

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ  
DEPARTAMENTO ACADÊMICO DE COMPUTAÇÃO  
CURSO DE CIÊNCIA DA COMPUTAÇÃO

ANDRÉ RENATO BONIZI

**DESENVOLVIMENTO DE UM SERIOUS GAME PARA O ENSINO DE  
LÓGICA COMPUTACIONAL**

TRABALHO DE CONCLUSÃO DE CURSO

**MEDIANEIRA**

**2019**

ANDRÉ RENATO BONIZI

**DESENVOLVIMENTO DE UM SERIOUS GAME PARA O ENSINO DE  
LÓGICA COMPUTACIONAL**

Trabalho de Conclusão de Curso apresentado ao Departamento Acadêmico de Computação da Universidade Tecnológica Federal do Paraná como requisito parcial para obtenção do título de “Bacharel em Computação”.

Orientador: Prof. Me. Jorge Aikes Junior

**MEDIANEIRA**

**2019**



---

## **TERMO DE APROVAÇÃO**

### **DESENVOLVIMENTO DE UM SERIOUS GAME PARA O ENSINO DE LÓGICA COMPUTACIONAL**

Por

**ANDRÉ RENATO BONIZI**

Este Trabalho de Conclusão de Curso foi apresentado às 13:30h do dia 21 de novembro de 2019 como requisito parcial para a obtenção do título de Bacharel no Curso de Ciência da Computação, da Universidade Tecnológica Federal do Paraná, Câmpus Medianeira. O candidato foi arguido pela Banca Examinadora composta pelos professores abaixo assinados. Após deliberação, a Banca Examinadora considerou o trabalho aprovado.

---

Prof. Me. Jorge Aikes Junior  
UTFPR - Câmpus Medianeira

---

Prof. Dr. Pedro Luiz de Paula Filho  
UTFPR - Câmpus Medianeira

---

Prof. Me. Cesar Angonese  
UTFPR - Câmpus Medianeira

A folha de aprovação assinada encontra-se na Coordenação do Curso.

## RESUMO

BONIZI, André Renato. DESENVOLVIMENTO DE UM SERIOUS GAME PARA O ENSINO DE LÓGICA COMPUTACIONAL. 66 f. Trabalho de Conclusão de Curso – Curso de Ciência da Computação, Universidade Tecnológica Federal do Paraná. Medianeira, 2019.

Cada aluno pode ter uma velocidade de aprendizado diferente, o método tradicional de aprendizado não pode explorar todo o potencial dos alunos acima da média e pode até desestimular os alunos abaixo da média, forçando-os a seguir o ritmo de outros alunos, o que não é sempre possível de se realizar. Algumas ferramentas estão surgindo no campo da educação para auxiliar nesta dificuldade são os jogos eletrônicos, que podem ser vistos com frequência na rotina de um aluno como uma forma de entretenimento, mas não como uma forma educativa. O uso de jogos eletrônicos permite que o conteúdo esteja presente para o aluno e seja desenvolvido de forma individual, além de permitir o armazenamento do progresso do aluno, viabilizando a aplicação de uma ferramenta de avaliação automática. Este trabalho tem como o objetivo apresentar o desenvolvimento de um serious game que possa ser utilizado para auxiliar no aprendizado de lógica computacional. O serious game foi desenvolvido utilizando o MDA Framework com a ferramenta de motor gráfico Unity Engine. De maneira a realizar uma avaliação do protótipo foi utilizado um formulário de avaliação respondido por um público misto, com e sem conhecimentos de lógica computacional. O jogo educacional segue o gênero RPG clássico e conta com recursos de texto para transmitir a informação necessária para progredir durante o jogo, utilizando-se de recursos visuais como objetos tridimensionais para exemplificar conceitos de lógica computacional de forma prática. Com base nas avaliações percebeu-se que o protótipo foi bem recebido pelo público, deixando claro que sua aplicação como recurso educacional é viável.

**Palavras-chave:** educação, lógica, jogo

## ABSTRACT

BONIZI, André Renato. DEVELOPMENT OF A SERIOUS GAME FOR COMPUTATIONAL LOGIC TEACHING. 66 f. Trabalho de Conclusão de Curso – Curso de Ciência da Computação, Universidade Tecnológica Federal do Paraná. Medianeira, 2019.

Each student can have a different learning speed, learning in the traditional way cannot explore the full potential from the students who are above average and it can even discourage the students who are below average, forcing them to follow other students rhythm, what is not always possible to accomplish. One of the tools that are emerging to help dealing with some of these difficulties found in the field of education are the video games, which often can be seen in the routine of a student as a form of entertainment, but not as an educational tool. Video games allows that the content can be shown individually to each student, besides allowing the storage of the students progress, it can also make it viable to produce a automatic evaluation tool. This work has as its objective to present the development of a serious game that can be used to help in learning about computer logic. The serious game was developed using MDA Framework with the graphic engine tool Unity. To evaluate the prototype an evaluation form was answered by a mixed group, some with previous knowledge about computer logic. The serious game follows the model of classic RPG genre and it uses text information to transmit the necessary information to advance in the game, using visual resources such as tree dimensional objects to exemplify computer logic concepts. Using the evaluation results as a reference it can be perceived that the prototype was well received by the group, reinforcing the viability of the application as an educational tool.

**Keywords:** education, logic, game

## **AGRADECIMENTOS**

Gostaria de agradecer a Flávia Manente,  
que esteve ao meu lado e me deu apoio incondicional,  
alguém que nunca vou esquecer e que amei verdadeiramente.

Ao Jorge, meu orientador,  
que me ajudou incansavelmente,  
que não só me ensinou e ajudou na minha formação,  
mas foi um grande mestre,  
superou minhas expectativas e me trouxe inspiração.

Aos voluntários que participaram do meu trabalho,  
que me deram seu tempo,  
e sua dedicação,  
para que eu pudesse realizar minha contribuição.

“As pessoas perdem muito tempo.  
Então querem mais.

Mais horas em seus dias.  
Mais dias em seus anos.  
Mais anos em suas vidas.

Por que se tivessem todo esse tempo extra,  
elas poderiam consertar qualquer erro!

Eu não preciso de horas,  
nem dias,  
nem anos.  
Só preciso de segundos!

Esse é o lance do tempo.  
Se você não pode aproveitar ao máximo todos os momentos,  
não merece nem um segundo a mais!”

