

**UNIVERSIDADE DE CAXIAS DO SUL**

**HENRIQUE PAGNO DE LIMA**

**DESENVOLVIMENTO DE UM *SERIOUS GAME* PARA A APRENDIZAGEM DO  
PENSAMENTO COMPUTACIONAL**

**CAXIAS DO SUL - RS**

**2017**



**HENRIQUE PAGNO DE LIMA**

**DESENVOLVIMENTO DE UM *SERIOUS GAME* PARA A APRENDIZAGEM DO  
PENSAMENTO COMPUTACIONAL**

Trabalho de Conclusão de Curso para  
obtenção do título de Bacharel em Ciência da  
Computação pela Universidade de Caxias do  
Sul, no Centro de Ciências Exatas e da  
Tecnologia.

Orientadora Profa. Elisa Boff.

**Caxias do Sul - RS**

**2017**



## RESUMO

O objetivo principal deste trabalho foi a modelagem, desenvolvimento e avaliação de um protótipo de *serious game* que auxilie crianças a desenvolverem lógica computacional. Os *serious games* tem se tornado cada vez mais populares nos dias de hoje. Isso ocorre principalmente porque os mesmos proporcionam uma forma divertida e engajante de aprendizagem, trabalhando como um auxílio de ensino para aqueles que o usufruem. A partir disso, foi selecionado o tema do pensamento computacional para agir como objetivo de aprendizagem do *serious game*, dada a sua importância como objeto de estudo nos dias de hoje, já que o conhecimento do mesmo é benéfico de várias formas. O *serious game* proposto limita-se a um público-alvo específico de seis a oito anos de idade, visando ensinar aos mesmos os conceitos básicos desta forma de pensamento. A seleção desta faixa etária baseou-se na Teoria Cognitiva de Piaget, que estuda a evolução cognitiva do ser humano ao longo do seu desenvolvimento. Através do estudo do *framework* DPE (*Design, Play, Experience*), pudemos identificar uma base eficiente para estruturar a análise do protótipo de *serious game* proposto, dividindo-o nas camadas apresentadas pelo *framework*. O *framework* é organizado de forma que o objetivo de ensino do *serious game* seja o foco na criação do mesmo, não permitindo que este seja tomado como segundo plano no projeto. Concluindo, o protótipo de *serious game* proposto neste trabalho é direcionado a plataformas *mobile* e objetiva-se em ensinar a crianças os conceitos básicos de lógica para computação de uma forma divertida e interessante.

**Palavras-chave:** *Serious games*. Pensamento computacional. *Framework* DPE. Lógica computacional.



## **ABSTRACT**

The main purpose of this paper was the modeling, development and evaluation of a prototype of serious game that helps children develop computational logic. Serious games have become more and more popular nowadays. That happens mainly because they provide a fun and engaging form of learning, working as a knowledge support to those who play it. From that, we selected computational thinking as the theme to act as the objective of learning of the serious game, given its importance as the object of study nowadays, since its knowledge is beneficial in many ways. The serious game proposed is limited to a specific target audience from six to eight years old, aiming to teach the basic concepts of this form of thinking. The selection of this age group is based on Piaget's theory of Cognitive Development, which studies the cognitive evolution of human beings during their development. Through the study of the DPE (Design, Play, Experience) framework, we were able to identify an efficient base to structure the analysis of the serious game prototype proposed, dividing it in the layers presented by the framework. The framework is organized in a form in which the objective of teaching of the serious game becomes the focus in its creation, not allowing that it becomes a background objective. In conclusion, the serious game prototype proposed in this paper is directed to a mobile platform and focus on the objective of teaching kids the basic concepts of computational logic in a fun and interesting way.

**Keywords:** Serious Games. Computational Thinking. DPE framework. Computational logic.



## LISTA DE ILUSTRAÇÕES

|   |    |
|---|----|
| Figura 1 – Diagrama do Processo de Adaptação .....                                      | 18 |
| Figura 2 – Esquema de Funcionamento da Zona de Desenvolvimento Proximal.....            | 21 |
| Figura 3 – Nove Eventos de Instrução de Gagné .....                                     | 27 |
| Figura 4 – <i>Screenshot</i> do jogo <i>CodeCombat</i> .....                            | 30 |
| Figura 5 – <i>Screenshot</i> de seleção do <i>CodinGame</i> .....                       | 31 |
| Figura 6 – <i>Screenshot</i> de jogo do <i>CodinGame</i> .....                          | 32 |
| Figura 7 – <i>Screenshot</i> do jogo <i>Robo Logic</i> .....                            | 33 |
| Figura 8 – <i>Screenshot</i> do Jogo <i>Lightbot</i> .....                              | 34 |
| Figura 9 – Componentes do <i>framework</i> MDA .....                                    | 37 |
| Figura 10 – Processo iterativo de <i>design</i> do <i>framework</i> DPE .....           | 38 |
| Figura 11 – Estrutura do <i>framework</i> DPE .....                                     | 39 |
| Figura 12 – Teoria de Fluxo de Mihaly Csikszentmihalyi .....                            | 42 |
| Figura 13 – <i>Screenshot</i> da tela de introdução do <i>serious game</i> .....        | 46 |
| Figura 14 – <i>Screenshot</i> do painel de final de nível do <i>serious game</i> .....  | 49 |
| Figura 15 – <i>Screenshot</i> de painel de instruções.....                              | 52 |
| Figura 16 – <i>Screenshot</i> da tela de jogo do <i>serious game</i> .....              | 54 |
| Figura 17 – <i>Screenshot</i> do menu do <i>serious game</i> .....                      | 55 |
| Figura 18 – Resultado de objetivo do <i>serious game</i> .....                          | 58 |
| Figura 19 – Resultados de jogabilidade do <i>serious game</i> .....                     | 59 |
| Figura 20 – Resultados de compreensão dos botões do <i>serious game</i> .....           | 59 |
| Figura 21 – Resultados do <i>loop</i> do <i>serious game</i> .....                      | 60 |
| Figura 22 – Resultados do procedimento do <i>serious game</i> .....                     | 60 |
| Figura 23 – Resultados de estética do <i>serious game</i> .....                         | 61 |
| Figura 24 – Resultados de desafio do <i>serious game</i> .....                          | 61 |
| Figura 25 – Resultados de usuários que continuariam jogando o <i>serious game</i> ..... | 62 |



## LISTA DE TABELAS

|  |    |
|--|----|
| Tabela 1 – Estágios de Desenvolvimento Cognitivo de Piaget ..... | 19 |
| Tabela 2 – Tipos de Regras em Jogos .....                        | 24 |
| Tabela 3 – Nove Eventos de Instrução de Gagné .....              | 28 |
| Tabela 4 – Conceitos Abordados no <i>CodeCombat</i> .....        | 31 |
| Tabela 5 – Habilidades do Pensamento Computacional .....         | 35 |
| Tabela 6 – Síntese dos Trabalhos Relacionados.....               | 36 |
| Tabela 7 – Botões de ação do jogo .....                          | 47 |
| Tabela 8 – Estrutura das matrizes do primeiro nível .....        | 50 |
| Tabela 9 – Estrutura das matrizes do segundo nível .....         | 50 |
| Tabela 10 – Estrutura das matrizes do terceiro nível.....        | 51 |
| Tabela 11 – Estrutura das matrizes do quarto nível.....          | 51 |



## LISTA DE SIGLAS

| <b>Sigla</b> | <b>Significado</b>                        |
|--------------|---|
| DPE          | <i>Design Play Experience</i>             |
| RPG          | <i>Role-playing Game</i>                  |
| MDA          | <i>Mechanics, Dynamics and Aesthetics</i> |
| PC           | <i>Personal Computer</i>                  |
| AR           | <i>Augmented Reality</i>                  |
| VR           | <i>Virtual Reality</i>                    |



## SUMÁRIO

|              |   |           |
|--------------|---|-----------|
| <b>1</b>     | <b>INTRODUÇÃO .....</b>   | <b>13</b> |
| 1.1          | PROBLEMA E QUESTÃO DE PESQUISA .....  | 13        |
| 1.2          | OBJETIVOS DO TRABALHO .....   | 14        |
| 1.3          | ESTRUTURA DO TRABALHO .....   | 14        |
| <b>2</b>     | <b>DESENVOLVIMENTO DO PENSAMENTO COMPUTACIONAL.....</b>                                     | <b>15</b> |
| 2.1          | TEORIA COGNITIVA DE PIAGET.....   | 16        |
| <b>2.1.1</b> | <b>ESQUEMAS.....</b>  | <b>17</b> |
| <b>2.1.2</b> | <b>PROCESSO DE ADAPTAÇÃO .....</b>  | <b>18</b> |
| <b>2.1.3</b> | <b>ESTÁGIOS DE DESENVOLVIMENTO COGNITIVO.....</b>   | <b>19</b> |
| 2.2          | ZONA DO DESENVOLVIMENTO PROXIMAL.....   | 20        |
| 2.3          | DEFINIÇÃO DA FAIXA ETÁRIA .....   | 22        |
| <b>3</b>     | <b><i>SERIOUS GAMES</i> .....</b>   | <b>23</b> |
| 3.1          | ENTRETENIMENTO DOS <i>SERIOUS GAMES</i> .....   | 25        |
| 3.2          | FATORES DE SUCESSO .....  | 25        |
| 3.3          | TRABALHOS RELACIONADOS .....  | 29        |
| <b>3.3.1</b> | <b><i>CODECOMBAT</i> .....</b>  | <b>29</b> |
| <b>3.3.2</b> | <b><i>CODINGAME</i> .....</b>   | <b>31</b> |
| <b>3.3.3</b> | <b><i>Robo Logic</i> .....</b>  | <b>32</b> |
| <b>3.3.4</b> | <b><i>Lightbot</i> .....</b>  | <b>33</b> |
| <b>3.3.5</b> | <b>CONSIDERAÇÕES SOBRE O CAPÍTULO .....</b>   | <b>34</b> |
| <b>4</b>     | <b>O PROCESSO DE DESENVOLVIMENTO DO <i>SERIOUS GAME</i> “AS AVENTURAS DE SALAZAR” .....</b> | <b>37</b> |
| 4.1          | METODOLOGIA.....  | 37        |
| <b>4.1.1</b> | <b>CAMADA DE APRENDIZAGEM .....</b>   | <b>39</b> |
| <b>4.1.2</b> | <b>CAMADA DE NARRATIVA .....</b>  | <b>40</b> |
| <b>4.1.3</b> | <b>CAMADA DE <i>GAMEPLAY</i>.....</b>   | <b>41</b> |

|              |  |           |
|--------------|--|-----------|
| <b>4.1.4</b> | <b>CAMADA DE EXPERIÊNCIA DO USUÁRIO .....</b>                      | <b>43</b> |
| <b>4.1.5</b> | <b>CAMADA DE TECNOLOGIA .....</b>                                  | <b>43</b> |
| <b>4.2</b>   | <b>DEFINIÇÃO DA TECNOLOGIA.....</b>                                | <b>43</b> |
| <b>4.3</b>   | <b>DEFINIÇÃO DO PROTÓTIPO DO <i>SERIOUS GAME</i> .....</b>         | <b>44</b> |
| <b>4.3.1</b> | <b>CAMADA DE APRENDIZAGEM.....</b>                                 | <b>44</b> |
| <b>4.3.2</b> | <b>CAMADA DE NARRATIVA .....</b>                                   | <b>46</b> |
| <b>4.3.3</b> | <b>CAMADA DE <i>GAMEPLAY</i> .....</b>                             | <b>47</b> |
| <b>4.3.4</b> | <b>CAMADA DE EXPERIÊNCIA DO USUÁRIO .....</b>                      | <b>52</b> |
| <b>5</b>     | <b>VALIDAÇÃO E RESULTADOS .....</b>                                | <b>57</b> |
| <b>5.1</b>   | <b>RESULTADOS DO <i>PLAYTESTING</i>.....</b>                       | <b>58</b> |
| <b>5.2</b>   | <b>AVALIAÇÃO DOS RESULTADOS DO <i>PLAYTESTING</i> .....</b>        | <b>63</b> |
| <b>6</b>     | <b>CONSIDERAÇÕES FINAIS .....</b>                                  | <b>65</b> |
| <b>6.1</b>   | <b>TRABALHOS FUTUROS .....</b>                                     | <b>67</b> |
|              | <b>REFERÊNCIAS BIBLIOGRÁFICAS .....</b>                            | <b>68</b> |
|              | <b>APÊNDICE A – QUESTIONÁRIO DE <i>PLAYTESTING</i>.....</b>        | <b>71</b> |
|              | <b>APÊNDICE B – RESPOSTAS COMPLETAS DE <i>PLAYTESTING</i>.....</b> | <b>77</b> |

## 1 INTRODUÇÃO

*Serious games* podem ser definidos como: “Jogos que não têm entretenimento, prazer, ou diversão como seu principal propósito” (LAAMARTI, EID e EL SADDIK, 2014) (tradução nossa).

Os *serious games* têm se tornado, ao longo dos últimos anos, cada vez mais atraentes. Em parte porque, com o avanço da tecnologia, os dispositivos eletrônicos se tornaram muito acessíveis à população como um todo. Além disso, pelos estímulos que os jogos geram aos usuários. A diversão que os games proporcionam advém do fato de aprender a superar um desafio, de dominar uma nova tarefa, de compreender um novo quebra-cabeça (FARDO, 2013). Jovens de todas as idades estão acostumados com jogos, sejam eles eletrônicos ou não, em seu dia a dia e durante seu crescimento. São raros aqueles que não tem um *smartphone* em mãos ou simples acesso a um computador. Isso proporciona uma oportunidade para o uso dos mesmos no auxílio do seu desenvolvimento acadêmico e pessoal, afinal, as novas gerações da sociedade possuem uma grande afinidade com a tecnologia. O que não se pode negar é a forte influência que esse contexto tecnológico exerce nas atividades humanas (FARDO, 2013).

Tornar a aprendizagem agradável e prazerosa é, sem dúvida, a melhor forma de estimulá-la. No entanto, é preciso avaliar a melhor forma na qual um *serious game* deve ser desenvolvido para o estímulo do conhecimento, não interferindo no seu real objetivo. Ou seja, como pode-se criar um ambiente de aprendizagem estimulante e eficiente.

### 1.1 PROBLEMA E QUESTÃO DE PESQUISA

O estudo de lógica para computação orientado para crianças é de muita utilidade para elas, uma vez que será aproveitado independente da área e profissão escolhida para cursar e exercer ao longo da vida adulta. Assim, o problema proposto é encontrar uma forma simples, divertida e engajante para o aprendizado e a prática desse tema com foco em crianças. Como se pode desenvolver um *serious game* com foco no desenvolvimento de lógica computacional que seja bem sucedido nesta tarefa?

## 1.2 OBJETIVOS DO TRABALHO

O objetivo principal do trabalho é a modelagem, desenvolvimento e avaliação de um *serious game* que auxilie crianças a desenvolverem lógica computacional.

Assim, foram definidos os seguintes sub-objetivos para alcançar o sucesso do objetivo mencionado acima:

- a) Conhecer os conceitos apresentados pela Teoria Cognitiva de Piaget;
- b) Conhecer os conceitos de pensamento computacional;
- c) Definir uma faixa etária para que o jogo criado se proponha a atender;
- d) Conhecer os conceitos de *serious games*;
- e) Pesquisar trabalhos relacionados;
- f) Analisar e desenvolver um protótipo de jogo para o auxílio do desenvolvimento de lógica computacional;
- g) Avaliar os resultados produzidos pelo protótipo de jogo criado.

## 1.3 ESTRUTURA DO TRABALHO

Este trabalho está estruturado da seguinte forma:

O segundo capítulo definiu uma faixa etária para abordagem, baseando-se na pesquisa de teorias do pensamento computacional, como a Teoria Cognitiva de Piaget e a Zona do Desenvolvimento Proximal de Vygotsky.

O terceiro capítulo buscou o estudo dos *serious games*, suas definições e diferenças em relação a jogos comerciais convencionais e aborda uma comparação em relação a alguns trabalhos relacionados.

O quarto capítulo define o protótipo de *serious game* que este trabalho visou criar e a tecnologia a ser utilizada, a partir das camadas definidas pelo *framework* DPE estudado.

O quinto capítulo apresenta uma explicação da montagem e análise dos resultados das etapas de testes realizadas, cujo objetivo é avaliar se o protótipo é bem sucedido como um *serious game*.

O sexto e último capítulo diz respeito às considerações finais levantadas em relação ao trabalho construído e o seu tema abordado, além de apontar alguns tópicos que podem ser abordados como trabalhos futuros.

## 2 DESENVOLVIMENTO DO PENSAMENTO COMPUTACIONAL

Para que possamos estudar o desenvolvimento do pensamento computacional, precisamos antes definir exatamente o que é esta forma de pensamento e quais são os seus benefícios em relação à sociedade e seus indivíduos.

Jeannette M. Wing (2006), professora presidente do departamento de Ciência da Computação na universidade de Carnegie Mellon, cita algumas definições para o pensamento computacional, sendo uma delas: “Pensamento computacional é o uso do raciocínio heurístico para descobrir uma solução. É planejar, aprender, e dimensionar na presença da incerteza.” (tradução nossa).

O pensamento computacional está diretamente relacionado ao desenvolvimento da capacidade de pensarmos em soluções para variados problemas, sejam eles acadêmicos ou não. A lógica que desenvolvemos pode ser utilizada para otimizar o tempo nas prateleiras do supermercado, para escolher qual caixa será o mais rápido, ou para compreendermos com mais facilidade conceitos acadêmicos não-relacionados à computação. Jonassen (2004) complementa este raciocínio, afirmando que aprender a resolver problemas é a habilidade mais importante que um estudante pode desenvolver, em qualquer situação.

Pensamento computacional é usar abstração e decomposição ao enfrentar uma tarefa grande e complexa ou projetar um sistema grande e complexo. É a separação de preocupações. É escolher uma representação apropriada para um problema ou modelar os aspectos relevantes de um problema para torná-lo tratável. É usar invariantes para descrever o comportamento de um sistema de forma sucinta e declarativa. É ter a confiança de que podemos, de forma segura, usar, modificar e influenciar um sistema grande e complexo sem compreender todos os seus detalhes. (M. WING, 2006) (tradução nossa).

Wing (2006) vai mais longe ainda, ao utilizar conceitos específicos do pensamento computacional para descrever sua utilidade em tarefas cuja relação com a computação não é diretamente aplicável. A abstração e decomposição de tarefas para a sua simplificação são métodos comuns na computação, que podem ser aplicados em qualquer atividade que tenha maior complexidade.

Ainda conforme Wing, o pensamento computacional é uma habilidade fundamental para todos, não apenas para cientistas computacionais. Além de leitura, escrita e aritmética, deveríamos adicionar o pensamento computacional para as habilidades analíticas de todas as crianças.

Introduzir o conhecimento do pensamento computacional no início dos estudos de jovens é introduzi-los aos benefícios desta forma de pensamento. A solução bem sucedida de problemas requer que estudantes gerem e testem soluções em sua mente antes de testá-las no mundo físico. (JONASSEN, 2004) (tradução nossa). O desenvolvimento da lógica pode auxiliar em todas as tarefas e desafios que sejam enfrentados ao longo do crescimento acadêmico e pessoal, uma vez que estes podem ser visualizados de forma mais linear, separando-os em pequenos passos para encontrar uma solução. Quando estudantes estão solucionando problemas, eles aprendem e compreendem mais. (JONASSEN, 2004) (tradução nossa). Humanos processam informações, humanos computam dados. Em outras palavras, o pensamento computacional não requer uma máquina (M. WING, 2008) (tradução nossa).

