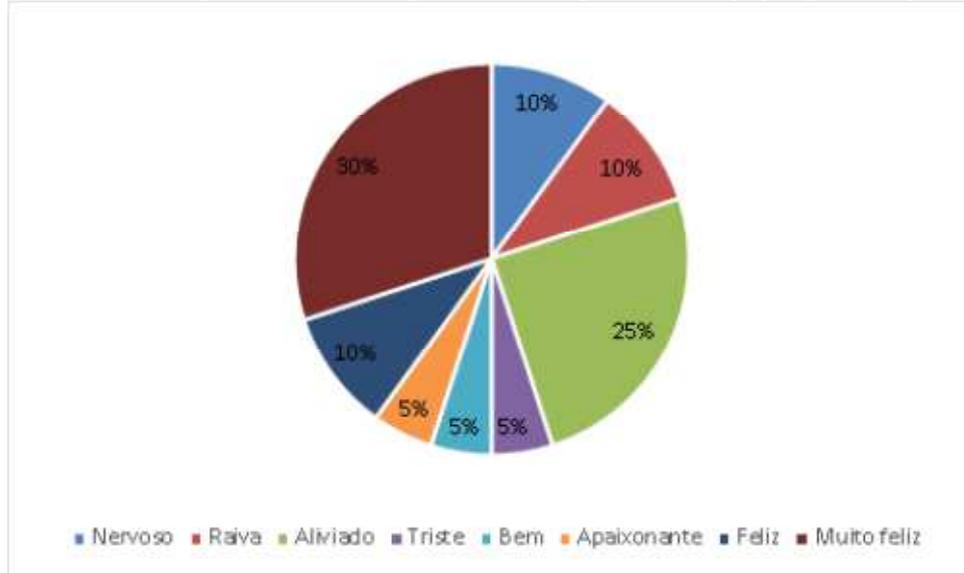
| | | | | | | |
|-----------------|---------|--------------|-------|-------------|-------------------------|-------------|
| | | | | | | |
| Tédio, Chato | Nervoso | Insatisfeito | Raiva | Aliviado | Aborrecido, irritado | Confuso |
| | | | | | | |
| Triste | Exausto | Pouco feliz | Bem | Apaixonante | Feliz | Muito Feliz |

Fonte: Próprio autor

Os sentimentos apontados pelas crianças ao conseguir criar o jogo que queriam foram: “Nervoso”, “Raiva”, “Aliviado”, “Triste”, “Bem”, “Apaixonante”, “Feliz” e “Muito feliz”.

É possível observar no Gráfico 5 que a maioria dos sentimentos apontados pelas crianças é “Muito Feliz” e “Aliviado”. Segundo Xavier (2013), as pessoas tornam-se mais criativas e imaginativas quando elas se sentem mais relaxadas e felizes. Os sentimentos positivos despertam motivação nas pessoas, se um *software* só provocar sentimentos negativos à criança não terá estímulo para utilizá-lo.

Gráfico 5: O que você sentiu quando conseguiu criar o jogo que desejava:



Fonte: Próprio autor

Observa-se ainda na análise desta questão que algumas crianças tiveram mais dificuldade do que as outras e acabaram gerando sentimentos negativos em relação ao jogo, como por exemplo, “raiva” em 10% das crianças e “triste” em 5%.

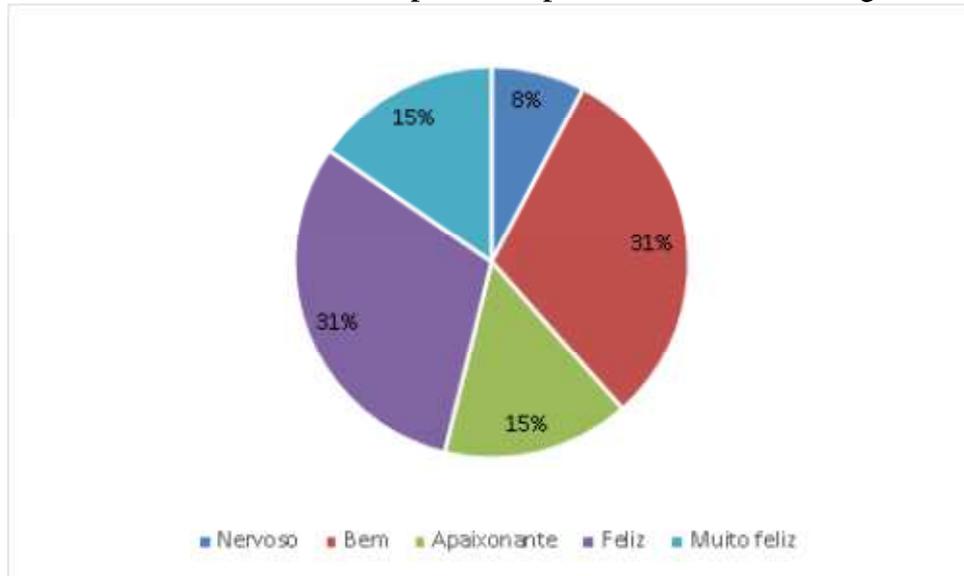
Para “apontar a experiência de utilizar o jogo” o usuário foi questionado a partir dos *emoticons* representados no quadro 7.