Ekko

## LISTA DE FIGURAS

FIGURA 1	– Diagrama de Metodologia de Desenvolvimento .....	31
FIGURA 2	– Caixa de diálogo contendo os comandos básicos .....	37
FIGURA 3	– Caixa de diálogo citando o termo Variável .....	38
FIGURA 4	– Interação com estrutura de condicional simples .....	39
FIGURA 5	– Interação com estrutura condicional de mensagem descritiva .....	40
FIGURA 6	– Operador Ativado .....	41
FIGURA 7	– Leitor Ativado .....	43
FIGURA 8	– Jogador interagindo com o Declarador .....	44
FIGURA 9	– Inimigo em estado de espera .....	44
FIGURA 10	– Interação com uma estrutura de loop .....	45
FIGURA 11	– Robô emitindo luzes azuis e operador de comando no loop .....	46
FIGURA 12	– Estrutura de condicional simples satisfeita .....	48
FIGURA 13	– Personagem principal atacando o inimigo .....	48
FIGURA 14	– Modelo do personagem principal. ....	50
FIGURA 15	– Modelo dos inimigos .....	51
FIGURA 16	– Primeiro desafio do jogador .....	52
FIGURA 17	– Gráfico de Resultados da Pergunta 1 .....	55
FIGURA 18	– Gráfico de Resultados da Pergunta 2 .....	56
FIGURA 19	– Gráfico de Resultados da Pergunta 3 .....	57
FIGURA 20	– Gráfico de Resultados da Pergunta 4 .....	58
FIGURA 21	– Gráfico de Resultados das Perguntas 5 a 8 .....	59
FIGURA 22	– Gráfico de Resultados da Pergunta 9 .....	60
FIGURA 23	– Gráfico de Resultados da Pergunta 10 .....	61
FIGURA 24	– Gráfico de Resultados da Pergunta 11 .....	61
FIGURA 25	– Gráficos de resultados da pergunta 12 .....	62



## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>9</b>
1.1	OBJETIVOS GERAL E ESPECÍFICOS	10
1.2	ORGANIZAÇÃO DO DOCUMENTO	10
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA</b>	<b>11</b>
2.1	TÓPICOS DE LÓGICA DE PROGRAMAÇÃO	11
2.1.1	Variáveis	11
2.1.2	Tipos de Dados	12
2.1.3	Estrutura Condicional	12
2.1.4	Laços de Repetição	13
2.2	ABORDAGENS E MÉTODOS DE ENSINO	14
2.2.1	Abordagem tradicional	14
2.2.2	Abordagem Comportamentalista	15
2.2.3	Abordagem Humanista	16
2.2.4	Abordagem Cognitivista	16
2.2.5	Método de Montessori	17
2.2.6	Métodos por Plano de Trabalho	17
2.2.7	Desafios	18
2.3	JOGOS NA EDUCAÇÃO	19
2.3.1	Introdução aos Serious Games	20
2.3.2	Benefícios	20
2.3.3	Trabalhos Relacionados	21
2.3.4	Gamificação	22
2.4	CONCEITOS DE GAME DESIGN	23
2.4.1	Perfil de Jogadores	24
2.4.2	Teoria do Fluxo	24
2.4.3	História e Contextualização	25
2.4.4	MDA Framework	26
2.5	ENGINES DE DESENVOLVIMENTO	27
<b>3</b>	<b>MATERIAS E MÉTODOS</b>	<b>29</b>
3.1	MATERIAIS	29
3.1.1	Unity Engine	29
3.1.2	Visual Studio Code	30
3.1.3	Windows	30
3.1.4	Google Forms	30
3.2	MÉTODOS	30
3.2.1	Planejamento	31
3.2.2	Desenvolvimento	32
3.2.3	Avaliação	32
<b>4</b>	<b>RESULTADOS</b>	<b>33</b>
4.1	MECÂNICAS	33
4.1.1	Movimentação	33

4.1.2	Interação	34
4.1.3	Batalha	35
4.2	DINÂMICAS	35
4.2.1	Controles de Dados	36
4.2.2	Dicas - Fonte de Conhecimento	36
4.2.3	Variáveis	37
4.2.4	Estruturas Condicionais	38
4.2.5	Operadores	40
4.2.6	Tipos de Dados	41
4.2.7	Leitores	42
4.2.8	Declaradores	42
4.2.9	Inimigos	43
4.2.10	Laços de Repetição (Loops)	45
4.3	ESTÉTICA	47
4.3.1	Iluminação	47
4.3.2	Batalha	47
4.3.3	História	49
4.3.4	Personagens e Assets	50
4.4	ESTRUTURA DE NÍVEIS	52
4.5	AVALIAÇÃO	53
<b>5</b>	<b>CONCLUSÕES</b>	<b>63</b>
5.1	PRINCIPAIS CONTRIBUIÇÕES	63
5.2	LIMITAÇÕES	64
5.3	TRABALHOS FUTUROS	64
	<b>REFERÊNCIAS</b>	<b>65</b>

## 1 INTRODUÇÃO

O raciocínio lógico tem um papel muito importante na ciência da computação, a resolução de problemas computacionais depende de uma sequência lógica de etapas que em conjunto representam a solução de um determinado problema. Mas algumas pessoas que não estão acostumadas com o ambiente computacional podem ter dificuldades em visualizar esse conjunto de partes necessárias para desenvolver soluções.

Podem existir diversos desafios na aprendizagem do pensamento computacional, dentre eles pode-se citar os métodos tradicionais de ensino, onde alguns estudantes podem enfrentar dificuldades por apresentarem particularidades que não são contempladas nas metodologias aplicadas (FRANÇA; TEDESCO, 2015).

Outra situação que pode ser entendida como um obstáculo no processo de aprendizagem é quando a variedade de informações adquiridas sobre o conteúdo é priorizada, nesse cenário os alunos peculiares podem não desenvolver a formação do raciocínio lógico por completo (MIZUKAMI, 1986).

Para manter os estudantes interessados no conteúdo proposto, recomenda-se que o estudo ocorra de forma dinâmica e variada buscando evitar o crescimento de graus de desatenção, o que pode ocorrer caso um recurso considerado estimulante seja utilizado de forma continuamente repetitiva (RANGEL, 2006).

Existem inúmeras formas de dinamizar o método de aprendizagem, dentre elas, pode-se destacar o uso de jogos digitais, que vem sendo apresentados como um fenômeno emergente por estimular os estudantes como uma forma prazerosa de estudo e potencializar a resolução de problemas em diversas áreas do conhecimento (FARDO, 2013).

Os jogos eletrônicos estão muito presentes no cotidiano dos jovens no contexto da pós-modernidade, por esse motivo é interessante adequar as metodologias para que essas ferramentas possam auxiliar os indivíduos na aprendizagem e deixem de ser interpretadas apenas como distrações ou entretenimento (NAVARRO, 2013).