Os métodos de ensino deste pensamento, no entanto, não são consolidados. Ao menos, não para crianças. É importante que os estudantes possam compreender, de fato, os conceitos do pensamento computacional, e não apenas como aplicá-los em sua ferramenta predominante, o computador. Um bom exemplo disso é o uso da calculadora, que pode afetar a compreensão de cálculos caso seja introduzida durante os primórdios do ensino de aritmética. É discutível também, qual seria uma ordenação adequada para a aprendizagem dos conceitos por crianças. Aprendemos números aos cinco anos de idade, álgebra aos 12 anos, e cálculo a partir dos 18. Podem haver muitas formas de estruturar a progressão dos conceitos de pensamento computacional; qual é o método mais eficiente para cada tipo de estudante? (M. WING, 2008) (tradução nossa).

É importante ressaltar, no entanto, que crianças estão em constante aprendizagem, e o fazem de muitas formas diferentes. “O aprendizado ocorre de várias formas e fora de salas de aula: crianças ensinam entre si; aprendem dos pais e família; aprendem em casa, em museus e bibliotecas; e aprendem através de passatempos, navegando na *internet* e experiências de vida.” (M. WING, 2008) (tradução nossa).

## 2.1 TEORIA COGNITIVA DE PIAGET

Para que possamos definir uma faixa etária na qual o jogo em questão possa ser focado, precisamos compreender como os jovens, de diferentes idades, respondem à estímulos e novas experiências ao longo de seu desenvolvimento cognitivo. A Epistemologia Genética de Jean Piaget (1963), tenta esclarecer como o crescimento intelectual humano ocorre ao

longo dos primeiros anos de vida. Piaget discordava da ideia de que a inteligência é uma característica fixa, e considerava o desenvolvimento cognitivo como um processo que ocorre devido à maturação biológica e a interação com o ambiente (MCLEOD, 2015) (tradução nossa).

O que Piaget queria fazer não era medir o quão bem crianças podiam contar, pronunciar ou solucionar problemas como uma forma de avaliar seu Q.I. O que ele estava mais interessado era a forma na qual conceitos fundamentais como a própria ideia de números, tempo, quantidade, casualidade, justiça e outros emergiam. (MCLEOD, 2015) (tradução nossa).

Assim, Piaget mostrou que jovens pensam de formas diferentes quando comparados com adultos. Antes de sua pesquisa, acreditava-se que crianças eram apenas pensadores menos competentes que adultos. De acordo com Piaget, crianças nascem com uma estrutura mental muito básica (geneticamente herdada e evoluída) na qual se baseia todo o aprendizado e conhecimento subsequente (MCLEOD, 2015) (tradução nossa).

Existem três componentes básicos para a teoria cognitiva de Piaget:

- Esquemas;
- Processo de Adaptação;
- Estágios de Desenvolvimento Cognitivo.

### 2.1.1 ESQUEMAS

Piaget (1952) definiu os esquemas como “uma sequência de ações coesas e repetidas possuindo componentes firmemente interconectados e governados por uma razão principal” (tradução nossa). A partir de uma explicação simples, Piaget chamava de “esquema” os blocos básicos de comportamento inteligente – uma forma de organizar o conhecimento (MCLEOD, 2015) (tradução nossa). Acreditava-se que o desenvolvimento mental de uma pessoa implicava no aumento em número e complexidade dos esquemas que dita pessoa tinha conhecimento.

Um esquema também pode ser considerado como conexões mentais de representação do espaço, usado para entender e responder a situações. Essas representações são armazenadas e utilizadas conforme necessidade, durante atividades diárias.

Piaget acreditava que bebês recém nascidos tem um pequeno número de esquemas inatos – mesmo antes de terem muita oportunidade para experienciar o mundo. Estes

esquemas neonatais são a estrutura cognitiva subjacente aos reflexos inatos. Estes reflexos são geneticamente programados em nós. Por exemplo, bebês tem um reflexo de chupar, que é disparado por algo tocando os lábios do bebê. Um bebê irá chupar um mamilo, uma chupeta ou o dedo de uma pessoa. Piaget, portanto, assumiu que um bebê tem um ‘esquema de chupar’. (MCLEOD, 2015) (tradução nossa).

## 2.1.2 PROCESSO DE ADAPTAÇÃO

Piaget imaginava o crescimento intelectual como um processo de ajuste ao mundo. Esse processo ocorria por meio de três fases (conforme exibido na Figura 1): Assimilação, Acomodação e Equilíbrio.

Figura 1 – Diagrama do Processo de Adaptação



Fonte: MCLOED (2015), adaptado pelo autor.

O processo de assimilação diz respeito ao uso de um esquema já existente para lidar com um novo objeto ou situação.

O processo de acomodação ocorre quando os esquemas existentes não se aplicam à situação ou ao objeto e devem ser alterados.

O processo de equilíbrio gera o processo de aprendizado, uma vez que estamos sempre em busca de se adaptar a novos desafios (acomodação). Quando novas informações são adquiridas, o processo de assimilação com o novo esquema irá prosseguir até a próxima vez em que ajustes ao mesmo sejam necessários. Piaget acreditava que o desenvolvimento cognitivo não progredia de forma constante, mas sim em saltos esporádicos (MCLEOD, 2015) (tradução nossa).

### 2.1.3 ESTÁGIOS DE DESENVOLVIMENTO COGNITIVO

Piaget definiu quatro estágios de desenvolvimento cognitivo, que definem faixas nas quais as diferentes idades estão separadas, conforme a Tabela 1.

Tabela 1 – Estágios de Desenvolvimento Cognitivo de Piaget

| <b>Estágio</b>       | <b>Faixa de Idade</b> |
|----------------------|-----------------------|
| Sensório-motor       | 0 a 2 anos            |
| Pré-operacional      | 2 a 7 anos            |
| Operacional Concreto | 7 a 11 anos           |
| Operacional Formal   | 11 anos e acima       |

Fonte: Próprio Autor

Cada criança passa pelos estágios na mesma ordem e o seu desenvolvimento é determinado por maturação biológica e a sua interação com o meio ambiente (MCLEOD, 2015) (tradução nossa).

Durante o estágio Sensório-motor, a criança desenvolve a noção da existência de objetos, independentemente dos mesmos conseguirem enxergar o mesmo ou não.

No estágio Pré-operacional, crianças são capazes de pensar em coisas simbolicamente, permitindo que uma palavra seja utilizada para definir algo, além dela mesma. Neste estágio, o pensamento ainda é egocêntrico, tornando difícil para a criança visualizar o ponto de vista de outras pessoas.

O estágio Operacional Concreto é considerado por Piaget como um marco no desenvolvimento cognitivo infantil, visto que ele marca o início do pensamento lógico ou operacional (MCLEOD, 2015) (tradução nossa). Assim, neste estágio, a criança é capaz de raciocinar e aplicar soluções em sua cabeça, ao invés de precisar observá-las fisicamente no mundo real.

Quanto ao estágio Operacional Formal, o mesmo começa aproximadamente a partir dos 11 anos e dura até a idade adulta. Durante o mesmo, é desenvolvida a habilidade de pensar em conceitos abstratos e testar hipóteses logicamente.

## 2.2 ZONA DO DESENVOLVIMENTO PROXIMAL

Além da Teoria Cognitiva de Piaget, é oportuno abordar na fundamentação deste trabalho, a Zona do Desenvolvimento Proximal de Vygotsky, definida como:

a distância entre o nível de desenvolvimento real, que se costuma determinar por meio da solução independente de problemas, e o nível de desenvolvimento potencial, determinado por meio da solução de problemas sob a orientação de um adulto ou em colaboração com companheiros mais capazes. (VYGOTSKY, 1998)

Conforme Fardo (2003), o desenvolvimento real é aquele cujos processos de desenvolvimento encontram-se já consolidados. São atividades que são realizadas por um indivíduo sem auxílio de um terceiro, uma vez que a pessoa já é capaz de solucionar a necessidade por conta própria. A Figura 2 apresenta o esquema de funcionamento da teoria apresentada por Vygotsky.

Figura 2 – Esquema de Funcionamento da Zona de Desenvolvimento Proximal



Fonte: Huguet (adaptado por FARDO, 2013)

O desenvolvimento potencial caracteriza aquilo que um indivíduo é capaz de realizar somente com algum tipo de ajuda, advinda de outro mais capaz ou mais experiente. São atividades que um indivíduo ainda não consegue solucionar por conta própria, uma vez que ainda não tem o total conhecimento necessário para executá-la.

Vygostky defendia que, por meio desta abordagem, era possível melhor estimar o atual nível de desenvolvimento de uma criança, pois ele levava em consideração aquelas funções que, apesar de não estarem completamente desenvolvidas, encontravam-se presentes no indivíduo (FARDO, 2013).

Estes dois níveis distintos de desenvolvimento eram considerados dinâmicos, uma vez que em algum momento o indivíduo poderia não precisar mais de auxílio em algum processo, movendo-o assim para o desenvolvimento real.

Convém ressaltar, portanto, que um *serious game* pode ser utilizado como o “auxílio” mencionado durante o nível de desenvolvimento potencial, uma vez que este pode agir como o facilitador para a aprendizagem de um conceito, tema ou teoria. Ou seja, nesse caso, os games podem proporcionar a interação com o outro da perspectiva de Vygotsky (FARDO, 2013). Isso significa que, ao interagir com um *serious game*, o aluno recebe auxílio até o momento em que ele consegue resolver os problemas sem assistência (SEVERGNINI, 2016).

### 2.3 DEFINIÇÃO DA FAIXA ETÁRIA

Tendo em vista os diferentes estágios de desenvolvimento cognitivo apresentados por Piaget, é prudente que seja definida uma faixa etária com a qual se deseja trabalhar, uma vez que estas faixas etárias indicam como pode se dar o desenvolvimento cognitivo. Além disso, é interessante que a faixa etária escolhida permita uma introdução satisfatória ao desenvolvimento do pensamento computacional, pois é aconselhável que esta seja apresentada às crianças o quanto antes, em vista de seus diversos benefícios para o pensamento dos mesmos. “De fato, se quiséssemos garantir uma base comum e sólida de compreensão e aplicação de pensamento computacional para todos, então este aprendizado deveria ser melhor realizado nos primeiros anos de infância.” (M. WING, 2008) (tradução nossa).

Assim, para fins de delimitação deste projeto, definiu-se como faixa etária ideal o final do estágio Pré-operacional e o início do estágio Operacional Concreto, idade em que ocorre o início do pensamento lógico.

### 3 *SERIOUS GAMES*

Os *serious games* são, conforme mencionado na introdução deste trabalho, definidos como jogos cujo objetivo principal não é o entretenimento, e sim a aprendizagem.

Nos importamos com os *serious games* no sentido de que estes jogos têm uma finalidade educativa explícita e cuidadosamente pensada, não se destinando a serem jogados principalmente pelo entretenimento. Isso não significa que os *serious games* não são, ou não deveriam ser, divertidos. (C. ABT, 1987) (tradução nossa)

A maioria das definições encontradas tanto em pesquisa quanto na indústria concordam que *serious games* incluem uma dimensão de entretenimento (LAAMARTI, EID e EL SADDIK, 2014). Esta dimensão, no entanto, visa estimular a atenção e o foco daqueles que o usufruem.

Laamarti, Eid e El Saddik (2014) também observam que muitos jogos eletrônicos, que não são caracterizados como *serious games*, podem apresentar oportunidades de aumento de conhecimento ou habilidade em relação à alguma atividade ou assunto. Podemos exemplificar estes casos com jogos de estratégia, como a série *Total War*, ou o simulador de corridas *Gran Turismo*, que podem agregar conhecimentos de história e estratégia, ou desenvolver reflexos e habilidades que podem ser utilizados em atividades diárias. Jogos em geral também podem aprimorar o conhecimento e habilidades linguísticas dos usuários. Estudantes do idioma inglês, por exemplo, que usufruem de jogos, podem utilizá-los como um treinamento adicional de suas habilidades. Assim, é importante ressaltar que a diferença de um *serious game* para um jogo comercial da indústria está na ênfase que lhe é adicionado para o objetivo principal de desenvolvimento de algum conhecimento, e não para o entretenimento em si. Este conhecimento está relacionado ao conceito específico do *serious game*, como bem-estar, educação e saúde (LAAMARTI, EID e EL SADDIK, 2014).

Podemos concordar, no entanto, com Severgnini (2016), quando o mesmo menciona que desenvolver um *serious game* não é tarefa fácil. Para que o mesmo seja, de fato, excelente na proposta que visa atender, é necessária uma equipe de desenvolvimento capaz de analisar todas as arestas do jogo, desde a sua programação, animação, arte, som, narrativa e claro, a jogabilidade voltada à aprendizagem. Em 2005, como parte da pesquisa de seu livro “*Serious Games: Jogos que Educam, Treinam e Informam*” (tradução nossa<sup>1</sup>), David Michael e Sande Chen enviaram uma pesquisa para desenvolvedores, educadores, e pesquisadores envolvidos

---

<sup>1</sup> Serious Games: Games That Educate, Train and Inform.

ou interessados em *serious games*. Houve um total de 63 respondentes para a mesma. Na pesquisa, foi levantado que 26,23% (16) dos respondentes tinham um orçamento entre 100.000 a 500.000 dólares, enquanto dezessete (17) dos respondentes tinham um orçamento maior que este. Apenas 18,03% (11) dos respondentes tinham um orçamento menor que 5.000 dólares. Além disso, 77% (45) dos respondentes tinham uma equipe de 1 a 10 pessoas. Todas as outras respostas informavam equipes maiores que 10 pessoas.

Ao mencionarmos jogos em geral, podemos observar, conforme Facchin (2013), que apesar dos jogos terem características diferentes, há alguns elementos que são a base de criação de bons jogos.

Um jogo deve sempre ter, no mínimo, um objetivo. Estes objetivos são o que determinam a quantidade de jogadores que o *game* irá atingir, bem como o nível de satisfação alcançado sobre eles. (FACCHIN, 2013). São necessárias também, regras para definir quais atos o jogador poderá e deverá realizar durante o jogo. Segundo Salen e Zimmerman (2004), existem três tipos de regras que se aplicam a um jogo, conforme a Tabela 2.

Tabela 2 – Tipos de Regras em Jogos

| <b>Tipo</b>          | <b>Descrição</b>   |
|----------------------|--|
| Regras Operacionais  | Estas são o que normalmente pensamos de uma regra: as diretrizes que um jogador precisa para jogar. Estas regras normalmente são o equivalente às regras impressas em jogo de tabuleiro.   |
| Regras Constitutivas | As regras constitutivas de um jogo são as estruturas formais que existem “abaixo” das telas de um jogo. São estruturas lógicas e matemáticas.  |
| Regras Implícitas    | Regras implícitas são as regras não escritas. São regras de etiqueta, bom comportamento ou de bom espírito esportivo. Por exemplo: podemos deixar uma criança voltar uma jogada em uma partida de xadrez, mas provavelmente não permitiremos o mesmo em uma partida de campeonato. |

Fonte: Próprio Autor

Além disso, os jogos precisam ter desafios, que são inclusos no jogo de forma que, ao longo do progresso do jogador, sejam cada vez mais complexos. Dessa forma, o jogo permanecerá atrativo ao jogador, mantendo sua atenção constantemente.

A narrativa e a estética do jogo também são importantes para o seu sucesso. A narrativa dá ao jogador a oportunidade de se envolver com o jogo, tornando-o mais pessoal e significativo ao mesmo. A estética, por sua vez, diz respeito à parte visual do jogo, e tem enorme impacto na primeira percepção do jogador em relação ao jogo. Uma estética atraente tende a desenvolver interesse por parte dos jogadores, tornando-os propícios a conhecerem os outros atrativos e as características do jogo.

Mcgonigal (2012), por sua vez, cita quatro elementos que definem jogos: metas (objetivos), regras, sistema de *feedback* e participação voluntária. O sistema de *feedback* é definido como a característica que diz ao jogador o quão perto está de atingir os objetivos. A participação voluntária exige que os jogadores estejam predispostos a usufruir do jogo e de seus respectivos objetivos e regras.

### 3.1 ENTRETENIMENTO DOS *SERIOUS GAMES*

Apesar dos *serious games* terem seu propósito no intuito de agregar o conhecimento, o entretenimento é tão necessário para o mesmo quanto para jogos comerciais, uma vez que o seu propósito é utilizar as qualidades de um jogo para tornar a aprendizagem um processo agradável. “Em outras palavras, se o jogador não achar que o jogo é divertido, é improvável que ele o escolha para jogar novamente.” (MICHAEL e CHEN, 2005) (tradução nossa).

É preciso ter cuidado para não desviar o foco do *serious game*, porque por mais que o jogador se divirta ao interagir com um jogo sério, se ele não conseguir aprender o conteúdo pretendido, o *serious game* terá fracassado (SEVERGNINI, 2016). Devemos ser bastante rígidos nesta questão, pois o foco de um *serious game* sempre deve ser a aprendizagem. No entanto, o entretenimento também deve ser integrado de forma satisfatória, para que o *serious game* não perca a sua essência: um jogo. Este provavelmente é o maior desafio durante o desenvolvimento de um *serious game*, manter o equilíbrio certo entre a diversão e a aprendizagem.

### 3.2 FATORES DE SUCESSO

Laamarti, Eid e El Saddik (2014) elencam alguns pontos que julgam pertinentes para que um *serious game* seja bem sucedido.

É ressaltada a importância de guiar o usuário ao longo do jogo, especialmente no início do mesmo, de forma que eles tenham o conhecimento necessário para executá-lo sem que se sintam perdidos ou confusos.

Consequências negativas devem ser evitadas, uma vez que não encorajam o jogador a melhorar sua eficiência, e apenas causam a perda de interesse.

Os autores sugerem também que os jogos colaborativos, por sua vez, são mais interessantes aos jogadores do que jogos *single-player*. Assim, criar *serious games* que possam ser jogados de forma conjunta por mais de uma pessoa, pode ter um efeito motivacional positivo.

Em um estudo realizado em 2007 por Y. Inal e K. Cagiltay, “Experiências de fluxo de crianças em um ambiente de jogo social interativo” (tradução nossa<sup>2</sup>), foi verificado que apresentar desafios em um *serious game* é o elemento que mais impactou na imersão das crianças. É importante que o nível dos desafios não seja muito alto, o que pode impactar negativamente na motivação, nem muito baixo, o que causaria desinteresse. Manter um nível apropriado de desafios irá garantir o contínuo interesse dos jogadores.