Quadro 7: *Emoticons* para apontar a experiência de utilizar o jogo

| | | | | | | | | | |
|-----------------|---------|-------|------------------------|--------|---------|-----|-----------------|-------|----------------|
| | | | | | | | | | |
| Tédio, Chato | Nervoso | Raiva | Aborrecido irritado | Triste | Exausto | Bem | Apaixona nte | Feliz | Muito Feliz |

Fonte: Próprio autor

E como resultado foi interessante observar os sentimentos que mais se destacaram, sendo eles “Feliz” e “Bem”. No Gráfico 6 estão elencados os sentimentos mais apontados para o questionamento.

Gráfico 6: Como você aponta a experiência de utilizar o Jogo

Fonte: Próprio autor

A maioria dos sentimentos foram positivos, tendo o sentimento “feliz” com 31%, “apaixonante”, 15% e “muito feliz” 15%. Apenas 8% das crianças apontaram que se sentiram nervosas com a experiência de utilizar o *software*. Segundo Norman (2004) as pessoas conseguem resolver os problemas com mais facilidade quando estão felizes, porém, pessoas ansiosas ou com medo não conseguem resolver os mesmos problemas.

A décima questão teve como objetivo saber se teve funcionalidade no *software* que as crianças não conseguiram utilizar. As possíveis respostas são visualizadas no quadro 8 abaixo.

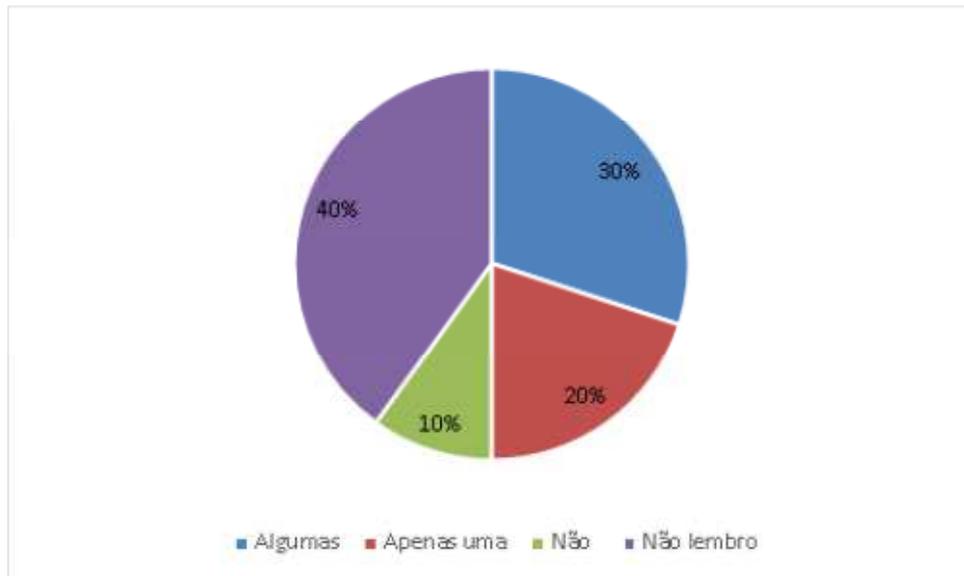
Quadro 8: Respostas possíveis para a questão 10.

| | | | | |
|--------|---------|------------|-----|------------|
| | | | | |
| Muitas | Algumas | Apenas uma | Não | Não lembro |

Fonte: Próprio autor

Percebe-se no gráfico 7 que a maioria das respostas são “Não lembro” e “Algumas”.