## 1.1 OBJETIVOS GERAL E ESPECÍFICOS

Esse trabalho tem como objetivo desenvolver um serious game, capaz de estimular e validar o aprendizado de lógica de programação. Esse objetivo principal pode ser dividido nos seguintes objetivos específicos:

- Definir uma método de desenvolvimento para o serious game;
- Desenvolver um plano de ensino para os tópicos de lógica de programação básica;
- Desenvolver módulos contextualizados para trabalhar o conteúdo dos tópicos;
- Avaliar o serious game.

## 1.2 ORGANIZAÇÃO DO DOCUMENTO

Esse documento será organizado da seguinte forma. O Capítulo 2 apresentará inicialmente um breve resumo dos tópicos abordados no ensino de lógica de programação que serão contemplados na plataforma que será desenvolvida seguido de algumas abordagens e metodologias de ensino utilizadas atualmente e suas características principais, em seguida, são apresentados os trabalhos correlatos recentes, com o intuito de situar este trabalho no estágio atual do conhecimento, apresentando os benefícios e desafios encontrados, posteriormente, serão discutidos alguns conceitos utilizados para auxílio na criação de jogos e no final do capítulo serão citadas algumas ferramentas utilizadas no desenvolvimento de jogos. Os materiais e métodos utilizados se encontram no Capítulo 3, nele são descritas todas as etapas para o desenvolvimento do projeto assim como as ferramentas utilizadas. O Capítulo 4 apresenta os resultados do desenvolvimento do protótipo e uma avaliação inicial. Por fim, no Capítulo 5 encontra-se a conclusão da proposta do trabalho de conclusão de curso.

## 2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo serão discutidos os estudos relacionados com os temas presentes no desenvolvimento da plataforma, dentre as seções principais estão destacadas os fundamentos de programação, as diferentes abordagens de ensino e os estudos relacionados com o uso de jogos na educação.

### 2.1 TÓPICOS DE LÓGICA DE PROGRAMAÇÃO

Nesse trabalho serão focados os tópicos básicos no planejamento do ensino de lógica de programação, os tópicos discutidos tem como foco o aprendizado do pensamento computacional.

#### 2.1.1 Variáveis

Um dos conceitos básicos necessários para que se estruture um programa é o conceito de variáveis, tipicamente o primeiro tema apresentado nas aulas de programação. Para entender esse conceito é importante o entendimento de que dentro de um programa existem diversos valores que serão trabalhados durante a execução, quando um valor permanece igual desde o momento que o programa é iniciado até o fim de sua execução ele é denominado “constante”, mas quando o valor muda de acordo com os processos presentes no programa ele é denominado de “variável”.

Uma variável se comporta como um identificador do valor desejado e preferencialmente deve ter um nome em que seja de fácil entendimento sobre o que

aquele valor representa.

Dependendo da linguagem de programação uma variável pode ter diferentes tipos de valor, os mais comuns encontrados em linguagens de alto nível são os valores numéricos que podem ser inteiros e/ou reais normalmente se dividindo em diferentes categorias dependendo de sua complexidade. Também é comum a utilização de variáveis do tipo caractere que são usados para representar uma letra ou elemento de um texto e por último as variáveis do tipo lógico que representam um valor que pode assumir apenas dois estados: o de verdadeiro ou falso (AGUIAR, 2008).

### 2.1.2 Tipos de Dados

Quando se declara uma variável, é necessário especificar qual será o tipo de dado do qual essa variável representará, existem inúmeros tipos de dados dependendo de linguagem utilizada, entres os tipos mais frequentemente utilizados pode-se destacar os tipos que estão na categoria dos numéricos, na categoria de caractere e na categoria dos lógicos.

A categoria de numéricos representa os números reais, podendo ser inteiros ou decimais, positivos ou negativos e até nulos.

A categoria dos caracteres representa símbolos que serão apresentados para o usuário, podendo conter os símbolos alfabéticos de forma geral como letras, palavras ou até frases, também sendo compostos por numerais inteiros, decimais ou nulos e podendo também representar símbolos lógicos de operações.

A categoria dos lógicos representa os símbolos que compõem uma expressão, podendo conter um operador ou valor. Os operadores representam uma relação entre os valores (MANZANO; OLIVEIRA, 2005).

### 2.1.3 Estrutura Condicional

Outro conceito muito importante da programação é o de condicional, que na maioria das linguagens é representado pela função SE (do inglês, IF) que é utilizado para determinar como um programa será processado baseado na condição que é definida através de uma

expressão lógica.

Uma condicional deve conter uma instrução que realiza um teste onde será feita uma comparação de valores. O valor testado irá encaminhar a execução do programa para um bloco de instruções, caso o teste seja verdadeiro, ou encaminhará para outro bloco que é criado de forma opcional caso o teste seja falso.

Para que os desvios da condicional sejam encaminhados corretamente os blocos devem ser mapeados. O início do bloco de instrução, caso a condicional seja verdadeira, deve ser indicado logo após a expressão lógica de teste. Caso o programador decida incluir um bloco para uma expressão lógica seja falsa, ele deve definir uma indicação ao final do bloco de instrução anterior utilizando a função SENA (do inglês, ELSE) e incluindo as indicações de mapeamento de início e fim do bloco de instruções (MANZANO; OLIVEIRA, 2005).

#### 2.1.4 Laços de Repetição

Dependendo da linguagem podem existir diferentes tipos de laços de repetição, representados por diversas funções. O conceito é similar ao de um desvio condicional, onde é necessário definir uma expressão lógica para testar um valor que determine como ocorrerá a execução de um programa, mas nesse conceito a condição irá determinar se o bloco de instruções irá ser executado novamente, ou não.

A função de repetição mais simples é a ENQUANTO (do inglês, WHILE) que é estruturada da forma mais simples possível, caracterizada pela expressão condicional e seguida pelo bloco de instruções com uma indicação mapeando o início e o fim da repetição desejada. A condição é testada e caso o valor seja verdadeiro o bloco de instruções é executado até o fim, em seguida a condição é testada novamente até que o valor seja falso. É importante garantir que uma variável presente na expressão de condição tenha seu valor alterado dentro do bloco de instruções da repetição, caso isso não ocorra o programa entrará em uma repetição infinita e nunca será finalizado.

Para garantir que um laço não repita infinitamente existem funções que são utilizadas quando o número de repetições desejadas é conhecido ou pré-definido, para esses casos a função mais comumente utilizada é a PARA (do inglês, FOR), nessa função a expressão lógica condicional é dividida em três instruções: a de atribuição inicial da variável; a condição de parada; o incremento da variável em cada iteração (AGUIAR, 2008).