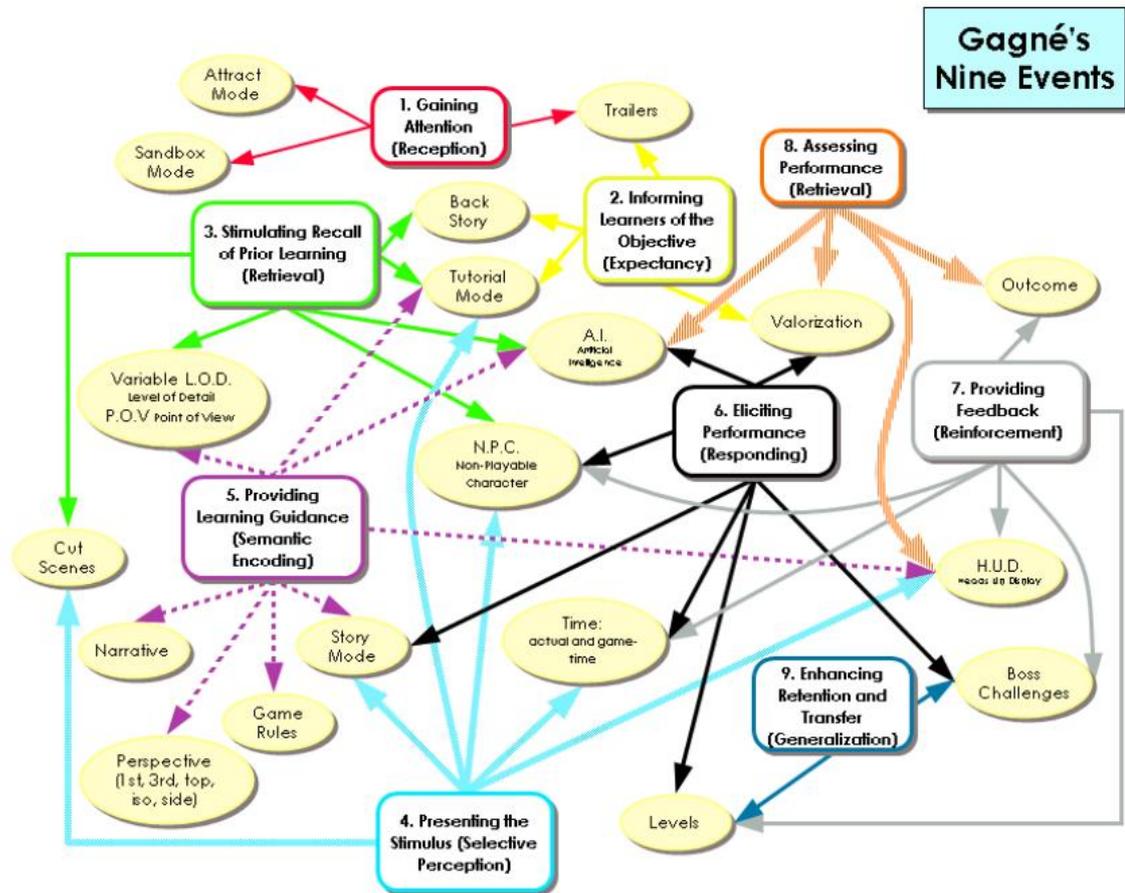
Abt (1987) cita também, em seu livro, que a implementação de pontuação em jogos é extraordinariamente importante. Os jogadores devem ser capazes de determinar a relativa eficiência de suas jogadas, quem venceu ou perdeu, e o que jogadas efetivas significam no processo específico sendo simulado no jogo. A pontuação também provém o que os psicólogos chamam de “*closure*” (ou seja, um senso de conclusão ou resolução) para a atividade, completando-a de uma forma psicologicamente satisfatória.

K. Becker (2008), cita que o foco em jogos educacionais aponta um modelo instrucional já utilizado em jogos comerciais. Esse modelo instrucional, *Gagné's Nine Events of Instruction* (Gagné, R. M., Briggs, L. J., & Wager, W. W., 1992), é apresentado conforme a Figura 3.

---

<sup>2</sup> Flow experiences of children in an interactive social game environment.

Figura 3 – Nove Eventos de Instrução de Gagné



Fonte: Becker (2008)

Segundo Gagné (1988), a instrução é um conjunto de eventos externos que podem apoiar e facilitar os processos de aprendizagem. Ela representa um conjunto de habilidades intelectuais ordenadas de tal forma que produziriam uma transferência positiva de uma capacidade mais simples para uma mais complexa (ZERBINI e ABBAD, 2008). Como a Figura 3 demonstra, elementos de jogos podem incorporar direta e indiretamente todos os elementos do modelo. Conectar elementos de jogos a componentes de um modelo instrucional serve para ilustrar quais elementos dos jogos podem ser utilizados para implementar abordagens instrucionais aprovadas. (BECKER, 2008) (tradução nossa). A Tabela 3 descreve brevemente a função de cada evento do modelo instrucional.

Tabela 3 – Nove Eventos de Instrução de Gagné

| Evento  | Descrição   |
|---|---|
| Adquirindo Atenção (Recepção)                                     | As sequências iniciais de um jogo devem adquirir e manter a atenção de um jogador, enquanto apresentam a estrutura do mesmo.                                    |
| Apresentando o Objetivo (Expectativa)                             | É repassado ao jogador o objetivo do jogo, enquanto provém informações necessárias para guiar o jogador sem causar frustração.                                  |
| Estimulando a Lembrança de Aprendizagens Anteriores (Recuperação) | O progresso ao longo de um <i>serious game</i> deve estimular os aprendizados anteriores, conforme a complexidade do mesmo aumenta.                             |
| Utilizando Estímulos (Percepção Seletiva)                         | Estímulos visuais são úteis para manter a atenção e motivação dos jogadores, por isso é interessante que estes estímulos sejam reforçados ao longo do jogo.     |
| Provendo Auxílio para o Aprendizado (Codificação Semântica)       | Deve-se prover dicas para auxiliar o jogador em seu progresso ao longo do jogo, auxiliando-o em possíveis situações que possam causar muita dificuldade.        |
| Elicitando Performance (Respondendo)                              | É importante que um <i>serious game</i> auxilie o jogador e aponte-o no caminho certo para a conclusão de tarefas do jogo, otimizando sua performance no mesmo. |
| Provendo <i>Feedback</i> (Reforço)                                | Informar ao jogador como está seu progresso no jogo e como o mesmo pode melhorá-lo é uma boa forma de mantê-lo interessado no jogo.                             |
| Estimando Performance (Recuperação)                               | Estimar a performance diz respeito a permitir que o jogador possa cometer   |

|   |  |
|---|--|
|   | erros sem que estes sejam extremamente punitivos. Por exemplo, o jogador não pode ser obrigado a começar o jogo inteiro do início caso falhe em alguma etapa.  |
| Aprimorando Retenção e Transferência<br>(Generalização) | É importante que o jogador de um <i>serious game</i> retenha as informações aprendidas durante a utilização do jogo, de forma que o mesmo possa tirar proveito deste conhecimento e transferi-lo para novas tarefas. |

Fonte: Becker (2008), adaptado pelo autor.

### 3.3 TRABALHOS RELACIONADOS

O estudo e pesquisa destes trabalhos relacionados têm por objetivo conhecer os *serious games* que já foram projetados, de forma que possamos entender suas características, seus pontos fortes e a forma como são estruturados.

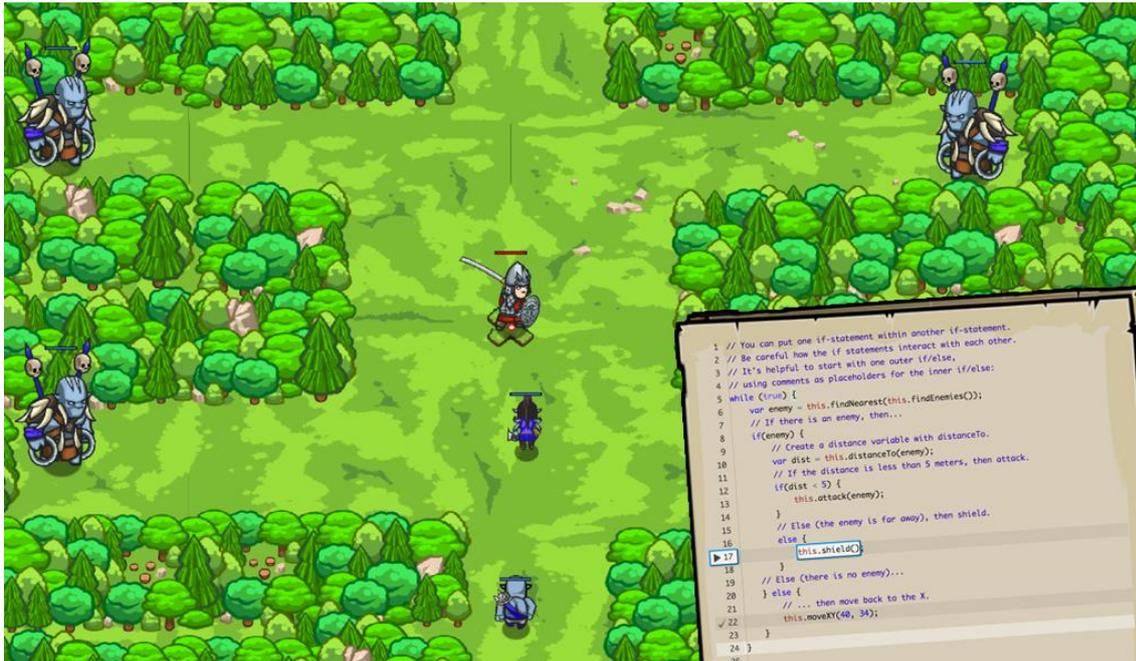
Uma vez que podemos incluir simuladores como *serious games*, existem muitos jogos relacionados que podemos considerar para avaliação. No entanto, como este trabalho visa a aprendizagem de lógica para computação, vamos focar nos *serious games* que também estão voltados para este estudo em específico.

#### 3.3.1 CODECOMBAT

“*CodeCombat* é uma plataforma para estudantes aprenderem ciência da computação através de um jogo real.” (CODECOMBAT, 2013). O mesmo é classificado como um *serious game* de RPG *multiplayer*, cujo propósito é a aprendizagem de programação para crianças com idade acima de nove anos, mesmo que não tenham conhecimento algum de programação. O jogo já tem, atualmente, mais de 5 milhões de jogadores, e disponibiliza várias linguagens de programação como opção. Além disso, conta com uma comunidade *open-source*, que contribui para o crescimento do jogo. Uma imagem do jogo pode ser visualizada na Figura 4.

Durante o jogo, o jogador enfrenta diversos desafios de lógica, dispondo apenas de comandos específicos para cumprir seus objetivos. Em um dos níveis, por exemplo, o personagem deve coletar todas as gemas espalhadas pelo cenário, além de derrotar o inimigo que as vigia. Para conseguir realizar esta tarefa, o jogador deve mover seu personagem pelo mapa de modo a coletar as gemas e comandar um ataque contra o inimigo. Traduzindo para a linguagem do jogo, isso significa que o jogador deve ordenar seu personagem com comandos de movimentação (*move up*, *move right*, *move left* e *move down*) e de ataque (*attack*), descrevendo um algoritmo que resolve o problema. (SEVERGNINI, 2016).

Figura 4 – Screenshot do jogo *CodeCombat*



Fonte: <http://br.codecombat.com/about>

Tratando-se da aprendizagem, o jogo é dividido em quatro capítulos: *Kithgard Dungeon*, *Backwoods Forest*, *Sarven Desert*, *Cloudrip Mountain*. Em cada um dos capítulos, o jogador aprende alguns conceitos da programação, conforme descrito na Tabela 4.

Tabela 4 – Conceitos Abordados no *CodeCombat*

| <b>Kithgard Dungeon</b> | <b>Backwoods Forest</b> | <b>Sarven Desert</b>  | <b>Cloudrip Mountain</b> |
|-------------------------|-------------------------|-----------------------|--------------------------|
| Sintaxe básica          | Comando SE              | Aritmética            | Objetos Literais         |
| Métodos                 | Operadores Lógicos      | Laços WHILE           | Laços FOR                |
| Parâmetros              | Propriedades de Objetos | Comando BREAK         | Funções                  |
| Strings                 | Leitura de dados        | Vetores               | Desenhos                 |
| Laços de Repetição      |                         | Comparação de strings | Módulo                   |
| Variáveis               |                         | Encontrar Min./Máx.   |                          |

Fonte: <http://br.codecombat.com/teachers>, adaptado por Severgnini (2016).

### 3.3.2 CODINGAME

O *CodinGame* é uma plataforma muito semelhante ao *CodeCombat*. O jogador deve programar soluções para problemas introduzidos dentro de um jogo. Cada jogo é apresentado com uma história e com um problema de programação, onde esta programação efetuada pelo jogador é pontuada baseada na eficiência de sua solução. Existem vários desafios, nos quais cada um deles apresenta um nível de dificuldade. Conforme o jogador avança, são apresentados desafios com conceitos mais complexos de programação.

Ao selecionar um desafio, é apresentado ao jogador um resumo da tarefa (conforme a Figura 5), bem como qual é o conceito que será trabalhado.

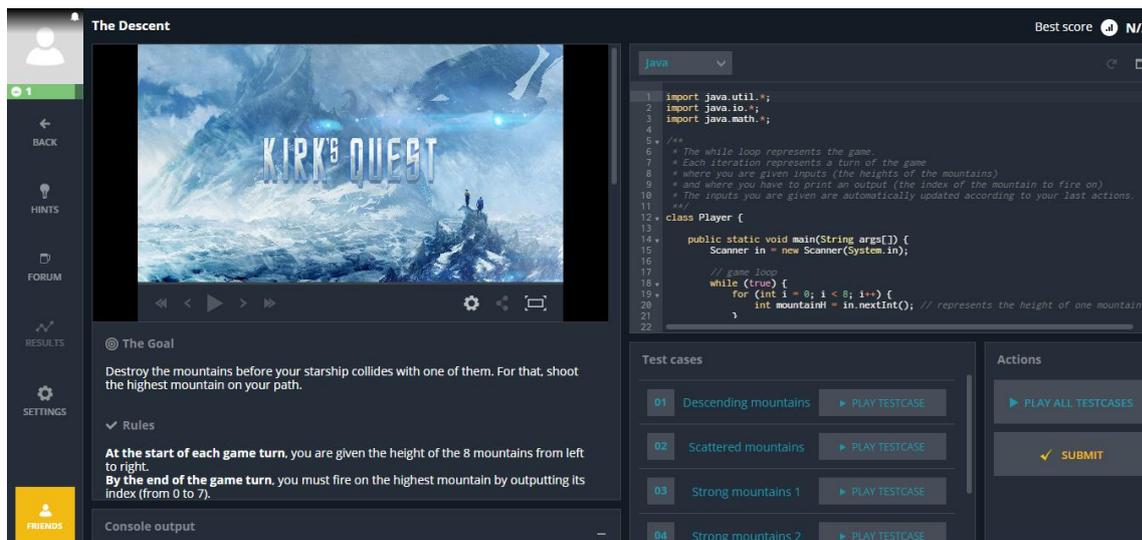
Figura 5 – Screenshot de seleção do *CodinGame*

The screenshot displays the 'THE DESCENT' puzzle page on CodinGame. At the top, it shows the breadcrumb 'PRACTICE > CLASSIC PUZZLE - EASY >' and the puzzle title 'THE DESCENT'. Below the title, it indicates 'Difficulty: Easy' and 'Community success rate: 61%'. A yellow 'SOLVE IT' button is visible on the right. The navigation bar includes 'DETAILS', 'DISCUSSIONS', 'SOLUTIONS', and 'SHARE'. The main content area is divided into three columns: 'WHAT WILL I LEARN?' (with a sub-section 'Loops' and a description of learning loops), 'MY PROGRESS' (showing 0% completion), and 'ACHIEVEMENTS' (showing 0/2 achievements). The 'STATEMENT' section at the bottom describes the puzzle: 'A simple problem to try out the CodinGame platform: your program must find the highest mountain out of a list of mountains.'

Fonte: <https://www.codingame.com/training/easy/the-descent>

Entrando no desafio, o jogo abre uma nova tela (conforme a Figura 6), permitindo que o jogador visualize a tela de programação, onde o mesmo deve desenvolver a solução, podendo escolher entre inúmeras opções de linguagens. É informado ao usuário o objetivo do desafio, assim como as regras, e as variáveis de entrada. Além disso, o jogo dispõe de uma tela onde o usuário pode verificar, em tempo real, a execução dos casos de teste. Caso todos os casos de teste sejam executados com sucesso, o jogador passa do nível com êxito.

Figura 6 – Screenshot de jogo do *CodinGame*



Fonte: <https://www.codingame.com/ide/puzzle/the-descent>

### 3.3.3 Robo Logic

O *Robo Logic* é um jogo *mobile* onde o jogador deve mover um robô através de um cenário, ativando caixas específicas do trajeto para completar cada nível. O trajeto, cujo robô irá executar, é definido pelo jogador, no qual o mesmo monta um algoritmo, informando a ordem na qual o personagem do jogo deve efetuar as ações disponíveis para alcançar o objetivo proposto. Você deve programar os movimentos do robô arrastando comandos para a memória do mesmo. (DIGITALSIRUP, 2009) (tradução nossa). A Figura 7, uma imagem do jogo, demonstra a forma como o mesmo é estruturado. No topo, temos a imagem do jogo, que mostra o cenário e o robô. Abaixo, temos os comandos disponíveis, que são utilizados para programar as ações do robô. Por fim, temos as pilhas de execução das ações do robô, onde fica destacado as ações que serão realizadas, além da ordem das mesmas.

Figura 7 – Screenshot do jogo *Robo Logic*

Fonte: <https://www.digitalsirup.com/app/robologic/?lang=en>

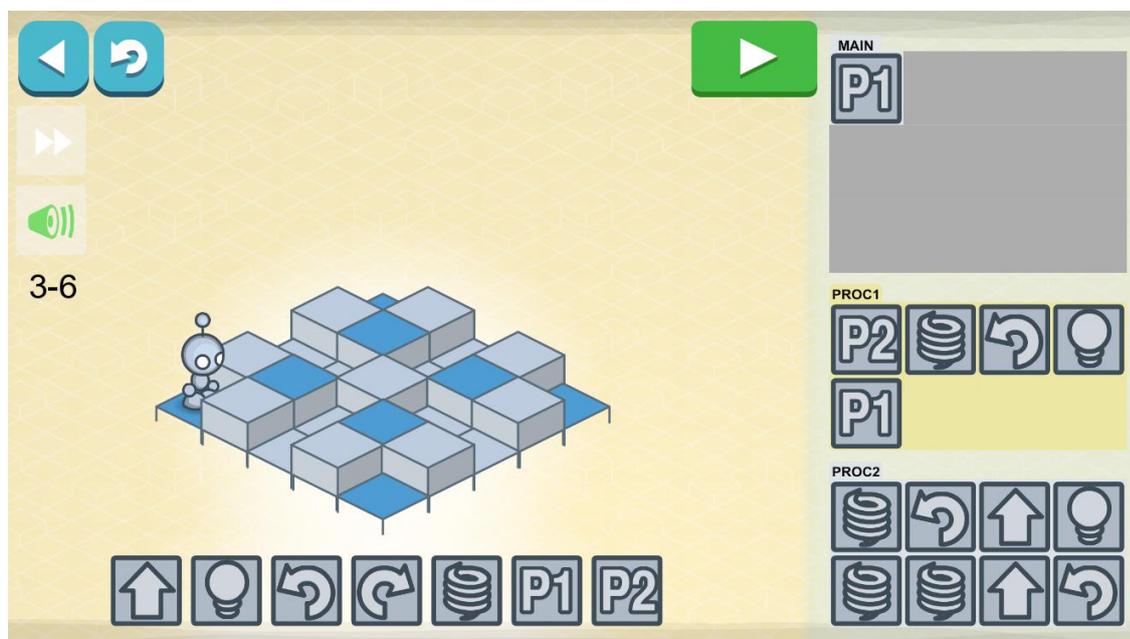
Logo o jogador perceberá, no entanto, que utilizar apenas a memória principal não é o bastante, e precisará criar e chamar funções para atingir os objetivos. (DIGITALSIRUP, 2009) (tradução nossa). As memórias “F1” e “F2” do jogo são utilizadas para prover ao jogador o conceito de funções e procedimentos, nas quais o jogador pode programar sub-rotinas que podem ser chamadas na memória principal ou até mesmo entre elas, apresentando assim o conceito de *loops*. *Robo Logic* é muito fácil de aprender, mas difícil de masterizar. (DIGITALSIRUP, 2009) (tradução nossa).