Gráfico 7: Teve funcionalidades que você não conseguiu utilizar:



Fonte: Próprio autor

Em relação à funcionalidade do *software*, 40% responderam que não lembrava se tinha funcionalidade que não conseguiam utilizar, 30% responderam “algumas”, 20% apenas uma e apenas 10% responderam “não”.

A função do produto é um dos quatro componentes que compõe o nível comportamental proposto por Norman (2008), quando o produto não funciona da forma como devia o usuário começa a despertar diferentes reações emocionais que são traduzidas em sentimentos negativos. É importante destacar que o *software* não teve tanta aceitação em relação as suas funcionalidades.

A décima primeira questão é sobre as emoções/sentimentos que estão relacionados ao utilizar os recursos de navegação (menus, ícones, links e botões). No quadro 9 será visualizado as possíveis respostas.

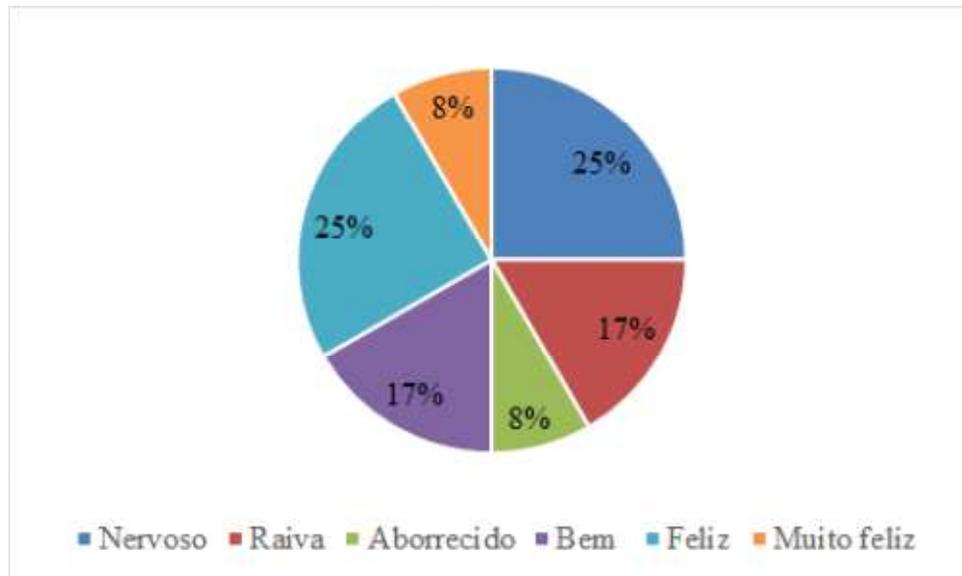
Quadro 9: Emoticons para representam as possíveis respostas da questão 10

| | | | | | | | | | |
|-----------------|---------|-------|-----------------------|--------|---------|-----|-----------------|-------|----------------|
| | | | | | | | | | |
| Tédio, Chato | Nervoso | Raiva | Aborrecido imitado | Triste | Exausto | Bem | Apaixona nte | Feliz | Muito Feliz |

Fonte: Próprio autor

Ao analisar o gráfico 8 percebe-se que os sentimentos mais selecionados pelas crianças foram “Nervoso” e “Feliz”.

Gráfico 8: Aponte os *emoticons* que estão relacionados às suas emoções/sentimentos ao utilizar os recursos de navegação (menus, ícones, links e botões):



Fonte: Próprio autor

Foi interessante observar que nessa questão os sentimentos se concentraram em positivos e negativos, ou seja, 50% dos sentimentos apontados pelas crianças foram positivos e 50% foram negativos.

Os sentimentos apontados pelos entrevistados em relação aos recursos de navegação do *software* foram 25% “nervoso” e 25% feliz, 17% apontaram o sentimento de raiva e 17% apontaram bem. 8% apontaram “muito feliz e 8% “aborrecido”.

Os problemas relacionados aos recursos de navegação fez gerar alguns sentimentos negativos em algumas crianças, como: Raiva, Nervosismo e Aborrecimento. Esses sentimentos deixam as crianças sem motivação para continuar usando o produto.