## 2.2 ABORDAGENS E MÉTODOS DE ENSINO

Nessa seção serão descritas metodologias de ensino seguido de uma comparação entre elas, buscando fundamentar as quais podem ser adaptadas ao uso de serious games para o ensino de lógica de programação.

Existem diferentes tipos de abordagens que podem ser utilizadas em diversos cenários. Cada uma pode favorecer ou desfavorecer o aprendizado dependendo da situação ou das peculiaridades do estudante, a seguir serão apresentadas algumas dessas abordagens.

### 2.2.1 Abordagem tradicional

A mais comumente utilizada nas escolas é a metodologia da aula expositiva onde o professor é o participante ativo que detém o conhecimento e passa a informação para os estudantes da maneira que achar pertinente. Nesse cenário o aluno tem uma participação predominantemente passiva no processo de aprendizagem, onde na maioria dos casos o seu papel é de ouvir, ou ler, buscando armazenar o máximo de informação possível. O método de avaliação normalmente se faz com o aluno tendo que reproduzir um processo ou método apresentado durante a aula ou simplesmente expondo a informação memorizada de maneira escrita ou oral. O método de ensino pode variar dependendo da escolha do professor, que pode optar por diferentes maneiras de apresentar e/ou trabalhar o conteúdo (MIZUKAMI, 1986).

A abordagem tradicional é muito utilizada como um método aplicado à grupos por se tratar predominantemente de um ensino coletivo, podendo variar entre exposição do professor, leituras orientadas, debates e apresentações elaboradas pelos próprios alunos. É comum que os estudantes tenham preferência em esclarecer suas dúvidas com os colegas de classe que tiverem maior afinidade. Essa coletividade no aprendizado pode agir como motivação e engajamento entre os estudantes, em casos onde esse engajamento coletivo não está presente o aluno pode se sentir excluído dos demais estudantes e isso poderá causar uma desmotivação no aluno excluído prejudicando o desenvolvimento do aprendizado (RANGEL, 2006).

Garantir que todos os alunos presentes durante a aula adquiram o mesmo conhecimento ou sejam capazes de reproduzir o conteúdo de maneira eficaz pode ser um desafio por conta de que cada estudante pode apresentar diferentes facilidades ou dificuldades de aprendizado.



Dentre essas peculiaridades pode-se citar alunos que tem facilidades auditivas, ou até os que possuem facilidade visual, dependendo do método de ensino escolhido cada um desses perfis de alunos pode ter seu processo de aprendizagem favorecido ou desfavorecido (MIZUKAMI, 1986).

### 2.2.2 Abordagem Comportamentalista

Nessa metodologia o ambiente precisa ser previamente preparado para o processo de aprendizagem, levando em consideração as diferentes situações que podem se desencadear de acordo com as ações do aluno. Para que o ambiente seja controlado é necessário que o mesmo seja testado de diversas formas antes de colocado em prática como ambiente de ensino. Nessa abordagem destaca-se o conhecimento como forma de manipulação do ambiente inserido para que se possa atingir os objetivos propostos, onde a experiência e a descoberta são elementos fundamentais no processo de aprendizagem.

Nesse cenário o professor é colocado como desenvolvedor do sistema ou tecnologia utilizado no processo de aprendizagem e como monitor para auxiliar caso o aluno tenha dificuldades em resolver os problemas propostos, mas nessa abordagem deve-se evitar ao máximo transmitir o conhecimento adquirido para não suprimir o estímulo causado pela descoberta, que é o foco dessa abordagem. Um dos desafios dessa metodologia é que o aprendizado se torna fortemente autônomo, e caso o estudante se torne desmotivado por algum fator exterior pode acabar comprometendo todo o processo de aprendizagem (MIZUKAMI, 1986).

A abordagem comportamentalista é caracterizada por ser predominantemente individual e ativa, onde a maioria das atividades são desenvolvidas pelo aluno e, caso o mesmo possuir dúvidas ou dificuldades, pode requisitar o auxílio do professor. Esse método está relacionada com o desenvolvimento da autonomia, iniciativa e independência do estudante e para que sua implementação seja viável é necessário preparar o ambiente de estudo com tarefas e resoluções separadas em pequenas etapas para que o estudante possa desenvolver essas atividades de maneira auto-suficiente, onde os princípios da aprendizagem são baseados na auto-reflexão, reinício e autocorreção (RANGEL, 2006).

### 2.2.3 Abordagem Humanista

Semelhante à abordagem comportamentalista essa abordagem tem a autodescoberta e as experiências do aluno como ponto principal, mas nesse caso dá-se ao mesmo a responsabilidade do aprendizado e avaliação do conhecimento e não se enfatiza técnica ou método utilizados nessa abordagem, pois requer o conhecimento do aluno como ser humano levando em conta suas experiências prévias e os sentimentos que ele possui. Nessa abordagem o professor se posiciona como um facilitador do processo de aprendizagem proporcionando um clima favorável para o desenvolvimento do aluno, promovendo a liberdade de aprendizado da forma como o aluno se adaptar melhor. A autonomia do aluno é de grande importância, o que pode desencadear resultados positivos como a satisfação do aluno dentro do processo de aprendizagem é facilmente atingida, mas por outro lado a subjetividade da avaliação do conhecimento pode ser frequente (MIZUKAMI, 1986).

É necessário enfatizar a influência do meio social e cultural nessa abordagem para determinar o processo de ensino em que o estudante será inserido, pois essas intervenções podem causar grandes alterações no resultado do desenvolvimento das atividades (BASTOS et al., 2015).

### 2.2.4 Abordagem Cognitivista

Essa abordagem é caracterizada pelas etapas de uma estrutura que corresponde ao processo de aprendizagem, onde o conceito de ensino é dado de forma incremental, ou seja, quando o estudante atinge uma certa etapa da estrutura de ensino supõe-se que o mesmo tenha adquirido o conhecimento necessário para desenvolver a etapa atual por ter sido capaz de desenvolver as etapas anteriores.

Nesse cenário é importante enfatizar o esforço e a tomada de decisões do aluno, onde a variedade de informações adquiridas pode não ser relevante caso o indivíduo seja capaz de desenvolver soluções mesmo com um conhecimento limitado.

A organização do processo de aprendizagem é fundamental nessa abordagem, onde o professor é responsável por apresentar situações que podem desestruturar a linha de raciocínio do estudante para criar frequentemente um nova forma de pensamento (MIZUKAMI, 1986).