### 3.3.4 *Lightbot*

O *Lightbot* é, assim como o *Robo Logic*, um jogo *mobile* (embora o *Lightbot* também esteja disponível em outras plataformas); são muito semelhantes. Seu objetivo, regras, e modo de jogo são os mesmos. O *Lightbot*, no entanto, se destaca esteticamente. Seus modelos e animações são mais bem trabalhados, em relação ao *Robo Logic*, como podemos ver na Figura 8. O *Lightbot* apresenta as pilhas de execução das ações à direita, na seguinte ordem: Pilha principal; procedimento 1; procedimento 2. Na parte esquerda da tela, é possível ver o

personagem e o cenário do jogo. Abaixo do cenário, vemos as opções de ações disponíveis para serem utilizadas, sendo elas, respectivamente: Caminhar; ligar; girar para a esquerda; girar para a direita; pular; chamar pilha do procedimento 1; chamar pilha do procedimento 2.

Figura 8 – Screenshot do Jogo *Lightbot*



Fonte: *Lightbot*

### 3.3.5 CONSIDERAÇÕES SOBRE O CAPÍTULO

A partir da avaliação dos trabalhos relacionados, podemos fazer uma comparação entre os mesmos, visando a melhor compreensão de suas características, semelhanças e diferenças. A Tabela 5, proposta por COSTA et al. (2015), descreve as habilidades do pensamento computacional que podem ser aprofundadas.

Tabela 5 – Habilidades do Pensamento Computacional

| <b>Habilidade</b>        | <b>Descrição</b>  | <b>Exemplo</b>   |
|--------------------------|---|--|
| Lógica condicional       | Consiste na habilidade de compreender as consequências dos valores verdadeiro e falso. Usualmente através da utilização da construção se-então-senão. | Dados 3 números x, y e z. Se x é maior que y e y é maior que z. Então, sabemos que x é o maior. Senão, o maior valor será y ou z. (tabela verdade).  |
| Construção de algoritmos | Consiste na habilidade de resolver um determinado problema, utilizando um conjunto de lógicas condicionais em uma abordagem passo a passo.            | Dados 3 números x, y e z. Primeiro verifica-se se x é maior que y e z. Se for verdade, então x é o maior. Senão, preciso verificar se y é maior que z. Se for verdade então y é maior. Se não z é o maior.   |
| Depuração                | Consiste no ato de encontrar erros lógicos em um algoritmo que não funciona como esperado.  | O algoritmo pode se não comportar como esperado para uma determinada entrada. Em sala de aula é explorada através de atividades que envolvam encontrar um erro dentro de um algoritmo.   |
| Simulação                | Consiste no ato de modelar ou testar um algoritmo. São usadas tanto para a depuração quanto para a construção de algoritmos.                          | Dado um algoritmo que retorna o maior valor, podemos simular o algoritmo para diferentes combinações de valores de x, y e z. Em sala de aula é explorada através de teste de mesa e execuções passo a passo.   |
| Socialização             | Refere aos aspectos sociais do pensamento computacional. Onde a solução de um problema pode ser alcançada e compartilhada entre uma ou mais pessoas.  | A solução de um problema pode ser alcançada pela divisão e distribuição do problema para uma ou mais pessoas. Dojo de programação é um exemplo de metodologia que enfatiza essa habilidade (SATO et al., 2008 apud COSTA, SOUZA, et al., 2015). Ela enfatiza a programação em pares que codificam e compartilham com todos as suas soluções. |

Fonte: COSTA, SOUZA, et al. (2015)

A Tabela 6 fornece uma síntese das informações dos trabalhos relacionados elencados neste capítulo, com base nas habilidades descritas na Tabela 5.

Tabela 6 – Síntese dos Trabalhos Relacionados

| Nome       | Gênero         | Plataforma        | Habilidades   | Forma de Iteração | Faixa Etária |
|------------|----------------|-------------------|---|-------------------|--------------|
| CodeCombat | Luta           | Web               | Lógica condicional, Construção de algoritmos, Depuração, Socialização | Texto             | 10+          |
| CodinGame  | Aventura       | Web               | Lógica condicional, Construção de algoritmos, Depuração, Socialização | Texto             | 10+          |
| Robo Logic | Quebra-cabeças | iOS               | Construção de Algoritmos, Simulação                                   | Visual            | 5 a 8 anos   |
| Lightbot   | Quebra-cabeças | Web, iOS, Android | Construção de Algoritmos, Simulação                                   | Visual            | 5 a 8 anos   |

Fonte: Próprio Autor

Assim, podemos concluir que o *Robo Logic* e o *Lightbot* são, dos trabalhos relacionados mencionados, aqueles que mais se assemelham ao protótipo que este trabalho visa projetar. O *CodeCombat* e o *CodinGame* são projetos muito mais complexos, que permitem que os jogadores tenham acesso ao desenvolvimento de uma gama mais completa de habilidades do pensamento computacional, uma vez que trabalham diretamente com programação. Estes, no entanto, não são adequados a iniciantes do pensamento computacional, uma vez que sua interface não permite uma aprendizagem de conceitos de uma forma mais simples.

Portanto, é importante ressaltar que o projeto do protótipo deste trabalho poderá se basear nos conceitos, técnicas e estética utilizados no *Robo Logic* e *Lightbot*.

## 4 O PROCESSO DE DESENVOLVIMENTO DO *SERIOUS GAME* “AS AVENTURAS DE SALAZAR”

O capítulo de desenvolvimento deste trabalho tem como objetivo definir a metodologia e as tecnologias que foram utilizadas durante o desenvolvimento do protótipo, bem como apresentar uma definição de forma de aprendizagem, de estrutura interna e de estética do jogo.

### 4.1 METODOLOGIA

O *framework* MDA (*Mechanics, Dynamics and Aesthetics*) é uma abordagem formal para a compreensão dos jogos, que visa conectar a lacuna entre o *design*, desenvolvimento, críticos e equipe de pesquisa de jogos. (HUNICKE, LEBLANC e ZUBEK, 2004) (tradução nossa). Este *framework* formaliza o consumo de jogos, dividindo-os em seus componentes distintos e estabelecendo as partes de seu projeto, conforme a Figura 9.

Figura 9 – Componentes do *framework* MDA



Fonte: Hunicke, Leblanc e Zubek (2004), adaptado pelo autor.

A mecânica (*mechanics*) descreve os componentes específicos do jogo, no nível da representação de dados e algoritmos.

A dinâmica (*dynamics*) descreve, em tempo real, o comportamento das mecânicas sobre as ações do jogador e as suas saídas equivalentes.

A estética (*Aesthetics*) descreve as respostas emocionais que desejamos causar no jogador quando o mesmo interage com o jogo.

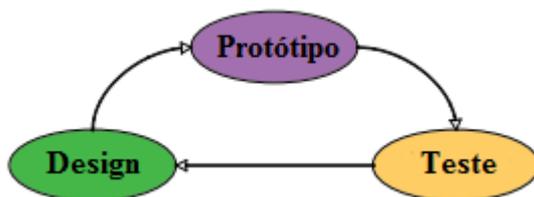
Da perspectiva do *designer* do jogo, as mecânicas dão suporte às dinâmicas do comportamento do sistema, o que por sua vez leva para as experiências da estética. Da perspectiva do jogador, a estética define o jogo, e é baseada a partir das dinâmicas observadas e, eventualmente, das mecânicas operadas. (HUNICKE, LEBLANC e ZUBEK, 2004) (tradução nossa).

O *framework* MDA, no entanto, não será a metodologia adotada neste trabalho. Esse método não considera especificamente nenhum aspecto além da jogabilidade, não englobando as características únicas dos *serious games*. (SEVERGNINI, 2016). Assim, passamos ao estudo do *framework* DPE (*Design, Play, Experience*), apresentado por Winn (2008), cujo objetivo é ser uma expansão do MDA focado às necessidades dos *serious games*.

Similar ao *framework* MDA, o DPE retrata a relação entre o *designer* e o jogador. O *designer* faz o jogo, o jogador o joga, o que resulta na experiência do jogador. (WINN, 2008) (tradução nossa).

Conforme Winn (2008), para projetar um jogo de forma eficiente, o *designer* deve antes definir os objetivos para a experiência que se deseja alcançar. Estes objetivos podem ser usados tanto para guiar o projeto quanto para medir a eficiência do mesmo quando implementado. Assim, conforme a Figura 10, isso irá refletir no processo de *design* do jogo, incluindo o projeto, prototipação, testes do jogo e voltando ao projeto do jogo, baseando-se na experiência retratada pelos testes.

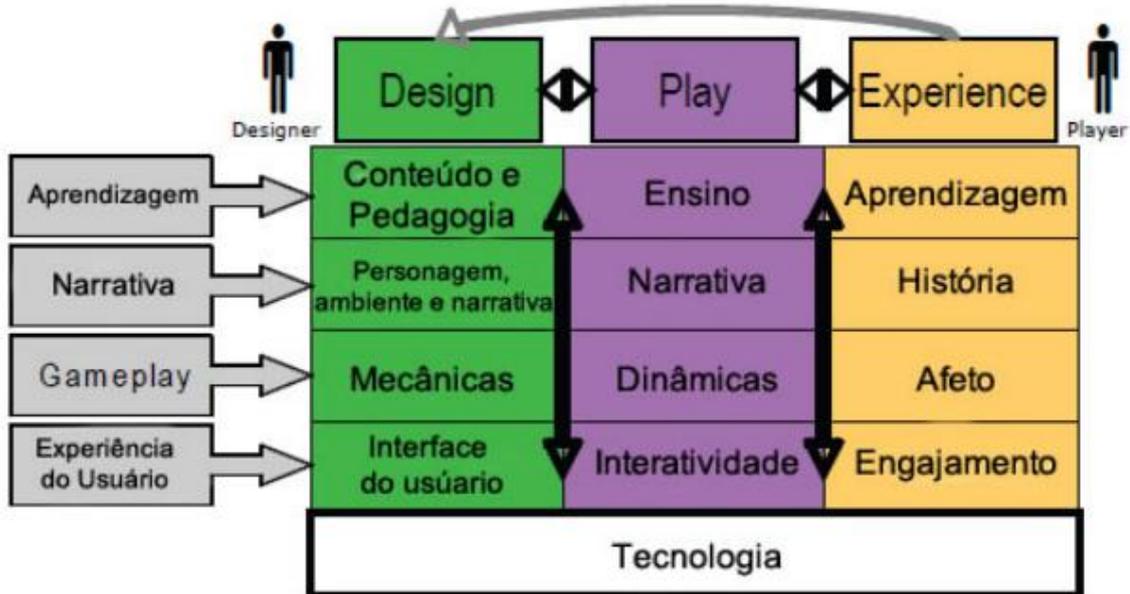
Figura 10 – Processo iterativo de *design* do *framework* DPE



Fonte: Winn (2008), adaptado pelo autor.

Ainda conforme Winn (2008), é necessário termos ciência de que jogar é uma experiência mediada. Esta experiência é muito influenciada pelo jogador, sua capacidade de cognição, sociabilidade, cultura e suas experiências de vida. Assim, a experiência de um jogador pode ser profundamente diferente da experiência de outros jogadores. Portanto, o público alvo para o jogo deve ser considerado durante o processo de projeto do mesmo.

A Figura 11 descreve os subcomponentes de *design* de um *serious game* e sua conexão com os componentes principais, incluindo a aprendizagem, narrativa, *gameplay* e experiência do usuário. A tecnologia é mencionada abaixo da estrutura, uma vez que o projeto do jogo é realizado na mesma.

Figura 11 – Estrutura do *framework* DPE

Fonte: Winn (2008), adaptado por Severgnini (2016).

Os subcomponentes mencionados serão descritos de forma mais específica nas seções seguintes.

O *framework* também dá uma estrutura para decompor os elementos de um projeto. Podem ser avaliados e discutidos, separadamente, seu aprendizado, narrativa, jogabilidade, experiência do usuário e tecnologia. Podem ser avaliados quais eram os objetivos específicos do jogo, e assim determinar se os objetivos foram alcançados por meio do projeto.

#### 4.1.1 CAMADA DE APRENDIZAGEM

A camada de aprendizagem diz respeito ao conteúdo e à pedagogia, visando resultar na aprendizagem do jogador enquanto o mesmo joga o jogo. O conteúdo e a pedagogia do jogo devem ser projetados com o intuito de alcançar um objetivo de aprendizagem resultante da experiência do jogo. É importante definir, no início do projeto, os objetivos de aprendizagem que serão introduzidos no jogo. Estes objetivos também formam a base para a avaliação da eficiência de ensino que o jogo provém aos jogadores. Desta forma, o *designer*

do jogo deve definir os objetivos da aprendizagem para que, posteriormente, possa elaborar os conteúdos para atender a esses objetivos. (SEVERGNINI, 2016).

Winn (2008) sugere a Taxonomia de Bloom para Ensinar e Aprender<sup>3</sup> (1956) como recurso para estruturar e gerar os resultados de aprendizagem de um *serious game*. Este trabalho, no entanto, cita a Zona de Desenvolvimento Proximal de Vygostky e a Teoria Cognitiva de Piaget (Capítulo 2) como teorias dominantes para a estrutura da aprendizagem do tema em questão.

#### 4.1.2 CAMADA DE NARRATIVA

Existem, conforme Rouse (2001 apud WINN, 2008), duas perspectivas de narrativa em jogos: a narrativa do *designer* e a do jogador. A narrativa do *designer* é aquela projetada dentro do jogo. Ela pode ser utilizada para preparar o cenário; dar propósito e engajar; prover conteúdo; além de outras utilidades.

Segundo Winn (2008), a narrativa que ocorre durante o jogo, combina narrativa do *designer* com as interações e escolhas feitas pelo jogador. A experiência resultante disso gera o conceito da narrativa do jogador. Alguns jogos têm narrativas de *designer* muito complexas e trabalhadas, outros muito simples, e alguns não têm narrativa alguma. No entanto, todos os jogos têm uma narrativa do jogador, que no mínimo refletem a história dos desafios do jogo e como o jogador interpreta e age diante das mesmas. Ao iniciar o projeto, o *designer* precisa antes decidir que tipo de histórias ele ou ela deseja que o jogador possa experimentar, projetando o cenário, o personagem e a narrativa de forma a alcançar isso. (WINN, 2008) (tradução nossa).

Winn (2008) ressalta também que a camada de aprendizagem frequentemente complica a camada de narrativa em um *serious game*. Por exemplo, se estamos desenvolvendo um *serious game* para ensinar ciência, a narrativa pode integrar elementos de ficção científica? Dessa forma, estas decisões dos elementos da narrativa devem ser realizadas tendo em vista a camada de aprendizagem, de forma a não prejudicá-la.

---

<sup>3</sup> Bloom's Taxonomy on Teaching and Learning.

### 4.1.3 CAMADA DE *GAMEPLAY*

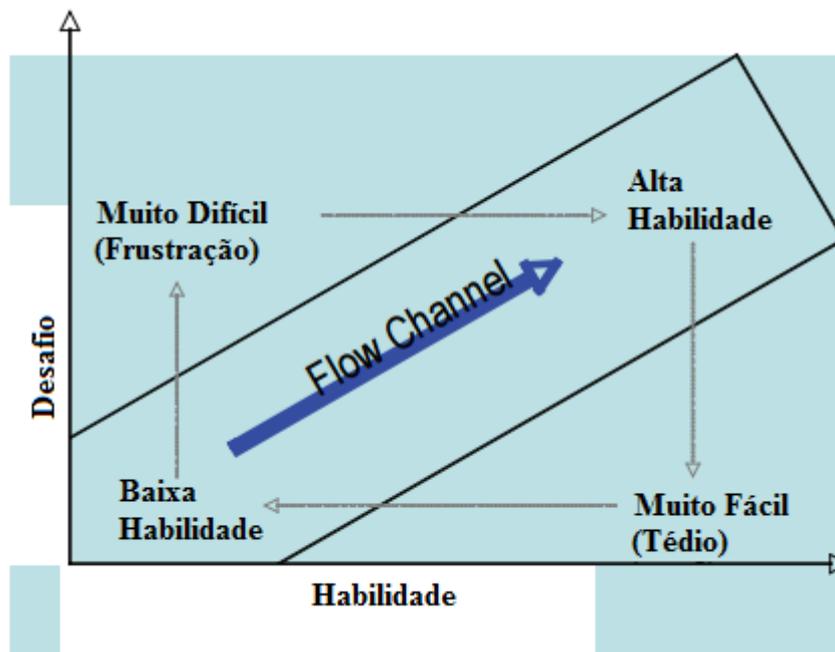
A camada de *gameplay* define o que o jogador, de fato, faz no jogo. Ou seja, as escolhas que o jogador pode fazer dentro do jogo e o que estas escolhas irão impactar ao longo do mesmo (ADAMS, ROLLINGS, 2007 apud WINN, 2008). Esta camada é dividida em mecânicas (*mechanics*), dinâmicas (*dynamics*) e afetos (*affects*).

Conforme Winn (2008), as mecânicas são as regras que definem a operação do jogo, o que o jogador pode fazer, os desafios apresentados ao jogador, e os objetivos do jogador. As dinâmicas são o comportamento resultante da influência das interações dos jogadores com as regras. Os afetos são as emoções e experiências resultantes nos jogadores.

Winn (2008) também aponta que a única forma de determinar se as mecânicas, de fato, atingem os objetivos em questão é por meio de testes do jogo. Com essa informação, o *designer* do jogo pode modificar as mecânicas até que as mesmas se tornem mais adequadas. Esse processo é conhecido como balanceamento do jogo. Várias iterações de projeção, prototipação, testes e revisão são frequentemente necessárias para balancear um jogo (WINN, 2008) (tradução nossa).

Segundo Winn (2008), existem algumas formas de balancear um jogo. Uma delas é por meio do nível de dificuldade. A teoria de fluxo do psicólogo Mihaly Csikszentmihalyi (1990), demonstrada na Figura 12, aponta que para que o estado de fluxo seja alcançado, o nível de desafio precisa ser igualado à habilidade do jogador, conforme a mesma aumenta. Se o desafio é muito grande, o jogador ficará frustrado e pode desistir. Se o desafio é muito pequeno, o jogador rapidamente ficará entediado e pode parar de jogar.

Figura 12 – Teoria de Fluxo de Mihaly Csikszentmihalyi



Fonte: Mihaly Csikszentmihalyi (1990), adaptado pelo autor.

Outra forma de balancear um jogo é por meio da frequência de recompensas dentro do mesmo. Um *designer* deve dar recompensas ao jogador durante os momentos mais desafiadores do jogo, de forma a manter a atenção do mesmo.