Segundo Norman (2008, p.28), “As emoções negativas se manifestam quando há falta de compreensão, quando as pessoas se sentem frustradas e fora de controle – primeiro vem à inquietação, depois irritação e, se a falta de controle e compreensão persistirem, até a raiva”.

A questão doze do questionário era discursiva e perguntava o que a criança achava da experiência de utilizar o questionário. As respostas que mais apareceram foi “Criativo”, “Legal” e “Interessante”. Com base nas respostas observa-se que as crianças gostaram de

utilizar o questionário para representar seus sentimentos com base na utilização do Kodu Game Lab.

Entre as respostas a que chamou mais atenção foi de uma criança que disse: “Muito bom, pelo menos contamos o que sentimos”. Os *emoticons* utilizados nos questionários fizeram com que as crianças conseguissem fazer uma ligação entre as palavras e os símbolos, facilitando assim expressar seus sentimentos diante do que foi perguntando com base na utilização do *Software*.

5 CONSIDERAÇÕES FINAIS

A avaliação do Design Emocional impede que os usuários tenham respostas negativas relacionadas ao uso do produto, a satisfação de uma criança, por exemplo, ao utilizar determinado *software* pode decidir se ele continuará ou não usando.

Este trabalho teve como principal objetivo avaliar o *software* educacional com base nos conceitos do Design Emocional. Para essa pesquisa ser realizada, foi preciso contar com apoio do público usado no projeto de extensão da Universidade Estadual da Paraíba campus VII.

A avaliação com base no Design Emocional pode apontar reações positivas ou negativas, com base nos resultados e na análise dos gráficos percebe-se que a maioria dos sentimentos despertados pelo Kodu Game Lab foram sentimentos positivos. Os principais sentimentos apontados pelos entrevistados foram: feliz, bem e muito feliz. Com base nas pesquisas relacionadas em torno do Design Emocional, verificou-se que esses sentimentos geram motivação nas crianças e desperta criatividade resultando assim, em uma aprendizagem mais eficiente e de qualidade.

Quando o *Software* Educacional não apresenta uma boa usabilidade, ou seja, quando não é fácil de usar ou não atende as necessidades dos usuários, isso poderá causar várias emoções, essas emoções despertam sentimentos negativos que contribuirão para afetar a sua qualidade e possivelmente o aprendizado do aluno. O *Software* em avaliação teve grande aceitação em relação à facilidade de usar, percebe-se que a maioria das crianças conseguiu utilizá-lo.

Os sentimentos de “raiva”, “Nervoso” e “aborrecido”, foram alguns dos sentimentos negativos provocados nas crianças ao utilizar o *software*, esses sentimentos podem interferir na qualidade do produto. Os jogos proporcionam desafios, esses desafios estimulam à criatividade da criança ocasionando uma tensão no jogador, quando ele consegue resolver esse desafio, a tensão vai embora e vem o alívio junto com alegria. É esse sentimento de alegria que motiva o usuário a continuar utilizando o produto.

Com base no questionário usado para avaliar o *software*, as crianças acharam interessante o uso de *emoticons*. Foi possível identificar nas respostas que elas se sentiram motivadas a responder devido a facilidade de compreender as alternativas.

Uma das principais contribuições da pesquisa é mostrar, através dos resultados, que o *Software* Educacional Kodu Game Lab não causou efeitos negativos nas crianças e que os

designs não devem se preocupar apenas nas funcionalidades dos produtos, eles devem também levar em consideração a opinião dos usuários ao utilizar o produto.

Além disso, foi possível observar que o uso de *emoticons* foi um importante recurso para as crianças representarem seus sentimentos com base na ferramenta avaliada, isso se justifica pelo fato de ser uma forma atraente e menos cansativa de se trabalhar com criança.

Quanto as limitações do trabalho, um dos principais problemas encontrados para a realização da pesquisa foi o tempo para a aplicação que não conduzia com o prazo de entrega. Pois, para a aplicação do questionário seria necessário que as crianças já utilizassem a ferramenta Kodu Game Lab, e para tanto foi utilizado o público do projeto de extensão da UEPB que tinha como objetivo essa prática, com isso a aplicação do questionário ficou dependendo do cronograma do projeto.