### 2.2.5 Método de Montessori

O método de Montessori tem como o foco a utilização predominante de objetos concretos e a associação da manipulação desses objetos aos conceitos que devem ser adquiridos. Dessa forma é necessário levar em conta a estrutura corporal do aluno, que em sua maioria é formada por crianças, por esse motivo os objetos concretos devem ser de fácil alcance e manuseio, onde o ambiente deve ser preparado para que a autonomia do aluno possa ser desenvolvida por completo. A abordagem tradicional está pouco presente ou quase inexistente quando se trabalha com esse método, principalmente tratando-se do espaço físico, este último deve promover a livre circulação e criatividade, e quando se trata da avaliação realizada pelo professor, onde, em Montessori, deve-se exemplificar os processos que estão relacionados ao conteúdo da aula sem definir qual processo possui maior eficácia para atingir os objetivos necessários, deixando para que o aluno possa escolher qual dos processos deverá ser utilizado para adquirir resultados mais eficazes através da tentativa e erro, a escolha do processo é identificada como forma de autoavaliação e não deve sofrer influência do educador (RANGEL, 2006).

Os conceitos abstratos devem ser evitados, onde as atividades não podem estar limitadas apenas a observação e devem conter experimentação como base desse método. A comunicação possui um aspecto fundamental quando se trata da avaliação, que pressupõe o aluno sendo capaz de relacionar os objetos concretos com o nome correspondente (MONTESSORI, 2004).

### 2.2.6 Métodos por Plano de Trabalho

Na utilização de métodos por plano de trabalho o estudante desenvolve as áreas de seu interesse, predominando a escolha de temas de estudo individualmente e o professor apresenta materiais que oferecem suporte ao desenvolvimento desse tema. Nesse método pode-se destacar o plano dalton, sistema winnetka e o plano por centros de interesse (RANGEL, 2006).

O plano dalton se caracteriza pela organização de tarefas para auxiliar o estudo, essas tarefas são classificadas em três níveis de dificuldade: fácil, médio e difícil. As características de avaliação das tarefas são determinadas de acordo com o seu nível de dificuldade. Esse plano dá suporte para que o aluno vá avançando gradativamente para níveis superiores de

acordo com o seu progresso. O sistema winnetka tem como característica principal a divisão do estudo por núcleos temáticos, que são compostos por tarefas de atividades coletivas onde os alunos formam grupos e tem que interagir entre si para solucionar os problemas relacionados ao núcleo temático. Os centros de interesse são similares ao sistema winnetka por sua organização ser baseada em temas de estudo onde nesse método são focadas as necessidades básicas do ser humano incluindo alimentação, abrigo, saúde e comportamento social. Todos os métodos citados nesta seção tem enfoque sobre o interesse nos temas e a diversificação de atividades por meio de um plano de trabalho (RANGEL, 2006).

### 2.2.7 Desafios

Existe uma variedade de desafios na aprendizagem de maneira geral, a seguir serão citadas algumas das dificuldades encontradas nas abordagens tipicamente utilizadas em sala de aula.

Um dos maiores desafios é identificar a linha de raciocínio dos estudantes, os problemas podem ser resolvidos de diversas maneiras e pode não ser muito claro para o professor o motivo pelo qual o aluno decidiu desenvolver uma solução partindo de um processo diferente do tradicional. Suprimir diferentes formas de resolução pode acabar por desestimular o aluno e fazê-lo acreditar ser incapaz de raciocinar de maneira lógica e coerente.

Frequentemente os conteúdos são ensinados de forma isolada, sem correlação com outros assuntos, dificultando a associação de conteúdos correlacionados e inviabilizando a relação do conteúdo adquirido com situações reais do cotidiano (GOMES; MELO, 2005).

Outro fator que pode ser identificado como um desafio na aprendizagem é quando o estudante desenvolve um método ou processo próprio para a resolução de um determinado problema que, apesar de ser funcional e/ou satisfatório, pode ser suprimido pelo professor por possuir uma preferência, normalmente baseado na experiência, de um método ou processo específico. Nesses casos, mesmo que o professor tenha intenções de guiar o aluno para um caminho que traga mais resultados, alguns alunos podem se sentir desmotivados ou que não possuem competência para desenvolver o raciocínio lógico por completo (KOSLOSKY, 1999).

Os alunos que não dominam a lógica de programação e não são estimulados a estudar fora da sala de aula tem uma tendência a enfrentar grandes dificuldades de aprendizado o que torna o processo de ensino altamente frustrante, por consequência existe um elevado número de desistentes em cursos relacionados à computação (LIMA; MEIRELES, 2014)

Muitos estudantes ao se deparar com o estudo de solução de problemas computacionais pela primeira vez sentem-se incapazes de programar por ainda não terem desenvolvido habilidades necessárias na disciplina como raciocínio lógico, capacidade de abstração e habilidades matemáticas, por consequência os cursos relacionados à computação e informática sofrem com altos índices de reprovação e desistência. Além disso como frequentemente lida-se com alunos que possuem diferentes graus de conhecimento, é necessário que as ferramentas de ensino trabalhem de acordo com o conteúdo prévio que o aluno recebeu de forma individual para que este possa adaptar-se ao conteúdo novo com maior facilidade, ensinando o aluno de acordo com o seu grau de conhecimento (NETO; SCHUVARTZ, 2007).

Além desses fatores, também pode-se identificar o medo de se expor, que pode prejudicar a maioria dos alunos, principalmente aqueles que não estão familiarizados com o ambiente ou com os demais estudantes, o que pode influenciar no momento de realizar apresentações ou até no esclarecimento de dúvidas com o professor ou com os alunos (JÚNIOR; RAPKIEWICZ, 2005).

No momento de apresentar o conteúdo a ser estudado, é necessário que o objetivo e o foco nas questões consideradas essenciais fiquem bem claros para o aluno, para que o mesmo entenda a importância de entender a informação apresentada. Isso pode não ocorrer quando o procedimento que direciona o estudo contém um grande quantidade de informações que fogem do enfoque principal do assunto abordado (RANGEL, 2006).

Os alunos possuem diferentes habilidades de uma forma geral. A maneira de avaliar o aprendizado adquirido pode favorecer habilidades específicas presentes em alguns estudantes colocando os que não possuem essas habilidades em desvantagem (NETO; SCHUVARTZ, 2007).

Um dos fatores pelo qual as abordagens tradicionais são criticadas é o isolamento de alunos perturbadores, como forma de solução para um problema sem a análise detalhada de sua causa. Para que um aluno que não se encaixa no modelo escolar tradicional deve-se levar em consideração a personalidade do aluno, assim como considerar seus antecedentes e sua relação familiar (PICADO, 2009).

### 2.3 JOGOS NA EDUCAÇÃO

Os jogos e seus elementos estão cada vez mais ganhando espaço na educação e em outras áreas para as mais diversas finalidades. Mas a maioria dos jogos é projetado para entreter

e pode não ser adequado para ser utilizado em sala de aula ou em contextos educacionais, por esse motivo surge a necessidade de criar ferramentas que possibilitem uma aprendizagem de maneira eficaz mantendo o foco no conteúdo proposto sem perder o aspecto que o torna atrativo e prazeroso.