Isso pode ser pensado como uma forma de condicionamento operacional, com as recompensas representando reforços em momentos chave, conforme o jogador aprende a jogar o jogo e supera os desafios. (WINN, 2008) (tradução nossa).

Por fim, Winn (2008) sugere que, para balancear o jogo, o *designer* limite as escolhas no início do jogo, e aumente-as conforme o progresso no mesmo, de forma que o jogador não se sinta sobrecarregado enquanto aprende a jogar. O padrão típico para introduzir novas escolhas é no momento em que são disponibilizados novos objetivos ao jogador. O jogador então, precisará adquirir uma nova habilidade para atingir o objetivo. Esta habilidade será praticada até que a mesma seja masterizada e o objetivo, atingido.

#### 4.1.4 CAMADA DE EXPERIÊNCIA DO USUÁRIO

A camada de experiência do usuário é a mais visível a partir da perspectiva do jogador e está relacionada principalmente com a interface, a interatividade e o engajamento do jogador. (SEVERGNINI, 2016).

Segundo Winn (2008), o propósito da interface do usuário é criar um veículo para realizar os propósitos sérios desejados de um *serious game*. A interface engloba tudo que o usuário vê, ouve, interage, e como a interação ocorre. O papel do *designer*, portanto, é desenvolver um jogo que atraia o jogador e que torne a experiência de jogo engajante.

Boas interfaces são ditas como transparentes, ou seja, o jogador não precisa focar sua atenção em como jogar o jogo (qual botão pressionar), mas sim na jogabilidade, narrativa e experiência de aprendizado. (WINN, 2008) (tradução nossa).

#### 4.1.5 CAMADA DE TECNOLOGIA

Segundo Winn (2008), algumas escolhas de *design* são mais dependentes que outras em relação à tecnologia escolhida, normalmente a partir dos subcomponentes ao final do framework para cima. Ou seja, a experiência do usuário é a mais atrelada à tecnologia.

Winn (2008) também observa que, a tecnologia pode ser tanto um habilitador como um limitador. Dependendo da complexidade dos mecanismos que deseja-se implementar, tecnologias com recursos mais sofisticados são necessárias. Assim, é necessário tomar ciência destas diferenças entre tecnologias durante esta tomada de decisão, de forma que o projeto não seja prejudicado.

#### 4.2 DEFINIÇÃO DA TECNOLOGIA

Existem muitas tecnologias que poderiam ser utilizadas para o desenvolvimento do trabalho proposto. Assim, é importante que optemos por uma opção que seja viável e adequada, cujos mecanismos sejam aderentes às nossas necessidades.

A *Unreal Engine 4* é uma *engine* poderosa e amplamente utilizada no mercado, que pode criar desde jogos pequenos até os maiores e mais complexos jogos disponíveis no mercado.

*Unreal Engine 4* é um pacote de ferramentas de desenvolvimento completo, criado para qualquer um que trabalhe com tecnologia de ponta. Desde aplicações empresariais e experiências cinemáticas a jogos de alta qualidade em diversas plataformas como PC, console, *mobile*, VR e AR, *Unreal Engine 4* proporciona tudo que é necessário para começar, disponibilizar, crescer e se destacar. (Unreal Engine, 2004) (tradução nossa).

A *Unreal Engine 4* também conta com uma comunidade grande e a possibilidade de criar jogos em diversas plataformas. Além disso, sua linguagem nativa é C++.

Em contrapartida, temos o *Unity3D*, uma *engine* concorrente à *Unreal Engine* que também é gratuita e muito utilizada atualmente. Esta também tem a possibilidade de criar jogos em diversas plataformas e uma grande comunidade disponível *online*. O *Unity3D*, no entanto, utiliza as linguagens C# ou JavaScript.

As duas *engines* mencionadas disponibilizam uma loja de *assets* onde é possível adquirir modelos, animações, partículas e outros artefatos que possam ser necessários ao *designer* do jogo, o que pode auxiliar no projeto proposto.

No caso deste trabalho, optou-se pelo uso do *Unity3D*, uma vez que a comunidade de desenvolvedores de jogos acredita que o mesmo seja, entre as duas opções, a mais simples e intuitiva para iniciantes. Além disso, a linguagem C#, disponibilizada pelo mesmo, é mais amigável, fácil e com muita documentação *online*, assim como a ferramenta escolhida.

O protótipo proposto neste trabalho foi desenvolvido apenas para o sistema operacional *Android*, uma vez que o sistema operacional *iOS* exige licenças específicas para o desenvolvimento de aplicativos e jogos.

### 4.3 DEFINIÇÃO DO PROTÓTIPO DO *SERIOUS GAME*

Esta seção expõe, a partir de cada camada do *framework* DPE estudado, as ideias e soluções implementadas durante o desenvolvimento do protótipo do *serious game*, onde procuramos criá-lo o mais bem sucedido possível, baseado nos fundamentos e aprendizados adquiridos ao longo deste trabalho.

#### 4.3.1 CAMADA DE APRENDIZAGEM

A camada de aprendizagem deste protótipo, assim como o *Robo Logic* e o *Lightbot*, visa apresentar aos jogadores os conceitos básicos da construção de um algoritmo e a

simulação do mesmo. Uma vez que o público proposto está em seus primeiros contatos com o pensamento computacional, a aprendizagem que devemos propor aos mesmos é simples e de fácil compreensão.

Assim, definiu-se que este *serious game* deveria apresentar ao jogador algoritmos sequenciais, o conceito de *loops* e de procedimentos. O jogador monta uma sequência de comandos (ou seja, um algoritmo) para mover o personagem e atingir um objetivo nas fases do jogo. Os comandos que o jogador pode utilizar para a construção do algoritmo são simples e devidamente introduzidos no conceito do *gameplay* do jogo, para que o jogador possa compreender o propósito dos mesmos no objetivo principal do jogo. O *serious game*, ao ser solicitado pelo jogador, move o personagem para demonstrar os passos que foram executados, proporcionando assim o conceito da simulação do algoritmo.

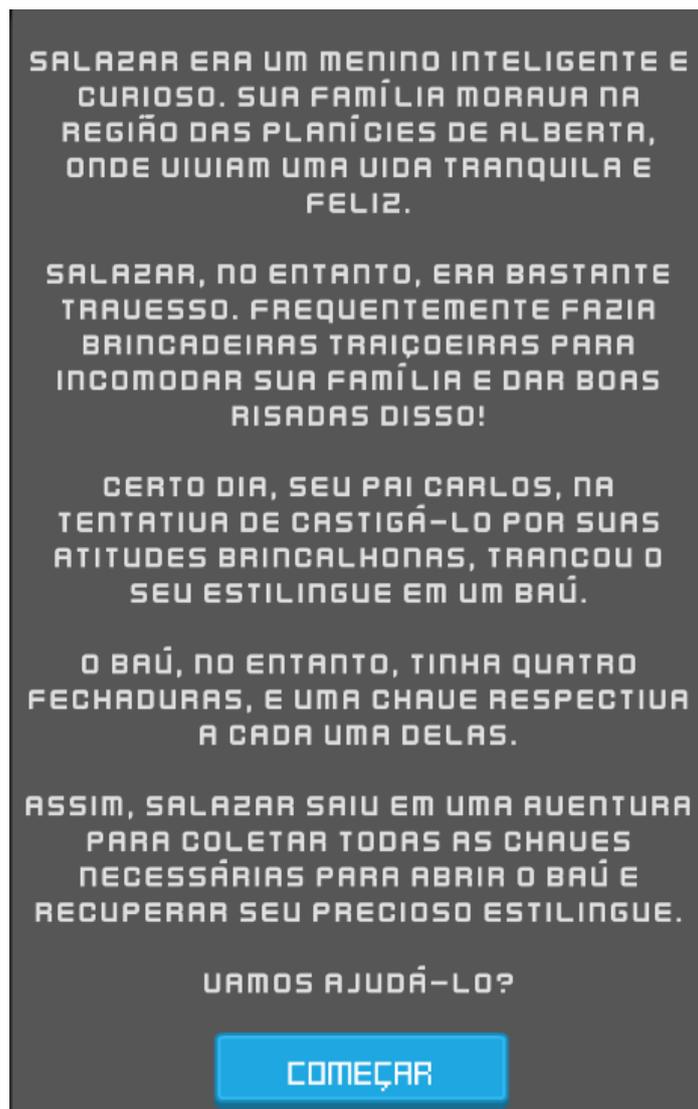
O *serious game* é estruturado de forma a desafiar o jogador ao longo das fases, adicionando níveis de dificuldade conforme o jogador avança. Assim, o mesmo foi dividido em quatro fases. Na primeira fase, o jogador deve utilizar apenas algoritmos sequenciais para passar de fase, tendo apenas uma pilha de comandos para fazê-lo. Na segunda fase, o usuário deve utilizar, além da pilha convencional de comandos, uma estrutura de *loop* para criar os algoritmos para passar de fase. A partir da terceira fase, é liberado apenas a pilha convencional de comandos e a estrutura de procedimentos, de forma que o usuário possa compreender os conceitos de *loop* e procedimento de uma forma onde não haja interferência entre os mesmos. Por fim, a quarta e última fase apresenta todos os conceitos ao usuário, permitindo que ele resolva o objetivo utilizando tanto *loops* quanto procedimentos, além da pilha de comandos convencional. A pilha de comandos principal tem um número máximo de 12 comandos, exigindo assim que os usuários utilizem, nas fases seguinte à primeira, as estruturas adicionais apresentadas.

Uma vez que o projeto proposto é um protótipo, não foi possível adicionar um grande número de fases ao projeto. Isso seria de grande auxílio ao sucesso do aprendizado, pois o jogador deve praticar os conceitos aprendidos, de forma a internalizá-los, antes de ser apresentado a desafios mais complexos.

### 4.3.2 CAMADA DE NARRATIVA

A camada de narrativa deste *serious game* baseia-se na história fictícia e original do personagem *Salazar*. Na introdução do jogo, a seguinte história é apresentada ao usuário, conforme Figura 13.

Figura 13 – Screenshot da tela de introdução do *serious game*



Fonte: Próprio Autor

A história apresentada na tela de introdução do jogo é a seguinte: “*Salazar era um menino inteligente e curioso. Sua família morava na região das planícies de Alberta, onde viviam uma vida tranquila e feliz. Salazar, no entanto, era bastante travesso. Frequentemente*

*fazia brincadeiras traiçoeiras para incomodar sua família e dar boas risadas disso! Certo dia, seu pai Carlos, na tentativa de castigá-lo por suas atitudes brincalhonas, trancou o seu estilingue em um baú. O baú, no entanto, tinha quatro fechaduras, e uma chave respectiva a cada uma delas. Assim, Salazar saiu em uma aventura para coletar todas as chaves necessárias para abrir o baú e recuperar seu precioso estilingue. Vamos ajudá-lo?”.*

A camada de narrativa do *serious game* foi desenvolvida com o intuito de apresentar um objetivo claro e de simples compreensão, de forma a não impactar negativamente a camada de aprendizagem. Ao mesmo tempo em que a narrativa é simples, também deve ser curiosa o suficiente para que seja divertida e engajante aos usuários da idade alvo proposta neste trabalho.

Assim, os objetivos dos cenários das fases estão estruturados no contexto da história, onde os usuários devem ajudar *Salazar* a coletar todas as chaves para abrir o baú, onde cada uma delas se encontra no fim de uma fase.

### 4.3.3 CAMADA DE GAMEPLAY

O protótipo de *serious game* proposto tem seu gênero definido como um quebra-cabeças, assim como o *Robo Logic* e o *Lightbot*. O jogador deverá estruturar uma pilha de ações (comandos), na estrutura e ordem correta, para mover o personagem até o objetivo do cenário.

O jogador poderá usar, conforme a Tabela 7, botões de ação (comandos) para estruturar os movimentos do personagem.

Tabela 7 – Botões de ação do jogo

| Botão   | Significado do Botão   |
|---|------------------------|
|  | Mover-se para Cima     |
|  | Mover-se para Baixo    |
|  | Mover-se para Esquerda |

|   |                                    |
|---|------------------------------------|
|  | Mover-se para Direita              |
|  | Pular                              |
|  | Acionar o Repita                   |
|  | Acionar o Procedimento             |
|  | <i>Play</i> – Executar os comandos |

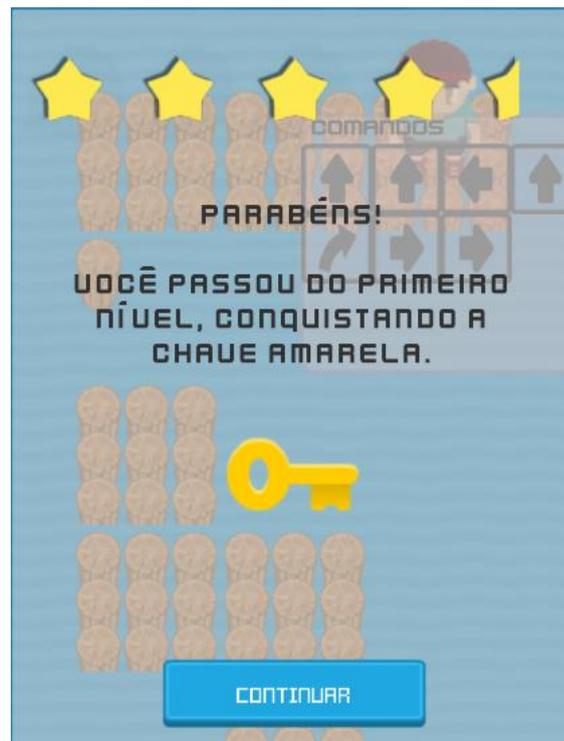
Fonte: Próprio Autor

Os comandos de repita e procedimento, no entanto, são liberados apenas nas fases nos quais os mesmos são necessários, de forma que o jogador não seja sobrecarregado pelas opções disponíveis.

Ao finalizar a estrutura das ações que devem ser executadas, o jogador deve pressionar o botão “*play*”, conforme Tabela 7, na tela para que a simulação da execução da pilha de comandos seja executada, movendo o personagem no cenário.

Caso o usuário tenha executado movimentos errados, a mecânica do jogo procura demonstrar isso, tentando mover o personagem no movimento proposto, mas voltando à posição anterior. Sempre que o usuário efetua um movimento errado, é somado este erro, de forma que seja mostrado no fim da fase, conforme Figura 14, quantas estrelas o usuário recebeu naquele nível. Caso o usuário não realize nenhum movimento errado, será computado uma nota de cinco estrelas. A cada movimento errado, o usuário perderá meia estrela. Assim, com dez erros ou mais, o usuário não recebe nenhuma estrela de nota. Esta implementação tem o objetivo de prover ao usuário uma experiência de *feedback* de performance e um senso de competitividade ao *serious game*.

Figura 14 – Screenshot do painel de final de nível do *serious game*



Fonte: Próprio Autor

Os cenários do *serious game* são gerados a partir de matrizes definidas internamente no jogo (Tabela 8, Tabela 9, Tabela 10 e Tabela 11). Assim, o jogo lê, a cada fase, as posições da matriz e monta as *tiles* nas quais o personagem deve andar ou pular. Ao montar o cenário, o jogo armazena em uma classe *Position* as informações pertinentes a cada posição do cenário. Assim, é armazenado nessa classe a posição central de cada *tile*, de forma que seja possível efetuar o movimento do personagem até o centro de cada uma, durante a leitura de seus movimentos da pilha de comandos. Por vezes há situações onde o cenário é muito grande, o que exige que o usuário movimente o cenário para cima e para baixo, para visualizar o mesmo por completo.

Para que o *serious game* tenha um fator de *replayability*<sup>4</sup> maior, cada uma das quatro fases do jogo pode ser gerada a partir de três matrizes diferentes. Assim, ao gerar a fase, o jogo randomiza uma das três matrizes disponíveis para a fase, dando ao usuário a oportunidade de jogar mais vezes o jogo, uma vez que a sequência de passos para concluir o jogo não será sempre a mesma. As Tabelas 8, 9, 10 e 11 evidenciam a estrutura das matrizes utilizadas nas quatro fases do *serious game*. As posições em azul claro são as *tiles* nas quais o

<sup>4</sup> Termo utilizado para avaliar o potencial de um jogo digital em continuar sendo jogado após ser completado.

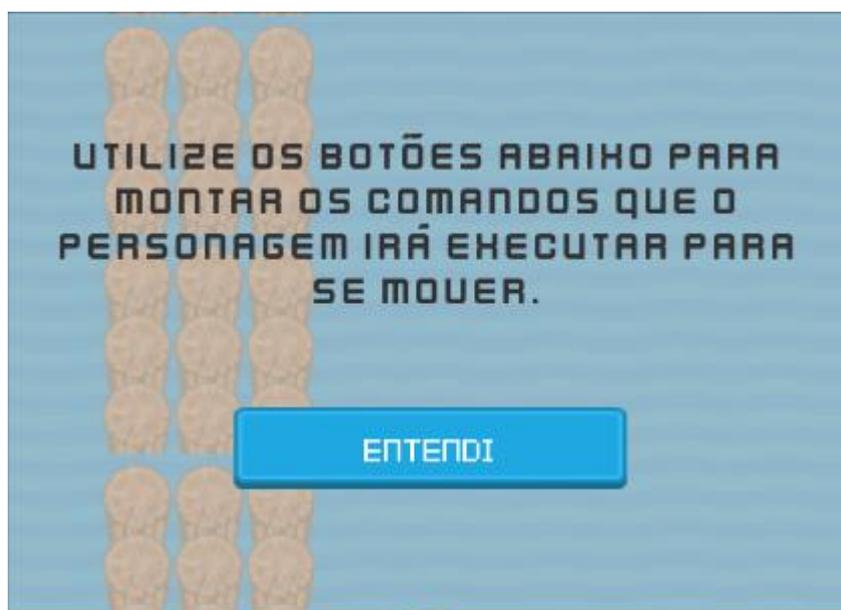




Os comandos montados pelo usuário são armazenados num vetor, e são lidos pelo jogo sempre que o usuário pressiona “*play*”, iniciando o trajeto do personagem sempre do início do cenário e, por consequência, do início da pilha de comandos. Os comandos informados na estrutura de repita e procedimento são armazenados em variáveis e vetores auxiliares, e são buscados nessas estruturas quando é identificado uma chamada para repita ou procedimento na pilha de comandos principal.

Ao carregar as fases, são abertos alguns painéis de instruções, conforme Figura 15, que dão dicas da jogabilidade ao jogador e explicam como utilizar os botões da tela. Na segunda e terceira fase são mostradas, respectivamente, algumas dicas sucintas de como o repita e o procedimento são utilizados.

Figura 15 – *Screenshot* de painel de instruções



Fonte: Próprio Autor

#### 4.3.4 CAMADA DE EXPERIÊNCIA DO USUÁRIO

A camada de experiência do usuário construída neste protótipo utiliza uma perspectiva *Top-Down*, abordada em um ambiente 2D.