Outro ponto que dificultou foi à quantidade reduzida de entrevistados, só foi possível trabalhar com 10 crianças, já que só foram elas que participaram do projeto e que conseqüentemente conheciam o Kodu Game Lab.

Como trabalhos futuros, sugere-se a aplicação do questionário em mais turmas para confirmar a boa avaliação do Kodu Game Lab e também a elaboração de um questionário padrão que servisse como modelo para avaliação de *software* educacional através do Design Emocional.

REFERÊNCIAS

ALLEN-CONN; KIM, Rose e. **Ideias poderosas para a sala de aula: usando Squeak para aprimorar a aprendizagem d Matemática e Ciências**. Califórnia, 2003. Disponível em: <<http://www.pensamentodigital.org.br/files/book.pdf>>. Acesso: 28 de Abril de 2016.

BARROS, Larissa. Disponível em: <http://www.b9.com.br/38376/tech/as-vantagens--da-tecnologia-no-ensino-das--criancas/>>. Acesso: 05 de maio de 2016.

BERKUN, S. *The role of flow in web design*. May, 2001. Disponível em: <<http://www.scottberkun.com/essays/13-the-role-of-flow-in-web-design/>>. Acesso em 20 abril de 2016.

CODE CLUB WORLD. Disponível em: < <https://codeclubworld.org/>>. Acesso: 29 de maio de 2016.

COOPER, S.; DANN, W.; PAUSCH, R. (2000). “Alice: A 3-D tool for introductory programming concepts”. *Journal of Computing Sciences in Small Colleges*, vol.15, n. 5, p.107-116.

CRISTOVÃO , H.M. E NOBRE, I.A.M. *Software educativo e objetivos de aprendizagem*. In NOBRE, I.A.M. (org). **Informática na Educação: Um caminho de possibilidades e desafios**. Serra (ES) Instituto Federal de Educação, Ciência e Tecnologia do Espírito Santo.

DAMÁSIO, Antônio. **O erro de Descartes**. São Paulo: Companhia das Letras, 1996

DETERS, J. I. SILVA, J. M.C.; MIRANDA, E M.; FERNANDES, A.M.R. (2008) “**Proposta de Diretrizes para Avaliação de Objetos de Aprendizagem Considerando Aspectos Pedagógicos e Técnicos**”. XIX Simpósio Brasileiro de informática na educação. http://sbie2008.virtual.ufc.br/CD_ROM_COMPLETO/workshops/workshop%20/O%20Desafio%20de%20Trabalhar%20com%20Alunos%20Repetentes%20na.pdf.

FALKEMBACH, G.A.M. O Lúdico e os Jogos Educacionais. Disponível:

<http://penta3.ufrgs.br/midiasedu/modulo13/etapa1/leituras/aros/Leitura_1.pdf> Acesso em: 20 de Setembro de 2016.

FRAGA, F., GEWERC, A. (2005). **Una experiência interdisciplinar em ed. primaria mediante el uso de Squeak. Santiago de Compostela: Innovación Educativa.** Acessado em 30 de Abril de 2016, em:

<http://campusvirtualunex.es/cala/editio/index.php?Journal=relatec&page=article&op=viewPDF&viewPDFInterstitial&path%5B%5D=283&path%5B%5D=267>.

GIL, Felipe. Videogame: Ame-o ou Deide-o. Disponível em:

<http://www.istoe.com.br/reportagens/6843_VIDEOGAME+AME+O+OU+DEIXE+O?pathI> Acesso em: 02 de setembro de 2016.

ISO/IEC, Normas ISO para Usabilidade, Disponível em:

<<http://homepages.dcc.ufmg.br/~clarindo/arquivos/disciplinas/eu/material/seminarios-alunos/normas-iso-kecia-elayne.pdf>> Acesso em 10 de Abril de 2016.

ISO 9126-JPG-Altura: 600 pixels.Largura 800 pixels.Time Color 24 bits kb. Formato JPEG. In Oliveira, Jurema.ISO/IEC 9126. São Paulo: 2009.

ISO 9241-10. Disponível em:

<<http://www.inf.ufsc.br/~cybis/ine5624/ISO9241parte10.pdf> >. Acesso em: 20 de Abril de 2016.

JORDAN, P. 1999b. Inclusive design. In: W.S. Green; P. W Jordan (eds). Human factors in product design: Current practice and future trends, Taylor & Francis, p. 171-181.