Essa é a proposta dos Serious Games, que fogem do foco de entretenimento e são projetados para o ensino ou treinamento relacionado à algum assunto específico.

### 2.3.1 Introdução aos Serious Games

Serious games são jogos voltados para a educação ou treinamento, fugindo do paradigma de entretenimento os jogos são projetados para familiarizar o jogador com algum conceito, método ou processo específico (RITTERFELD et al., 2009).

Dentro de um ambiente virtual são desenvolvidas situações manipuláveis em que o jogador deverá tomar decisões envolvendo informações reais do cotidiano. Devido à variedade de temas que podem ser abordados por um serious game, essa ferramenta possui um enorme potencial em diversas áreas de trabalho (ABT, 1987).

### 2.3.2 Benefícios

Jogar videogames de um forma geral pode ter seus benefícios. Esses benefícios podem variar dependendo do gênero do jogo ou da forma como ele é apresentado.

Apesar de ser uma área de estudo relativamente nova, existem autores que defendem o uso de jogos digitais, em especial os serious games, como foco para a formação de profissionais da área de computação devido às habilidades que estes podem adquirir. Pode-se desenvolver o raciocínio lógico, ou então a capacidade de tomar decisões rápidas com informações limitadas, reconhecimento de padrões e interpretação (MELO et al., 2013).

O ambiente dos jogos permite organizar as relações entre as ideias e informações necessárias para cumprir tarefas. Desse modo é possível simplificar o desenvolvimento da lógica adquirida no ambiente virtual e aplicar no mundo real, desde de que as situações apresentadas nos jogos sejam analisadas para direcionar o estudante no caminho certo para

a solução de determinado problema (CASSENOTE; ANTONIAZZI, 2015).

A capacidade de resolver problemas computacionais é uma necessidade atual e o estudo com o foco em crianças pode resultar em uma rápida inserção no mercado de trabalho dessa área, além de estimular o interesse dos estudos à ingressar em uma carreira na área de estudo da computação (GARCIA et al., 2008).

Outro ponto positivo da aplicação jogos na educação é o desenvolvimento da autonomia que o jogador desenvolve por ter a possibilidade de avançar no roteiro do jogo sem a necessidade do monitoramento de terceiros. Ainda existe a possibilidade de trabalhar diferentes disciplinas de uma única vez, quando são incorporados elementos de diversas áreas em um único contexto do jogo (GUEDES; SILVA, 2012).

O uso de ambientes virtuais no ensino tornam o processo mais prazeroso e dinâmico, e ajudam na assimilação da teoria com a prática (NETO; SCHUVARTZ, 2007).

Os jogos estarão associados com métodos de ensino ativos, onde o jogador realizaria diversas tarefas através de pequenas etapas de forma sequencial, tendo a possibilidade de revisar as tarefas e reconstruí-las (RANGEL, 2006). Além de tornar do o processo de aprendizagem mais dinâmico, o uso de jogos permite adquirir o conhecimento de métodos que podem ser aplicados para a solução de problemas do mundo real (AGUIAR, 2008).

O uso de jogos também promove o aprendizado cooperativo, que ocorre quando um aluno percebe que o colega enfrenta dificuldades que ele já superou. É comum ver casos onde ocorre um auxílio entre os alunos (CASSENOTE; ANTONIAZZI, 2015). Esse cenário, além de ajudar o professor na função de ensinar seus alunos com dificuldade, proporciona um desenvolvimento nas habilidades sócio-emocional dos alunos.

### 2.3.3 Trabalhos Relacionados

Uma das ferramentas de sucesso criada com o propósito de tornar a prática de programação mais agradável e intuitiva para crianças é o Scratch<sup>1</sup>, que foi desenvolvido pelo Instituto de Tecnologia de Massachusetts (do inglês, Massachusetts Institute of Technology, MIT) e possui uma interface baseada em arrastar e soltar blocos de programação, tornando o processo mais agradável e ignora a necessidade do aluno em estar atento à sintaxe da linguagem de programação (OLIVEIRA et al., 2014).

Essa ferramenta foi produzida com o propósito de ensinar a lógica de programação

---

<sup>1</sup><https://scratch.mit.edu/>

para pessoas que não tiveram nenhum contato com o assunto. A ferramenta trabalha com os comandos de programação como se fosse um quebra-cabeças, podendo agrupar as peças para obter diferentes resultados associados à forma como as peças são agrupadas e organizadas, colocando o foco no funcionamento do projeto (CASSENTE; ANTONIAZZI, 2015).

O treinamento de lógica computacional utilizando o Scratch estimula atividade cerebral em diferentes partes do cérebro e auxilia no desenvolvimento do raciocínio lógico (BASTOS et al., 2015).

Outra ferramenta com o conceito semelhante ao Scratch, é a plataforma online Code Studio<sup>2</sup> que contém cursos que envolvem a lógica de programação mas com o conceito de que é não preciso ter um talento natural ou facilidade de aprendizado para saber programar, basta ter determinação. A plataforma, assim como o Scratch, também utiliza blocos de comandos que substituem as linhas de códigos de uma linguagem de programação e dependendo da maneira como são organizados e agrupados realizam funções que podem ser associadas às utilizadas em programação. Nesta plataforma existem cursos para crianças a partir de 4 anos, onde elas devem solucionar desafios de maneira progressiva até que o curso esteja concluído. Os cursos do Code Studio possuem uma qualidade gráfica alta se comparada aos demais jogos educativos e contam com referências com temáticas populares entre o público alvo como: Star Wars, MineCraft, Frozen, entre outros (ZANETTI et al., 2017).

Outro trabalho que apresenta similaridades é a utilização de Robôs Educativos para o ensino de Lógica de Programação, que agem como motivadores e facilitadores no processo de aprendizagem, colaborando para o entendimento de vários níveis de abstração necessários para que os conceitos trabalhados nesta disciplina tenham um bom desenvolvimento (CAMBRUZZI; SOUZA, 2015).

#### 2.3.4 Gamificação

O conceito de gamificação se baseia em incorporar elementos de jogos em rotinas de trabalho, ensino ou em treinamentos. O objetivo da gamificação é tornar as rotinas mais atrativas para gerar maior engajamento e tornar essas rotinas mais produtivas (ZICHERMANN; CUNNINGHAM, 2011).

Para se alcançar esse objetivo é necessário que o usuário ou cliente esteja constantemente em um estado de fluxo, que é caracterizado por estar entre os níveis de

---

<sup>2</sup><https://studio.code.org/courses>



ansiedade e tédio, pela maior quantidade de tempo possível.