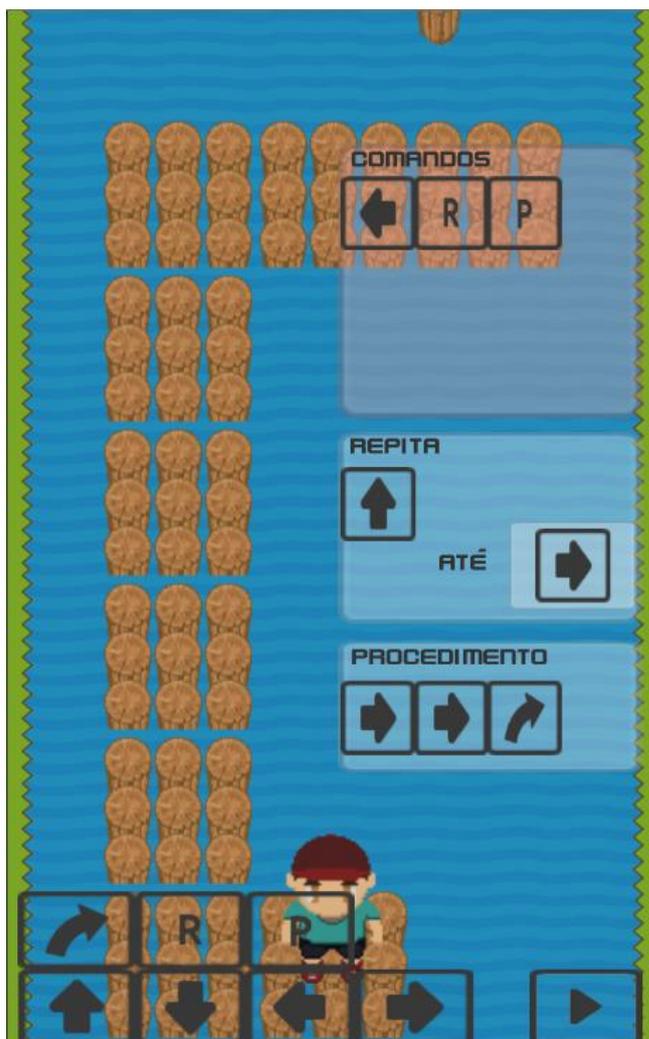
As texturas, *sprites*, materiais e sons que foram utilizados na construção deste *serious game* foram adquiridos na *internet*, uma vez que não é foco deste trabalho a produção de ilustrações digitais. Foi tomado o cuidado para que todos os materiais adquiridos na *internet* tivessem licenças de domínio público, permitindo seu livre uso na implementação

deste trabalho. Apesar de não requisitarem, todos os autores dos materiais utilizados foram devidamente mencionados na tela “Sobre”, disponível no *serious game*. Alguns materiais e animações foram criados com o auxílio de um terceiro, que também foi devidamente mencionado.

A construção das texturas do cenário foi idealizada com a intenção de mantê-lo contextualizado na narrativa proposta, ao mesmo tempo em que procura-se torná-lo divertido e interessante ao público-alvo do trabalho. As *tiles* utilizadas na criação do cenário foram idealizadas pelo autor do trabalho e criadas com um auxílio externo, criando uma ponte na qual o personagem, *Salazar*, caminha para atingir seus objetivos em cada cenário. As texturas de fundo, apesar de adquiridas na *internet*, tiveram algumas modificações pontuais para se adequarem ao contexto deste *serious game*. A animação da água, por sua vez, foi criada por um terceiro para melhorar o visual e a experiência do jogo, e consiste em *sprites* diferentes, onde as “ondas” da água mudam de posição.

O personagem do jogo não teve alterações, e suas animações de caminhar já estavam prontas. As animações de pulo, no entanto, foram todas feitas sob demanda para o jogo, de forma que a ação do pulo não parecesse estranha sem uma animação adequada. Foi optado por esta *sprite* para o personagem pela sua estética e por adequar-se ao personagem da narrativa, um garoto.

A Figura 16 exemplifica uma imagem completa da tela de jogo do *serious game*.

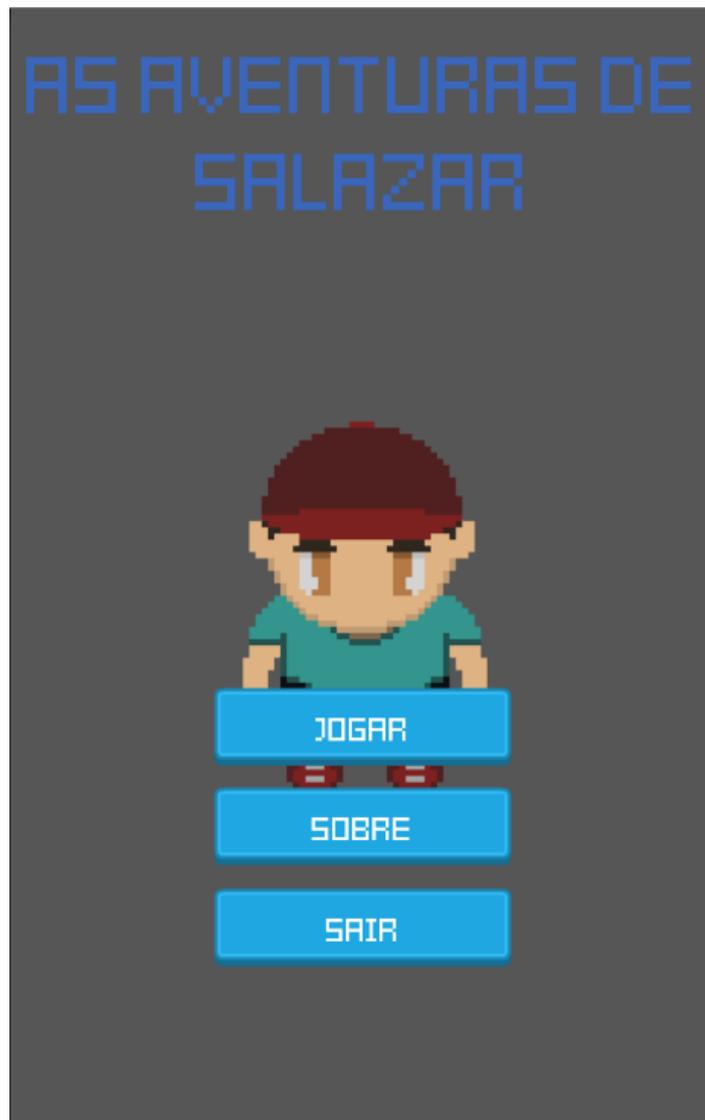
Figura 16 – Screenshot da tela de jogo do *serious game*

Fonte: Próprio Autor

Os botões e painéis do jogo foram construídos a partir de um modelo padrão, e tiveram algumas alterações pelo autor, para se adequarem ao contexto deste *serious game*. Uma vez que os botões ficam sob a visão do jogador para o cenário, e tem algum impacto na visualização do jogo, o fundo dos mesmos é transparente, na intenção de diminuir este impacto.

Foi criado um menu simples para o *serious game*, conforme Figura 17, utilizando o personagem do jogo como plano de fundo, dando ênfase ao mesmo e ao título do jogo. Dessa forma, dá-se ao jogo um senso de importância à narrativa do mesmo.

Figura 17 – Screenshot do menu do *serious game*



Fonte: Próprio Autor

Além disso, uma vez que a trilha sonora de um jogo é importante para torná-lo divertido, foram adicionadas ao jogo duas músicas, com um estilo de RPG, que são selecionadas de forma aleatória em cada uma das fases do jogo.



## 5 VALIDAÇÃO E RESULTADOS

Com a conclusão do desenvolvimento do protótipo do *serious game*, passamos à fase de *playtesting*<sup>5</sup>, a fim de validar o protótipo proposto e avaliar os resultados do mesmo. O *playtesting* é, conforme o *framework* DPE, indispensável durante o desenvolvimento de um *serious game*. Em um processo ideal, o *playtesting* deveria ocorrer várias vezes durante o desenvolvimento do projeto, uma vez que é durante esta fase que encontramos problemas e as principais melhorias para o jogo. Apenas monitorando a jogabilidade a partir do público-alvo podemos compreender o que, de fato, precisa ser ajustado ou melhorado. Como este projeto trata-se de um protótipo, apenas duas sessões de *playtesting* foram realizadas.

A primeira etapa foi aplicada com cinco crianças de 7 anos de idade, e uma de 9 anos. Desta amostra, 6 eram meninos e 1 era menina. Os testes ocorreram entre os dias 16/10 e 20/10, no Colégio São José de Caxias do Sul, durante a aula de programação de jogos, da escola *Classcode*. As sessões de teste duraram, em média, 15 minutos para cada usuário. Por participarem de uma aula de programação, estes usuários já tinham uma introdução ao pensamento computacional.

A segunda etapa foi aplicada com cinco crianças, de diferentes escolas, duas do sexo feminino e três do sexo masculino. As crianças tinham 7, 8, 9, 11 e 12 anos, sendo as respondentes de 8 e 11 anos, meninas. Esta etapa ocorreu entre os dias 02/11 e 05/11, fora do ambiente escolar. Além disso, estes testes também duraram uma média de 15 minutos para cada usuário. Esta segunda etapa diferencia-se da primeira etapa pois os usuários não tinham nenhuma introdução básica ao pensamento computacional.

Após testarem o jogo, os usuários responderam algumas perguntas, em forma de entrevista, a partir da estrutura apresentada no Apêndice A deste documento. O questionário foi construído procurando avaliar, principalmente, se os usuários compreendem o objetivo e as teorias apresentadas pelo jogo e se a narrativa e a estética do mesmo são engajadoras. Ou seja, se o jogo é bem-sucedido naquilo que é proposto: ser um *serious game* divertido para os usuários do público-alvo.

É importante ressaltar que, início dos testes, não foi dado aos jogadores nenhuma explicação prévia em relação ao jogo ou o que era esperado dos testes, pois o intuito era

---

<sup>5</sup> *Playtesting* é o processo onde desenvolvedores de jogos testam um novo jogo, validando sua construção e buscando possíveis melhorias ao mesmo.

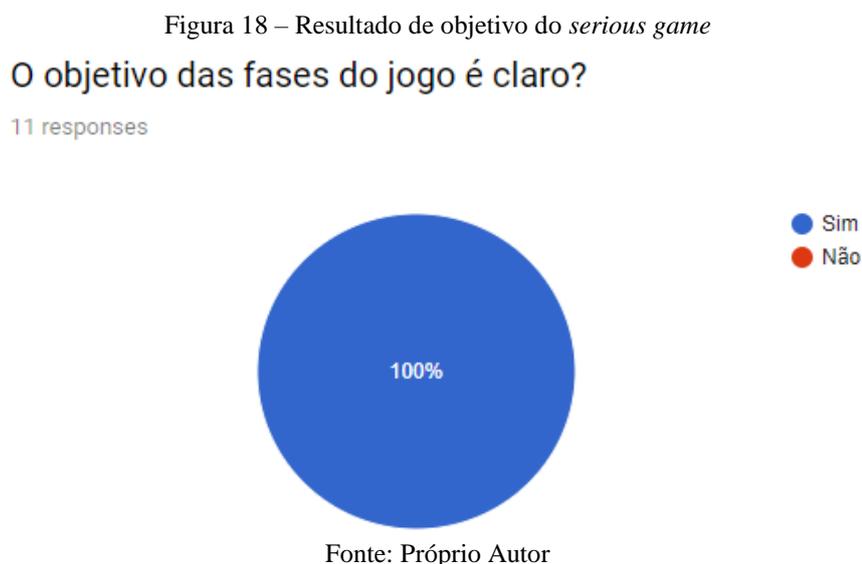
avaliar se o jogo era autoexplicativo e se os usuários conseguiriam compreender, por conta própria, como a jogabilidade e o aprendizado do mesmo eram desenvolvidos.

## 5.1 RESULTADOS DO *PLAYTESTING*

Durante o processo de *playtesting*, foram selecionados aleatoriamente um respondente de 6 anos e cinco respondentes de 7 anos, visto que estes pertencem à faixa-etária do público-alvo original do projeto. No entanto, para melhor medir os resultados, foram entrevistados também um usuário de 8 anos, dois usuários de 9 anos, um usuário de 11 anos e um usuário de 12 anos, totalizando assim os 11 participantes do *playtesting*. Além disso, dos 11 respondentes totais, três deles são do sexo feminino.

Os testes foram realizados procurando dividir os usuários entre aqueles que já tinham algum conhecimento introdutório ao pensamento computacional e aqueles que não tinham. Assim, dos 11 respondentes, seis tinham introdução ao pensamento computacional.

Ao elencarmos os resultados apresentadas pelas questões que dizem respeito ao *serious game*, podemos considerar, conforme Figura 18, que houve alguns resultados bastante positivos. Todos os usuários participantes do *playtesting* entenderam, com facilidade, qual era o objetivo pretendido nas fases do jogo.



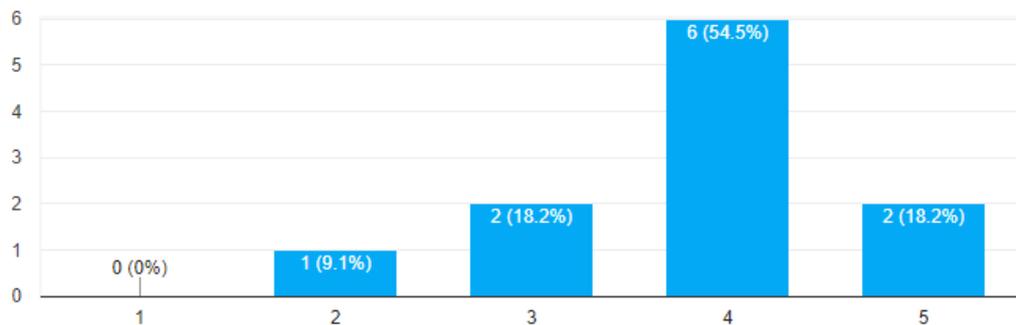
Ao avaliarmos a facilidade de compreensão da jogabilidade, conforme a Figura 19, verificamos que a maioria dos usuários foi capaz de compreender a jogabilidade do jogo para

passar das fases. Isso diz respeito a forma como o jogo deveria ser jogado. Ou seja, se o usuário compreendia que era necessário adicionar os comandos às pilhas dispostas a direita da tela para realizar a movimentação do personagem *Salazar*.

Figura 19 – Resultados de jogabilidade do *serious game*

Avalie a facilidade de compreensão da jogabilidade do jogo para passar das fases:

11 responses



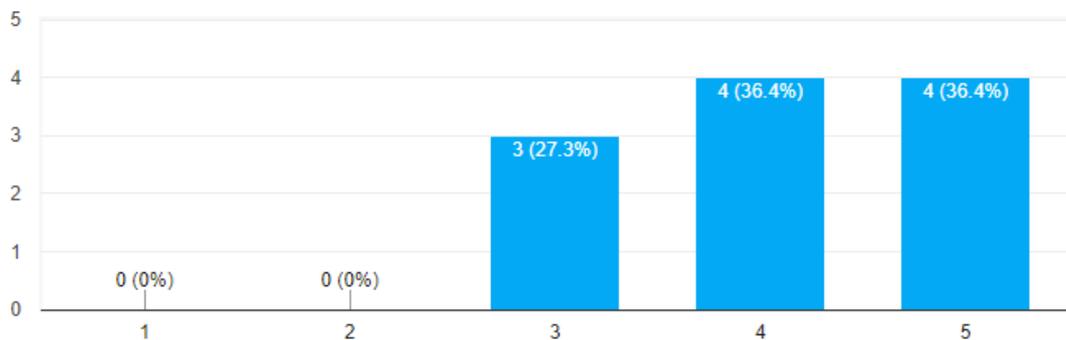
Fonte: Próprio Autor

Podemos verificar também que, conforme a Figura 20, as imagens utilizadas para os botões tiveram um resultado satisfatório, onde mais de 72% dos respondentes apontou um resultado de fácil a muito fácil para a compreensão dos mesmos.

Figura 20 – Resultados de compreensão dos botões do *serious game*

Avalie a facilidade de compreensão do uso de cada botão de comando:

11 responses



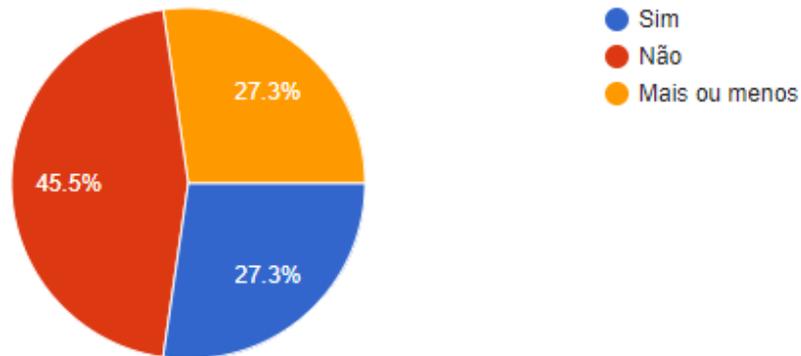
Fonte: Próprio Autor

No que diz respeito à avaliação da compreensão de *loop* e procedimento, podemos observar uma discrepância entre os resultados dos dois, conforme a Figura 21 e a Figura 22.

Figura 21 – Resultados do *loop* do *serious game*

## Você compreendeu a função do loop?

11 responses

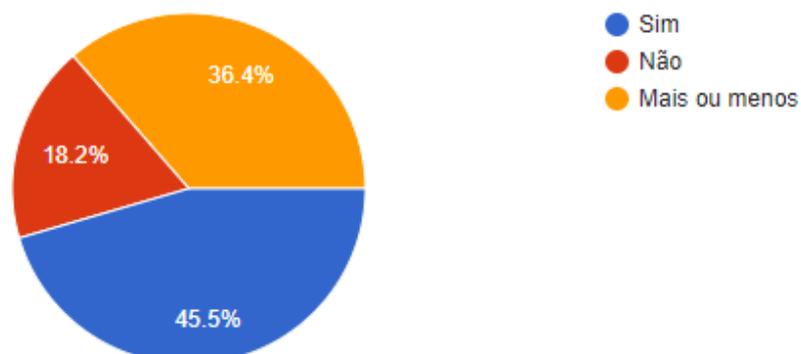


Fonte: Próprio Autor

Figura 22 – Resultados do procedimento do *serious game*

## Você compreendeu a função do procedimento?

11 responses



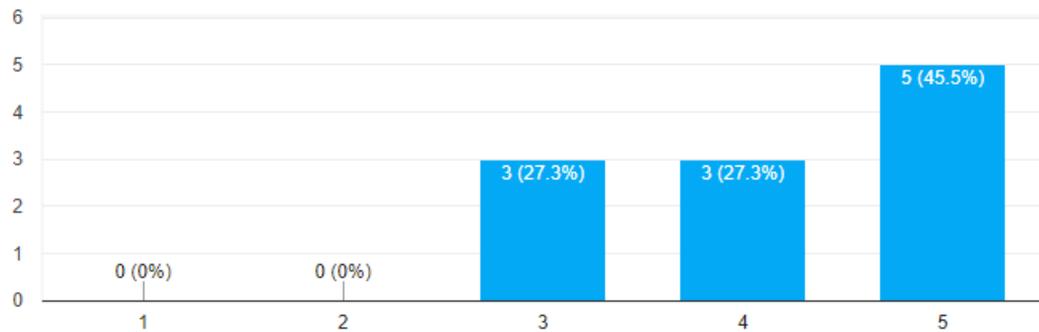
Fonte: Próprio Autor

Foi levantando também o resultado da estética do jogo com os usuários, para avaliar a capacidade de atração do mesmo no que diz respeito à sua parte visual, uma vez que um *serious game* deve chamar a atenção dos seus jogadores em geral e, principalmente, seu público-alvo. A Figura 23 expõem estes resultados.