KENSKI, Vani Moreira. **Tecnologias e ensino presencial e a distância.** 4. ed. São Paulo: Papirus. 2006.

LOVE, D. *A Conversation With Linus Torvalds, Who Built The World's Most Robust Operating System And Gave It Away For Free.* Jun. 7, 2014. Disponível em:

<<http://www.businessinsider.com/linus-torvalds-qa-2014-6>>. Acesso em 12 de Abril de 2016.

MARTINS, J. S. **A natureza emocional da marca: construção de empresas ricas**. 6. ed. Ver. E atualizada. Rio de Janeiro: Elsevier, 2007.

MEIRA, L. Making sense of Instructional Devices : The emergence of Transparence in Mathematical Activity, **Journal for Research in Mathematics Education**, vol. 29, n. 2, 1998, pp. 121-142.

MEMÓRIA, F. **Design para Internet: projetando a experiência perfeita**. Rio de Janeiro: Elsevier, 2005.

MINAYO, M.C.S.; SOUZA, E.R. Métodos, técnicas e relações em triangulação. In: MINAYO, M.C.S.; ASSIS, S.G.; SOUZA, E.R. (Org.). **Avaliação por triangulação de métodos**. Rio de Janeiro: Fiocruz, 2005. p.71-103.

MONT'ALVÃO, Claudia.; DAMAZIO, Vera. (Orgs.). **Design, ergonomia e emoção**. Riode Janeiro: Mauad X. FAPERJ, 2008;

M.SAELI, J. PERRENETI, W. M.G. JOCHEMS, B.ZWANE-WANEVELD. A Pedagogical Content Knowledgein programming education for secondary school: Perspective. *In ICER'11 proceedings of the 7th international workshop on Computing Education Research*. Volume 10 (1): 145-146, 2011.

NICOLACI-DA-COSTA, A. M. (2001), “**Gerando conhecimentos sobre os homens, mulheres e crianças que usam computadores: algumas contribuições da psicologia clínica**”, Anais do IV Workshop sobre Fatores Humanos em Sistemas Computacionais (IBC'2001). Florianópolis, 120-131.

NORMAN, D. A; **Design Emocional: por que adoramos (ou detestamos) os objetos do dia a dia**. Rio de Janeiro: Rocco, 2008.

NORMAN, D. 2004. Emotional design: Wh we love (or hate) everyday trings. New York, Basic Books, 272p.

OBAMA, B. Don't Just Play on Your Phone, Program It. The White House Blog. 2013. Disponível em: <http://m.whitehouse.gov/blog/2013/12/09/don--t-just-play-your-phoneprogram-it>. Acesso em 31 de Março de 2016.

PAPERT, Seymour. **A máquina das Crianças: repensando a escola na era da informática**. Porto Alegre: Artmed, 2001.

PAPERT, Seymour. **Logo: computadores e educação**. São Paulo: Brasiliense, 1985.

PAUSCH, Randy and KELLEHER, Caintlin (2005). “**Lowering the Barriers to Programming: A Taxonomy of Programming Environments and Languages for Novice Programmers**”. In: ACM Computing Surveys, Vol. 37, No. 2, pp. 83-137.

RUSHKOFF, D. **As 10 questões essenciais da era digital**. São Paulo: Editora Saraiva, 2012.
SCRATCH. *ABOUT Scratch (Scratch Documentation Site)*. Disponível em: http://info.scratch.mit.edu/About_Scratch . Acesso: 28 de Abril de 2016.

SALES, C. G. AND DANTAS, V. F. (2010). **ProGame: um Jogo para o Ensino de Algoritmos e Programação**. Simpósio Brasileiro de Informática na Educação.

SANTOS, M. E. K. L.D; AMARAL, L.H. **Avaliação de Objetos Virtuais de Aprendizagem no ensino de Matemática**.

SCRATCH. Disponível em: <https://scratch.mit.edu>. Acesso : 26 de Maio de 2016.

SERRANO, F. Geração Geek. *Revista Exame Informática*. São Paulo: Editora Abril, 2014.

SETZER, V. W. Computadores na educação: por quê, quando e como. In: SETZER, V.W. **Meios eletrônicos e Educação: uma visão alternativa**, 2ª. ed. São Paulo: Escrituras, 2002, p. 85-34.