Existem vários métodos de engajamento e o sucesso destes pode variar muito dependendo de cada situação. Pode-se destacar o Sistema de Recompensa como altamente difundido em processos de gamificação, o mais comum é o sistema de pontuação que determina o sucesso do jogador a partir dos pontos adquiridos pelo mesmo, mas em alguns casos os pontos podem não ser suficientes para engajar a maioria dos jogadores e muitos precisam de diferentes estímulos. Existem inúmeras alternativas e dentre elas destacam-se: os distintivos, que são uma maneira de diferenciar jogadores experientes e bem sucedidos dos iniciantes; o acesso, que permite jogadores que concluíram tarefas específicas o acesso à conteúdos diferenciados; O poder, que determina a influência da tomada de decisões de um jogador em relação aos demais, ou ao ambiente inserido; as coisas, que são caracterizadas por objetos presentes no jogo encontrados com frequência e que servem como uma maneira de interagir com o ambiente de forma mais concreta (ZICHERMANN; CUNNINGHAM, 2011).

## 2.4 CONCEITOS DE GAME DESIGN

Antigamente o processo para desenvolver um jogo era muito diferente, o projeto e a documentação contendo os requisitos do software eram muito bem definidos e passavam por uma série de discussões e aprovações até que finalmente o software contendo o jogo se tornaria algo concreto. Era comum a utilização do modelo em cascata, onde as ideias de funcionalidades eram incluídas na etapa inicial do projeto e dificilmente ocorriam mudanças após elas terem sido definidas, esse conceito sofreu muitas alterações ao longo do tempo. Atualmente é comum encontrar processos de desenvolvimento que utilizam o modelo em espiral, que prevê uma produção em grande quantidade de protótipos, que inicialmente são de muito baixa qualidade e vão sendo aprimorados para gerar versões mais detalhadas e próximas ao produto final, essa mudança de conceito tornou o processo de desenvolvimento mais dinâmico e barato (SCHELL, 2008).

### 2.4.1 Perfil de Jogadores

Além do projeto, existem muitas variáveis que podem afetar a receptividade de um jogo, e, conseqüentemente, o seu desenvolvimento. Uma questão que deve ser analisada antes de criar um jogo é definir a faixa etária que representa o público alvo dos jogadores, o que terá uma influência forte sobre o conteúdo que poderá ser apresentado no jogo, também pode ser informação valiosa para definir os recursos que serão utilizados, por exemplo, textos em grande quantidade não são recomendados quando se trata de um público infantil, outros casos como este devem ser levados em conta (SCHELL, 2008).

Frequentemente os jogos quando estão em fase de testes são avaliados por pessoas que têm uma certa experiência com jogos de uma forma geral, por este motivo conseguem facilmente reconhecer padrões que podem não ser nítidos para outras pessoas. Então, para obter resultados de avaliação mais abrangentes e completos é necessário realizar testes com uma variedade considerável de indivíduos com características divergentes. Também é preciso entender que existem diferentes perfis de jogadores e o aspecto atrativo de um jogo pode ser totalmente diferente dependendo desse perfil. Alguns perfis de jogadores possuem motivadores semelhantes e podem ser categorizados em quatro tipos, que são apresentados a seguir:

- Conquistadores, são aqueles que têm sua motivação ativada por objetivos e metas, que se sentem realizados ao cumprirem desafios;
- Exploradores, que sentem prazer em encontrar novas possibilidades e/ou caminhos, são estimulados através da descoberta do novo;
- Socializadores, são os que buscam estabelecer conexões com outros jogadores e são estimulados através da obtenção de relacionamentos estabelecidos dentro de um jogo;
- Matadores, são jogadores que apreciam a competição no seu nível mais alto, atingem a satisfação quando estabelecem uma relação de superioridade com os demais jogadores.

Construir um jogo que possa incluir esses quatro perfis pode ser desafiador mas poderá garantir uma gama extensa de jogadores (ZICHERMANN; CUNNINGHAM, 2011).

### 2.4.2 Teoria do Fluxo

Determinar se a abordagem presente no jogo será eficaz em motivar os jogadores é um

desafio. O desejado é manter o jogador em um estado de imersão constante, mas, de acordo com a teoria do fluxo, permanecer nesse estado depende do equilíbrio de dois parâmetros, a ansiedade e a habilidade. Quando o jogador não conseguir avançar no jogo, ou se os desafios apresentados estiverem além de suas habilidades, possivelmente ocorrerão frustrações que o levarão a abandonar o jogo e em contrapartida, se as habilidades do jogador estiverem altamente desenvolvidas tornarão os desafios do jogo triviais e levarão ao tédio (SCHELL, 2008).

Uma forma de evitar esse tipo de situação é aumentar a dificuldade gradualmente a cada meta ou objetivo atingido com sucesso. Aplicando esse conceito é possível controlar a progressão da dificuldade de maneira que seja lenta para jogadores inexperientes, pois estes terão maior dificuldade em atingir os objetivos iniciais, e rápida para os mais experientes, por estes serem capazes de realizar as metas iniciais com mais facilidade e rapidez.

Outra forma de possibilitar a imersão de diferentes níveis de jogadores é permitir escolha de dificuldade do jogo, permitindo que os menos experientes escolham dificuldades menores para não se frustrarem e os mais experientes tem a possibilidade de iniciar em uma dificuldade alta para aumentar o desafio.

Quando há diversas opções disponíveis no jogo, é possível proporcionar formas de jogabilidade que se caracterizam de duas maneiras, ousada, onde o jogador deve se arriscar mais para conseguir uma quantidade ou qualidade superior de prêmios caso tenha sucesso, e segura, onde o jogador pode evitar se arriscar mas, por consequência, os prêmios serão inferiores, essas formas são recomendáveis para integrar diferentes tipos de jogadores ao mesmo jogo e garantir a imersão de ambos (SCHELL, 2008).