Figura 23 – Resultados de estética do *serious game*

## Avalie a estética da interface do jogo:

11 responses



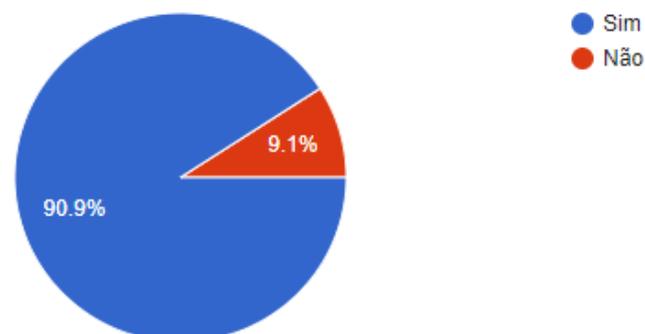
Fonte: Próprio Autor

Durante a entrevista, foi solicitado aos usuários se os mesmos julgavam o jogo desafiador às suas capacidades e se continuariam jogando o mesmo, caso houvessem mais fases disponíveis. Essas questões foram levantadas para averiguar se o *serious game* atendia a necessidade de apresentar desafios aos jogadores, mantendo-os interessados e engajados no jogo e no conhecimento que ele tem a oferecer. A Figura 24 aponta que 90,9% dos usuários acharam o jogo desafiador. Um fato interessante em relação a este dado é que o respondente que não julgou o jogo desafiador tinha nove anos.

Figura 24 – Resultados de desafio do *serious game*

## Você achou o jogo desafiador?

11 responses



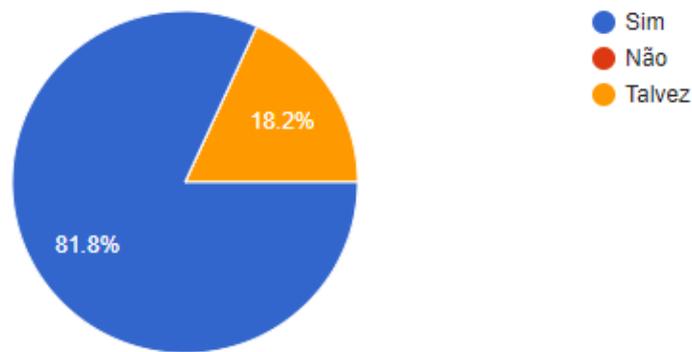
Fonte: Próprio Autor

A Figura 25 mostra o resultado dos respondentes que continuariam jogando o jogo caso houvesse mais fases. 81,8% continuariam jogando (os 18,2% restantes responderam “Talvez”).

Figura 25 – Resultados de usuários que continuariam jogando o *serious game*

### Você continuaria jogando se houvesse mais fases?

11 responses



Fonte: Próprio Autor

Na intenção de avaliar a narrativa do *serious game* e identificar se a mesma gera um sentimento de atração e afeto entre o usuário e o personagem do jogo, foi solicitado aos usuários se os mesmos gostavam do personagem *Salazar*, visto que a história apresentada aponta o mesmo como um garoto travesso que está tentando recuperar seu estilingue tomado por seu pai. Dos respondentes, 90,9% informaram que gostaram de *Salazar*, enquanto 9,1% responderam não gostar de *Salazar*.

Além disso, foi questionado aos usuários se os painéis de instruções, mostrados no início das fases, eram esclarecedores e auxiliavam na compreensão do jogo. 54,5% dos respondentes informaram que “sim”, e 45,5% responderam “mais ou menos”.

Por fim, procurando avaliar se a decisão de criar o protótipo do *serious game* focado em aparelhos móveis foi, de fato, uma boa escolha, solicitamos aos usuários se os mesmos acharam confortável jogar este jogo em um aparelho móvel. Dos respondentes, 100% informaram que sim.

Uma relação completa dos resultados de *playtesting* está disponibilizada no Apêndice B deste documento.

## 5.2 AVALIAÇÃO DOS RESULTADOS DO *PLAYTESTING*

Com a conclusão da etapa de *playtesting*, foi possível realizar uma análise mais detalhada dos resultados do protótipo do *serious game* proposto e implementado, interpretando e observando seu uso a partir de usuários do público-alvo do projeto.

Foi possível também, compreender melhor a importância da fase de *playtesting* para o projeto como um todo, visto que é muito difícil prever qual será a percepção do público-alvo. Tendo isso em mente, acredita-se que os resultados obtidos durante o *playtesting* foram bastante positivos. O projeto mostrou, apesar de ser apenas um protótipo, grande potencial como artefato de aprendizagem e diversão. O *serious game* também provou-se desafiador para o público-alvo, o que apontou oportunidades para melhorias.

Uma vez que a nossa amostragem é muito baixa, não podemos realizar nenhuma conclusão concreta. No entanto, ficou claro que a inclusão do *loop* como conceito de aprendizagem aumentou consideravelmente a complexidade do jogo. Desta forma, foi constatado que o *loop* causa um aumento inevitável na faixa-etária proposta pelo *serious game*, uma vez que as crianças submetidas aos testes com uma idade de nove anos ou mais conseguiram compreender o conceito de forma um pouco menos custosa.

Outro fator importante, que impacta na complexidade do jogo, são os painéis de instruções. Apesar dos usuários terem passado um resultado positivo à esta questão no durante a entrevista, os painéis de instruções certamente poderiam ter uma implementação mais trabalhada, pois houve casos onde alguma interferência externa foi necessária, de forma a auxiliar os usuários na compreensão das teorias apresentadas. Estes painéis de instruções também poderiam ser escritos com uma linguagem mais aderente à faixa-etária proposta pelo projeto.

Ao observarmos os usuários usufruindo do *serious game*, é possível concluirmos também que, o texto de narrativa apresentado na introdução do jogo é ineficiente e pouco atraente à faixa-etária do público respondente a pesquisa. Apesar de alguns usuários terem lido a narrativa (somente as crianças de 11 e 12 anos), muitos deles não o fizeram pois há muito texto, o que não é apelativo aos mesmos. A apresentação da narrativa do jogo seria muito mais bem-sucedida se a mesma fosse animada.

Observamos também que, como a implementação inicial do protótipo contava com um botão para excluir os comandos das pilhas, isso tornava esta ação confusa aos usuários, pois os mesmos não conseguiam compreender como realizá-la. Assim, o jogo foi alterado

para permitir apagar comandos das pilhas de comandos clicando diretamente na imagem dos mesmos, ao invés de utilizar um botão. Além de proporcionar uma melhoria estética, isso permitiu que o usuário excluísse comandos do meio da pilha, e não apenas o último, criando uma jogabilidade mais intuitiva, agradável e fluída.

Outra análise que podemos fazer em relação à implementação deste protótipo é que, na sua concepção, os comandos errados eram apresentados ao usuário com um movimento “em falso” do personagem, onde o mesmo realizava uma tentativa de mover-se na direção proposta e volta à posição anterior. Ao realizar os testes com os usuários da faixa-etária proposta, no entanto, foi possível perceber que esta ação não fica tão evidente aos mesmos quanto se desejava. Uma solução para esta situação seria adicionar um som que indicasse um comando errado durante a movimentação do personagem. Outra solução mais complexa, e provavelmente mais eficiente, seria fazer com que o personagem *Salazar* caísse na água quando houvessem movimentos errados, retornando o mesmo ao início da fase. Estas alternativas poderão ser consideradas em trabalhos futuros.

## 6 CONSIDERAÇÕES FINAIS

A partir da investigação e pesquisa dos conceitos e teorias do pensamento computacional efetuados neste trabalho, podemos compreender melhor e com mais clareza, a partir da Teoria Cognitiva de Piaget, como este é formado em pessoas, a partir dos primórdios do desenvolvimento cognitivo humano. Podemos, também, concluir que o pensamento computacional é útil a todos, independentemente de seu futuro profissional, e sua aprendizagem deveria ser estimulada e disponibilizada para crianças em seu início de vida acadêmica. Foi estudada, também, a Zona de Desenvolvimento Proximal de Vygotsky, e como a aprendizagem de conceitos ocorre até que seja internalizado por aquele que está aprendendo. Pode-se associar os *serious games* à Zona de Desenvolvimento Proximal, ao identificá-los como o facilitador que auxilia aquele que o usa para a aprendizagem.

Durante o desenvolvimento do trabalho, investigou-se também os conceitos dos *serious games* e as características que os tornam distintos de jogos comerciais convencionais. Compreendeu-se que um *serious game* deve, acima de tudo, manter-se focado em sua tarefa de prover conhecimento ao jogador, ao mesmo tempo que procura tornar a aprendizagem dos conceitos divertida e engajante. Buscou-se, também, elencar alguns trabalhos relacionados, cujo objetivo de ensino assemelha-se a este, para compreender como são estruturados e de que forma tornaram-se bem sucedidos em seu objetivo.

A motivação para criar o protótipo do *serious game* para plataformas *mobile* deve-se ao fato de que os jovens, nos quais este trabalho propõem-se a focar, estão atualmente muito conectados a estas plataformas. Assim, os jovens têm mais afinidade com estas plataformas do que com computadores convencionais.

A pesquisa do *framework* DPE (*Design, Play, Experience*) permitiu encontrar uma forma apropriada de estruturar a análise do protótipo de *serious game*, dividindo-a nas camadas necessárias para a definição apropriada do mesmo, iniciando pelo objetivo do que se quer ensinar e como se pretende fazê-lo. O *framework* também ensina que, para o desenvolvimento de um *serious game*, é muito importante definir um público-alvo para o mesmo, uma vez que a experiência de diferentes jogadores raramente será a mesma. Assim, o *serious game* deve estar focado na intenção de ser bem sucedido na aprendizagem dos jogadores do público alvo. Compreendeu-se, também, a dificuldade de criar um *serious game*, uma vez que o *framework* ensina que o projeto deve passar por várias fases de testes, *design* e prototipação até que esteja, de fato, concluído.

A primeira etapa deste trabalho, portanto, visou compreender a importância do pensamento computacional e como se pode introduzi-lo, da melhor forma possível, em um *serious game*, utilizando este para auxiliar futuras gerações a estarem habituadas às teorias e benefícios desta forma de pensamento.

Durante o desenvolvimento do protótipo, tivemos a oportunidade de compreender o desafio e a dificuldade de criar um *serious game*. Algumas implementações necessárias no jogo foram bastante desafiadoras, por não termos um conhecimento tão amplo da *engine Unity*. Foi necessário corrigir a resolução da tela, já que, no início do desenvolvimento, utilizamos uma resolução baixa que não se adaptaria corretamente à aparelhos mais recentes. Além disso, as bibliotecas nativas da *engine* para a identificação de toque em tela não são muito eficientes. Isso dificultou a implementação da movimentação da tela e da exclusão dos comandos nas pilhas, que acabaram se tornando funcionalidades complexas.

Apesar de termos realizado uma análise inicial das necessidades do projeto, e procurado estruturar o mesmo o mais corretamente possível, fica claro ao desenvolvedor os problemas de escalabilidade do código-fonte que surgem ao longo do projeto. Assim que o projeto começa a se tornar muito grande, torna-se muito difícil realizar implementações novas que não tenham sido muito bem pensadas durante os levantamentos de casos de uso do projeto. *Bugs* em geral também podem se tornar potencialmente perigosos, dependendo do nível da implementação no qual os mesmos se encontram. Assim, é muito importante que as classes do projeto sigam o princípio de terem, cada uma, uma única responsabilidade, permitindo uma organização muito melhor do código-fonte e da escalabilidade do projeto. Apesar de julgarmos que o desenvolvimento do projeto esteja bem organizado e estruturado, acredita-se que, caso o mesmo fosse iniciado hoje, algumas melhorias poderiam ser feitas para melhorá-lo neste sentido.

Vivenciar, observar e poder participar de uma fase de *playtesting*, principalmente realizada com crianças, é sem dúvidas uma experiência excelente. Não é fácil prever como os usuários irão se comportar diante das implementações que apresentamos a eles. Mais difícil ainda a aqueles que têm pouco contato com esta faixa-etária. Assim, compreender como os mesmos utilizam e desfrutam do jogo nos permite fazer avaliações e tirar conclusões importantes para aprimorar o projeto. Após a aplicação dos testes com as crianças, ficou evidente que o conceito de estrutura de repetição incluído no jogo situa-se no nível de desenvolvimento Operacional Formal, ou, como Piaget indica, na etapa que inicia aos 11 anos

de idade, em que é desenvolvida a habilidade de pensar em conceitos abstratos e testar hipóteses logicamente.

## 6.1 TRABALHOS FUTUROS

Em virtude dos resultados levantados e apresentados no *playtesting*, elencamos assim algumas melhorias que julgamos serem pertinentes à trabalhos futuros, e que tornariam este protótipo de *serious game* ainda mais bem-sucedido em seu objetivo:

- Alterar a mecânica do jogo, de forma que o personagem caia da ponte para a água (e retornando assim ao início da fase) quando o usuário realizar movimentos incorretos, tornando assim mais claro para o usuário de que aquele não é um comando adequado para dada situação;
- Tornar os painéis de instruções mais aderentes à faixa-etária do jogo, utilizando uma linguagem mais adequada e criando animações para simular e exemplificar a forma como as ferramentas disponibilizadas pelo jogo devem ser utilizadas;
- Apresentar a narrativa do jogo de uma forma animada, tornando-a assim mais divertida e engajante ao público-alvo e procurando utilizar menos texto, uma vez que os testes indicam que estes não são eficientes para esta faixa-etária;
- Alterar a ordem entre duas fases do jogo, apresentando o conceito de procedimento antes do conceito de repetição. Esta conclusão deve-se ao fato de que, durante os testes, foi observado que as crianças tinham mais facilidade em entender o funcionamento do procedimento. A “condição de parada” presente na estrutura de repetição adicionou uma complexidade grande de compreensão às crianças.

## REFERÊNCIAS BIBLIOGRÁFICAS

BECKER, K. Video game pedagogy: good games = good pedagogy. **Games: Their Purpose and Potential in Education**, New York, 2008.

C. ABT, C. **Serious Games**. Lanham: University Press of America, 1987.

CODECOMBAT. About: CodeCombat - Learn how to code by playing a game. **CodeCombat**, 2013. Disponível em: <<http://br.codecombat.com/about>>. Acesso em: 01 Maio 2017.

CODINGAME. **CodinGame**, 2017. Disponível em: <<https://www.codingame.com/home>>. Acesso em: 01 Maio 2017.

COSTA, S. S. et al. Um estudo exploratório dos games para introdução ao pensamento computacional. **7º Congresso Nacional de Ambientes Hiperídia para Aprendizagem**, São Luis, 2015.

DETERDING, S. CHI 2011 Workshop Gamification. **Using Game Design Elements in Non-Game Contexts**, 2011. Disponível em: <<http://gamification-research.org/wp-content/uploads/2011/04/01-Deterding-Sicart-Nacke-OHara-Dixon.pdf>>. Acesso em: 05 Março 2017.

DIGITALSIRUP. Robo Logic. **DigitalSirup**, 2009. Disponível em: <<https://www.digitalsirup.com/app/robologic/?lang=en>>. Acesso em: 13 Maio 2017.

FACCHIN, A. **Proposta de rede social gamificada para associações de câncer infantil**. Universidade de Caxias do Sul. Caxias do Sul. 2013.

FARDO, M. L. **A gamificação como estratégia pedagógica: Estudo dos Elementos dos Games aplicados em processos de ensino e aprendizagem**. Universidade de Caxias do Sul. Caxias do Sul. 2013.

GAGNÉ, R. M. **Mastery Learning and Instructional Design**. [S.l.]: Blackwell Publishing Ltd, v. 1, 1988.

HUNICKE, R.; LEBLANC, M.; ZUBEK, R. MDA: A Formal Approach to Game Design and Game Research, 2004. Disponível em: <<http://www.cs.northwestern.edu/~hunicke/MDA.pdf>>. Acesso em: 14 Maio 2017.

INAL, Y.; CAGILTAY, K. Flow experiences of children in an interactive social game environment. **British Journal of Educational Technology**, v. 38, 2007.

JONASSEN, D. H. **Learning to Solve Problems - An Instructional Design Guide**. San Francisco: Pfeiffer, 2004.

LAAMARTI, F.; EID, M.; EL SADDIK, A. An Overview of Serious Games. **International Journal of Computer Games Technology**, 2014.

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

M. WING, J. Computational Thinking. **Center for Computational Thinking**, Pittsburgh, Março 2006. Disponível em: <<http://www.cs.cmu.edu/~CompThink/papers/Wing06.pdf>>. Acesso em: Abril 2017.

M. WING, J. Computational thinking and thinking about computing. **Center for Computational Thinking**, 2008. Disponível em: <<https://www.cs.cmu.edu/~CompThink/papers/Wing08a.pdf>>. Acesso em: Abril 2017.

MCGONIGAL, J. **A Realidade Em Jogo**. Rio de Janeiro: Best Seller, 2012.

MCLEOD, S. Jean Piaget. **SimplyPsychology**, 2015. Disponível em: <[www.simplypsychology.org/piaget.html](http://www.simplypsychology.org/piaget.html)>. Acesso em: Abril 2017.

MICHAEL, D.; CHEN, S. **Serious Games: Games That Educate, Train, and Inform**. [S.l.]: Thomson Course Technology PTR, 2005.

SALEN, K.; ZIMMERMAN, E. **Rules of Play: Game Design Fundamentals**. Massachusetts: The MIT Press, 2004.

SEVERGNINI, L. F. **Serious game como ferramenta de ensino de lógica de programação para crianças**. Universidade de Caxias do Sul. Caxias do Sul. 2016.

UNITY - Game Engine. **Unity3D**. Disponível em: <<https://unity3d.com/pt/>>. Acesso em: 27 Maio 2017.

UNREAL Engine. **Unreal Engine**, 2004. Disponível em: <<https://www.unrealengine.com/unreal-engine-4>>. Acesso em: 26 Maio 2017.

VYGOTSKY, L. S. **A Formação Social da Mente: O Desenvolvimento dos Processos Psicológicos Superiores**. 6ª. ed. São Paulo: Martins Fontes, 1998.

WINN, B. M. The Design, Play, and Experience Framework. **Handbook of Research on Effective Electronic Gaming in Education**, Michigan State University, v. III, p. 1010-1024, 2008. Disponível em: <[http://gel.msu.edu/winn/Winn\\_DPE\\_chapter\\_final.pdf](http://gel.msu.edu/winn/Winn_DPE_chapter_final.pdf)>. Acesso em: 21 Maio 2017.

ZERBINI, T.; ABBAD, G. Aprendizagem induzida pela instrução em contexto de organizações e trabalho: uma análise crítica da literatura. **Períodos Eletrônicos em Psicologia**, São Paulo, Dezembro 2008. Disponível em: <<http://pepsic.bvsalud.org/pdf/cpst/v13n2/v13n2a03.pdf>>. Acesso em: 07 Maio 2017.



## **APÊNDICE A – QUESTIONÁRIO DE *PLAYTESTING***

A seguir, apresentamos o questionário de *playtesting* utilizado durante as entrevistas para a coleta dos resultados e experiências obtidos pelos usuários.



## As Aventuras de Salazar

Este questionário tem o objetivo de verificar e avaliar a experiência de jogo e aprendizagem oferecida pelo serious game "As Aventuras de Salazar", desenvolvido por Henrique Pagno de Lima e orientado pela professora Elisa Boff como Trabalho de Conclusão de Curso para a obtenção do título de Bacharel em Ciência da Computação pela Universidade de Caxias do Sul.