Disponível em: <http://www.ime.usp.br/~vwsetzer/PqQd.com.html>.
Acesso: 05 de maio de 2016.

SIENA, Osmar. Metodologia da pesquisa científica: elementos para a elaboração e apresentação de trabalhos acadêmicos. / Osmar Siena._Porto Velho: [s.n.], 2007, 200p.

SILVERMAN E YASMIN KAFAL. Scratch: programming for all. *Commun. ACM*, 52:60-67 November 2009.

SHELLY, GARY B., CASHMAN, THOMAS J., HEBERT, CHARLES W., *Alice 2.0: Introductory Concepts and Techniques*. SHELLY CASHMAN SERIES®, Course Technology, Cengage Learning, Boston Massachusetts, USA, 2007 (Printed in China).

TRAJANO, F. H. D, LINS, F. A. A, MELO, J. C B de (2010). “**A Informática a serviço do aprendizado**”. In: X Jornada de ensino, pesquisa e extensão –UFRPE, Recife/PE.

XAVIER, R. Uma abordagem híbrida para avaliação da experiência emocional de usuários, 2013.

ZOTOVICI, Andréa, MENEZES, Carlos Eduardo Dantas de, *Tutorial para lógica de programação usando ALICE*. Anais do Congresso Alice Brasil 2010. Disponível em: <http://www.mackenzie.com.br/fileadmin/Graduacao/EE/Eventos/Alice_Brasil/Arqs_2012/Alice_2010.pdf> Acesso em: 08 jun. 2016.

APENDICE A

Avaliação do Kodu Game Lab – Alunos

QUESTIONÁRIO

Idade:_____ Série/ano:_____

1. Qual seu gênero?

| | |
|--|--|
|  |  |
| <input type="checkbox"/> Feminino | <input type="checkbox"/> Masculino |

2. Você gosta de jogar videogames?

Sim Não

3. Você faz uso de jogos com frequência?

Sim Não

4. Quais os tipos de jogos que você mais utiliza?

- () Educativos
- () de tiros
- () de corrida
- () de raciocínio lógico
- () Diversos

5. O Kodu Game Lab é fácil de mexer?

| | | | | |
|---|---|---|--|---|
|  |  |  |  |  |
| Muito Difícil | Difícil | Mais ou menos Fácil | Pouco Difícil | Muito Fácil |

6. Você demorou aprender a utilizar o Kodu Game Lab?

| | | | | |
|---|---|---|--|---|
|  |  |  |  |  |
| Demorei muito | Demorei | Demorei mais ou menos | Demorei pouco | Não demorei nada |

7. Você conseguiu criar o jogo que queria?

| | | | | |
|---|---|---|--|---|
|  |  |  |  |  |
| Não consegui nada | Cosegui pouco | Conseguí mais ou menos | Conseguí | Conseguí tudo |

8. O que você sentiu quando conseguiu criar o jogo que desejava? (Aponte os emoticons).

| | | | | | | |
|---|---|---|---|--|---|---|
|  |  |  |  |  |  |  |
| Tédio, Chato | Nervoso | Insatisfeito | Raiva | Aliviado | Aborrecido, irritado | Confuso |
|  |  |  |  |  |  |  |
| Triste | Exausto | Pouco feliz | Bem | Apaixonante | Feliz | Muito Feliz |

9. Como você aponta a experiência de utilizar o Jogo?

| | | | | | | | | | |
|---|---|---|---|---|---|---|--|---|---|
|  |  |  |  |  |  |  |  |  |  |
| Tédio, Chato | Nervoso | Raiva | Aborrecido irritado | Triste | Exausto | Bem | Apaixona nte | Feliz | Muito Feliz |

10. Teve funcionalidades que você não conseguiu utilizar?

| | | | | |
|---|---|---|---|---|
|  |  |  |  |  |
| Muitas | Algumas | Apenas uma | Não | Não lembro |

11. Aponte os emotions que estão relacionados às suas emoções/sentimentos ao utilizar os recursos de navegação (menus, ícones, links e botões).

| | | | | | | | | | |
|---|---|---|---|---|---|---|--|---|---|
|  |  |  |  |  |  |  |  |  |  |
| Tédio, Chato | Nervoso | Raiva | Aborrecido irritado | Triste | Exausto | Bem | Apaixona nte | Feliz | Muito Feliz |

12. O que você achou da experiência de utilizar este questionário?