**\*Required**

1. Idade: \*

---

2. Sexo: \*

*Mark only one oval.*

Masculino

Feminino

3. Você tem algum conhecimento prévio de pensamento computacional? \*

*Mark only one oval.*

Sim

Não

4. O objetivo das fases do jogo é claro? \*

*Mark only one oval.*

Sim

Não

5. Avalie a facilidade de compreensão da jogabilidade do jogo para passar das fases: \*

*Mark only one oval.*

|               |                       |                       |                       |                       |                       |             |
|---------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-------------|
|               | 1                     | 2                     | 3                     | 4                     | 5                     |             |
| Muito difícil | <input type="radio"/> | Muito fácil |

---

6. Avalie a facilidade de compreensão do uso de cada botão de comando: \*

*Mark only one oval.*

|               |                       |                       |                       |                       |                       |             |
|---------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-------------|
|               | 1                     | 2                     | 3                     | 4                     | 5                     |             |
| Muito difícil | <input type="radio"/> | Muito fácil |

7. Você compreendeu a função do loop? \*

*Mark only one oval.*

- Sim  
 Não  
 Mais ou menos

8. Você compreendeu a função do procedimento? \*

*Mark only one oval.*

- Sim  
 Não  
 Mais ou menos

9. Avalie a estética da interface do jogo: \*

*Mark only one oval.*

|            |                       |                       |                       |                       |                       |              |
|------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|--------------|
|            | 1                     | 2                     | 3                     | 4                     | 5                     |              |
| Muito feio | <input type="radio"/> | Muito bonito |

10. Avalie a qualidade das músicas do jogo: \*

*Mark only one oval.*

|         |                       |                       |                       |                       |                       |       |
|---------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-------|
|         | 1                     | 2                     | 3                     | 4                     | 5                     |       |
| Péssima | <input type="radio"/> | Ótima |

11. Você achou o jogo desafiador? \*

*Mark only one oval.*

- Sim  
 Não

12. Você gostou de Salazar? \*

*Mark only one oval.*

- Sim  
 Não

13. Você continuaria jogando se houvessem mais fases? \*

*Mark only one oval.*

- Sim  
 Não  
 Talvez

14. Os painéis de instruções são esclarecedores e auxiliam na compreensão do jogo? \*

*Mark only one oval.*

- Sim  
 Não  
 Mais ou menos

15. Você achou confortável jogar este jogo em um aparelho móvel? \*

*Mark only one oval.*

- Sim  
 Não  
 Mais ou menos

16. Você tem alguma sugestão de melhoria para o jogo?

---

---

---

---

---



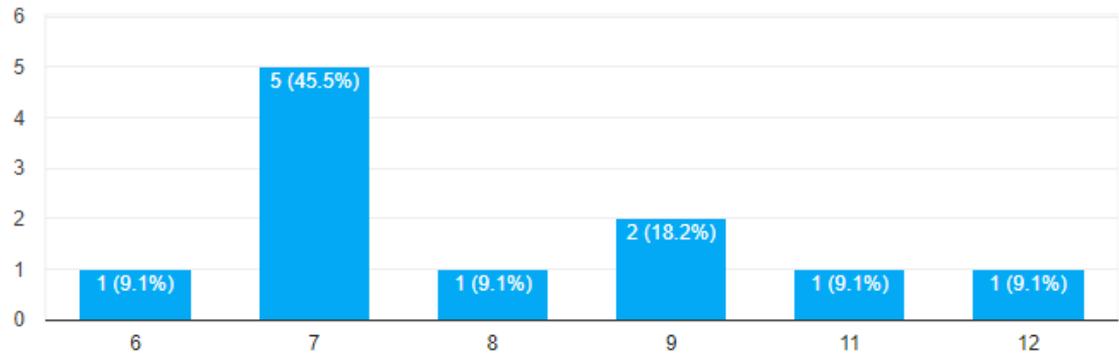
## **APÊNDICE B – RESPOSTAS COMPLETAS DE *PLAYTESTING***

A seguir, apresentamos um compilado dos resultados completos das etapas de *playtesting* realizadas durante o desenvolvimento deste projeto.



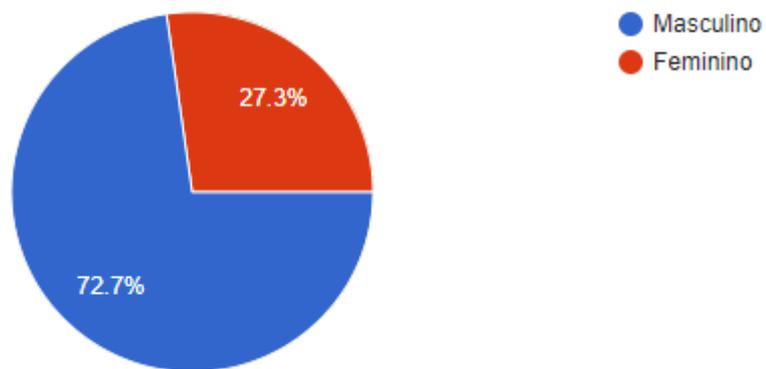
## Idade:

11 responses



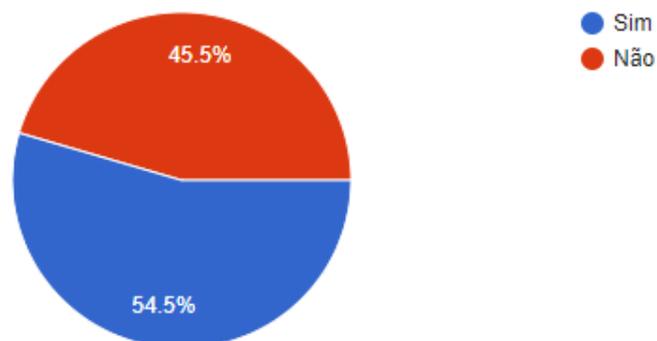
## Sexo:

11 responses



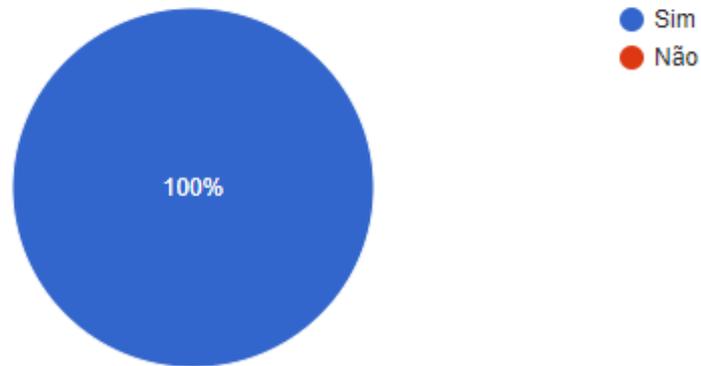
## Você tem algum conhecimento prévio de pensamento computacional?

11 responses



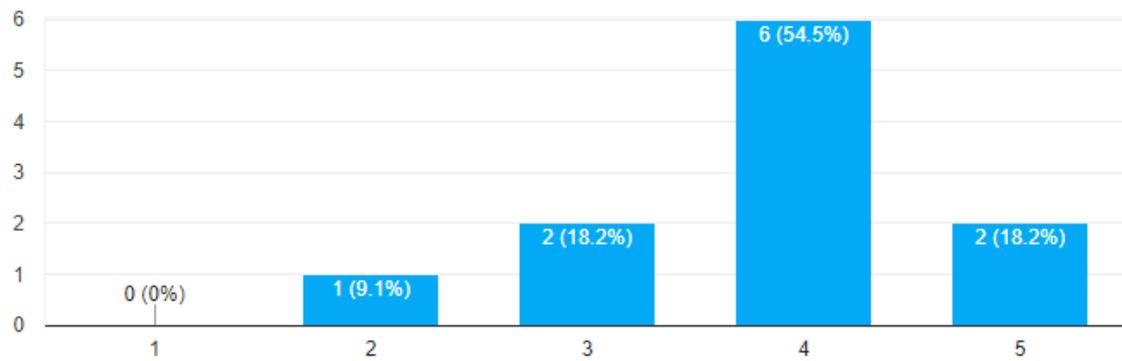
## O objetivo das fases do jogo é claro?

11 responses



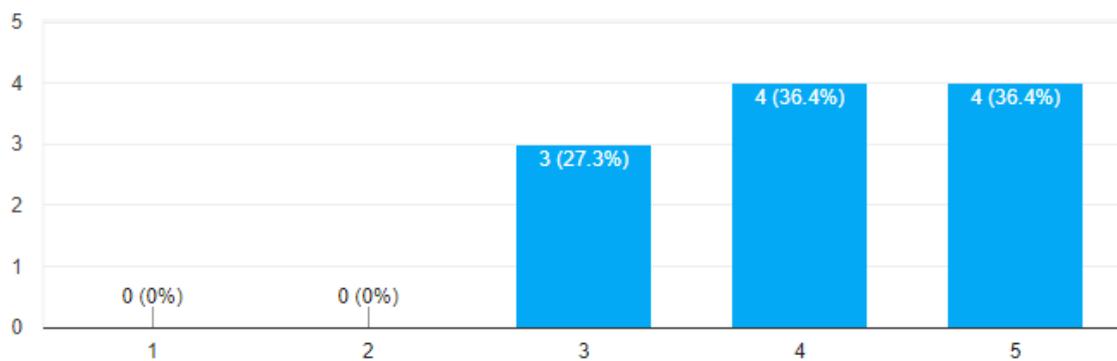
Avalie a facilidade de compreensão da jogabilidade do jogo para passar das fases:

11 responses



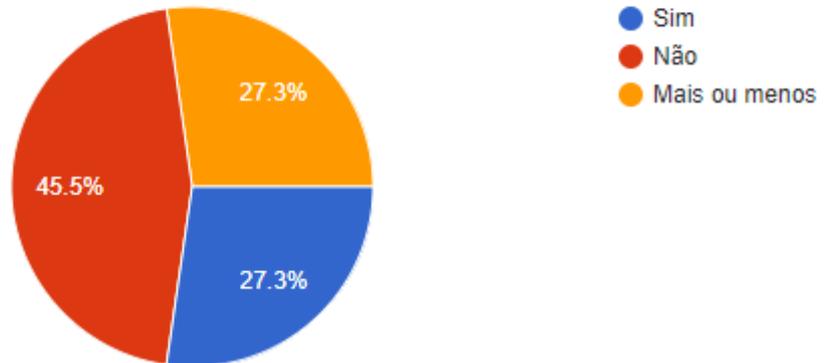
Avalie a facilidade de compreensão do uso de cada botão de comando:

11 responses



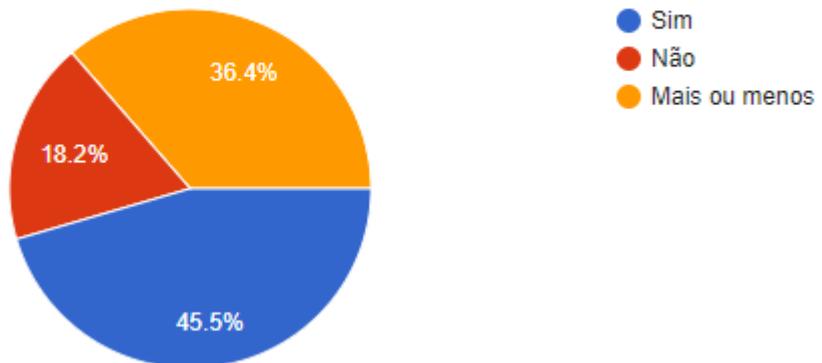
## Você compreendeu a função do loop?

11 responses



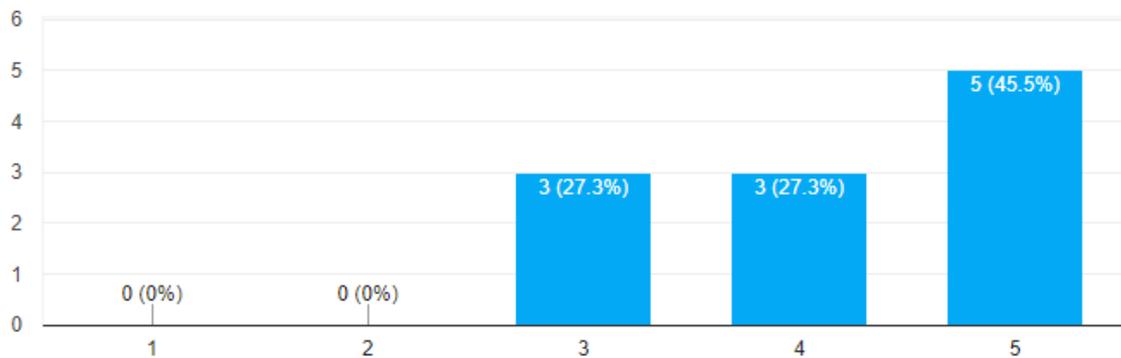
## Você compreendeu a função do procedimento?

11 responses



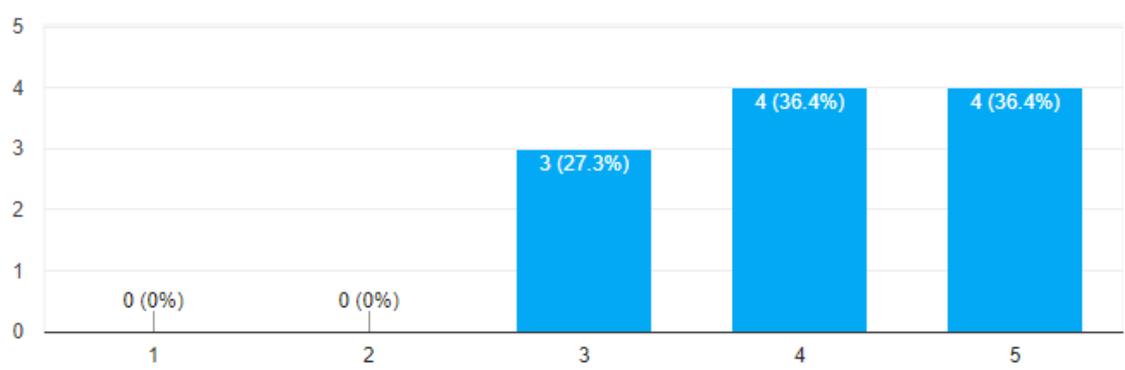
## Avalie a estética da interface do jogo:

11 responses



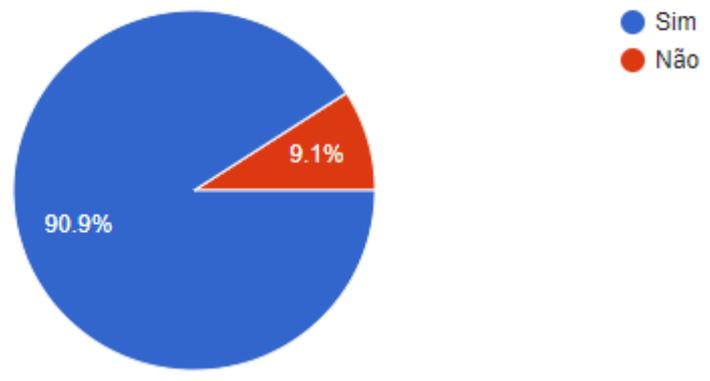
### Avalie a qualidade das músicas do jogo:

11 responses



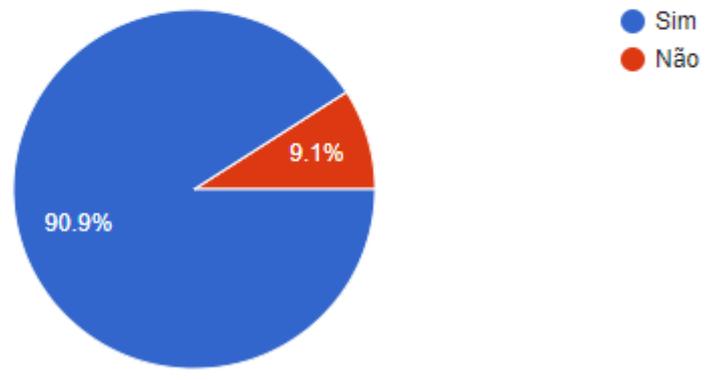
### Você achou o jogo desafiador?

11 responses



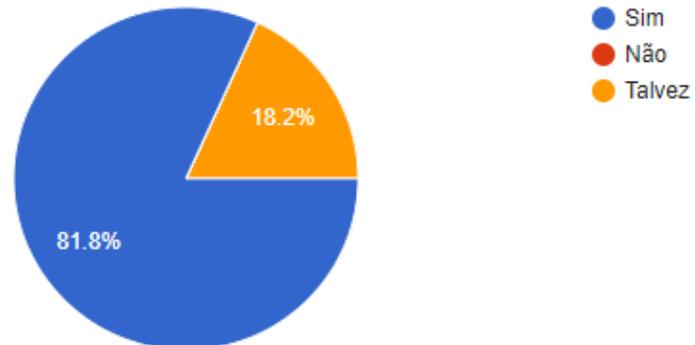
### Você gostou de Salazar?

11 responses



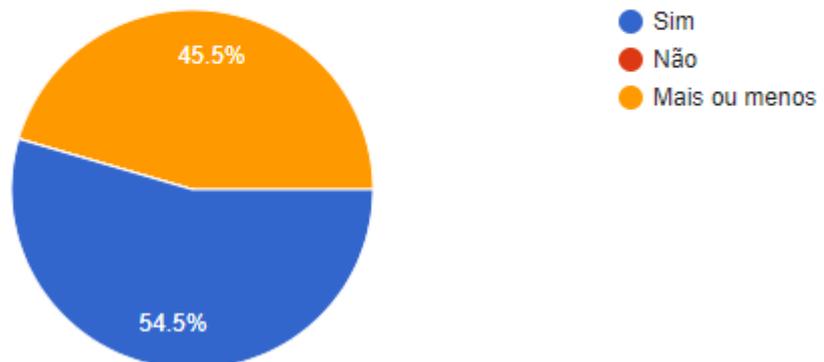
### Você continuaria jogando se houvesse mais fases?

11 responses



### Os painéis de instruções são esclarecedores e auxiliam na compreensão do jogo?

11 responses



### Você achou confortável jogar este jogo em um aparelho móvel?

11 responses



## Você tem alguma sugestão de melhoria para o jogo?

10 responses

O nome Salazar é muito difícil.  
No final ele tem que conseguir abrir o baú e pegar todos os seus brinquedos.

Complemento da resposta 2: Só o loop é complicado  
A fase 2 é muito difícil.

Complemento da resposta 6: A ponte tem que ser mais realista. Melhorar as imagens, deixar mais realista, em 3D.

Complemento da resposta 9: Gostei da história, gostei do nome, mas acho que o nome não chama muito a atenção.

Complemento da resposta 11: Ajuda muito!

Complemento da resposta 12: Sim, mas gostaria de jogar no computador também.

Botar uma mochila no personagem, com um monte de coisas dentro.

Adicionar inimigos no mapa.

O "repita" é muito difícil, e não conseguiu entender direito. Por isso não sabe se continuaria jogando caso houvessem mais fases.

Botar uns animais na tela.

Mais espaço na pilha de comandos

Melhorar gráficos

Deixar mais simples e mais intuitivo.