### 2.4.3 História e Contextualização

Outra forma de estimular o jogador é contextualizar o jogo com uma história bem elaborada, existem inúmeros jogos bem sucedidos onde o roteiro é o único aspecto atrativo. A história do jogo pode ter diferentes elementos que a tornem interessantes. A escolha do tema ou fazer com que o jogador se identifique com algum personagem pode ser uma ótima forma de engajamento e garantia fidelidade com o jogo. A seguir serão comentados dois métodos de integração de uma história em um jogo que podem representar a maioria dos jogos:

- Método da Cadeia de Pérolas: Consiste em uma apresentação não-interativa da história do jogo, chamada de cadeia, podendo ser representada por um texto, uma sequência de imagens ou uma animação. Após essa apresentação, no final de uma cadeia, o jogador é

colocado em um ambiente de livre movimento e controle chamado de pérola onde pode-se agir e tomar decisões para cumprir um objetivo determinado. Caso seja cumprido, o jogador é retirado da pérola e volta para a cadeia onde é apresentada a sequência da história anterior até que é colocado novamente em uma pérola e esse processo se repete até que a história chegue ao fim. Esse método pode ser criticado por não ser totalmente interativo, mas os jogadores apreciam essa forma de jogo independentemente;

- **Método da Máquina de Histórias:** Para entender esse método é necessário compreender que qualquer série de eventos pode ser entendida como uma história, e baseado nisso esse método se sustenta de forma que o ambiente do jogo permita que sejam criadas uma série de eventos que desencadeiam uma nova série de eventos e assim sucessivamente. Esse método torna a história do jogo totalmente interativa, tornando o jogador no autor das histórias criando seus próprios objetivos.

O contexto do jogo pode ser utilizado como uma forma de limitar as opções do jogador, como uma forma de guiá-lo, agindo como um facilitador diminuindo a possibilidade de erros e aproximando-o do objetivo do jogo. Se as possibilidades forem muito abrangentes, o jogador pode se encontrar frequentemente em uma situação onde o número de opções é tão grande que a possibilidade de cometer erros se torna muito maior, tendendo a deixar o jogador frustrado (SCHELL, 2008).

#### 2.4.4 MDA Framework

Uma plataforma de desenvolvimento muito citada em trabalhos recentes é o MDA Framework, a sigla MDA representa as palavras:

- **Mechanics (Mecânica),** que é associado às funcionalidades dos componentes do jogo e pode ser entendido como as ações que o jogador pode realizar dentro do jogo. Pode-se tomar como exemplo a movimentação do personagem, que pode ocorrer de diferentes formas dependendo do estilo de jogo;
- **Dynamics (Dinâmica),** que está relacionado às interações no ambiente do jogo e como elas ocorrem. Pode ser entendido como o resultado da interação da mecânica com o ambiente atual do jogo. Pode-se tomar como exemplo de dinâmica a forma como os inimigos reagem às ações do personagem, se eles possuem inteligência artificial, se perseguem o personagem ou se possuem uma rota fixa de movimento;

- Aesthetics (Estética), que representa o modo como as interações da mecânica com a dinâmica terão impacto no jogador, está associado com a sensação gerada pelas interações presentes no jogo. Essas sensações normalmente são induzidas através de aspectos visuais e sonoros. Pode-se tomar como exemplo um jogo de terror, que possui uma estética voltada totalmente para gerar o medo no jogador, agora se for tomado como exemplo um jogo infantil, a estética é constituída de forma a deixar o jogador relaxado e confortável, o oposto dos jogos de terror.

o entendimento desses conceitos pode facilitar muito o a organização para o desenvolvimento de um jogo e na definição dos seus elementos e aspectos motivacionais (ZICHERMANN; CUNNINGHAM, 2011).

## 2.5 ENGINES DE DESENVOLVIMENTO

- Unity<sup>3</sup> é um dos motores gráficos mais famosos utilizados atualmente. Além de possuir opções para desenvolver jogos em duas ou três dimensões têm ênfase em portabilidade para diversas plataformas, motivo pelo qual se destacou. Entre as plataformas disponíveis podem ser citadas citar os sistemas operacionais Windows, Linux e Mac, e também para dispositivos móveis como Android, Blackberry, Windows Phone e IOS, além de consoles como Playstation e Xbox. Possui compatibilidade com as linguagens de programação C Sharp e Javascript e uma variedade de recursos disponíveis gratuitos e pagos para download. Possui uma das maiores comunidades de desenvolvimento de jogos, o que torna fácil o acesso a recursos pré-fabricados e material acessível para aprendizado da plataforma.
- Unreal Engine<sup>4</sup> é um motor gráfico em terceira dimensão, inicialmente utilizado para jogos de tiro em primeira pessoa mas, posteriormente, passou a ser utilizado para desenvolver jogos de diferentes gêneros e atualmente é um dos motores gráficos mais famosos. Assim como o Unity é multiplataforma.
- RPG Maker<sup>5</sup> apesar não ser tão poderoso como as plataformas citadas anteriormente se tornou famosa por possibilitar a criação de jogos sem a necessidade do conhecimento prévio de programação, além deste motivo possui uma comunidade ativa que

---

<sup>3</sup><https://unity3d.com/pt>

<sup>4</sup><https://www.unrealengine.com/en-US/what-is-unreal-engine-4>

<sup>5</sup><https://www.rpgmakerweb.com/>

frequentemente disponibiliza recursos compatíveis com a plataforma e compartilha conhecimentos para auxiliar o desenvolvimento de jogos pela comunidade.

### 3 MATERIAS E MÉTODOS

Neste capítulo serão citados brevemente os materiais e métodos que serão utilizados no desenvolvimento do projeto em questão.

#### 3.1 MATERIAIS

Nesta seção serão apresentados os materiais utilizados no desenvolvimento da plataforma assim como a justificativa para a sua utilização.

##### 3.1.1 Unity Engine

Unity é uma plataforma de motor gráfico amplamente utilizada no mundo inteiro com uma distribuição gratuita. Com essa plataforma é possível desenvolver jogos e animações de alta qualidade a partir de uma série de recursos disponíveis que contam com uma comunidade ativa e uma variedade de conteúdo explicativo que auxiliam na aprendizagem da utilização da ferramenta.

Suas funções têm como base a interação de entidades denominadas "GameObjects" que podem representar qualquer tipo de recurso utilizado no jogo. Nesse contexto pode-se criar scripts na linguagem C# e estabelecer vínculos para descrever o comportamento dos GameObjects durante a execução do programa. Além disso é possível adquirir recursos adicionais pré-fabricados na "AssetStore", o ambiente oficial da Unity para obtenção e disponibilização de recursos que podem ser integrados à plataforma, onde pode-se encontrar uma variedade de recursos para as mais diversas finalidades.

### 3.1.2 Visual Studio Code

O Visual Studio Code é um editor de código-fonte desenvolvido pela Microsoft. O editor conta com uma interface simples com suporte para diversas linguagens de programação e a possibilidade da instalação de plugins que possibilitam a integração de diversas ferramentas, como, por exemplo, o Unity Engine.

### 3.1.3 Windows

O sistema operacional utilizado com ambiente de desenvolvimento será o Windows devido à sua abrangência de recursos disponíveis e compatibilidade com uma variedade de dispositivos e aplicativos. A escolha também deve-se a familiaridade com o sistema operacional.

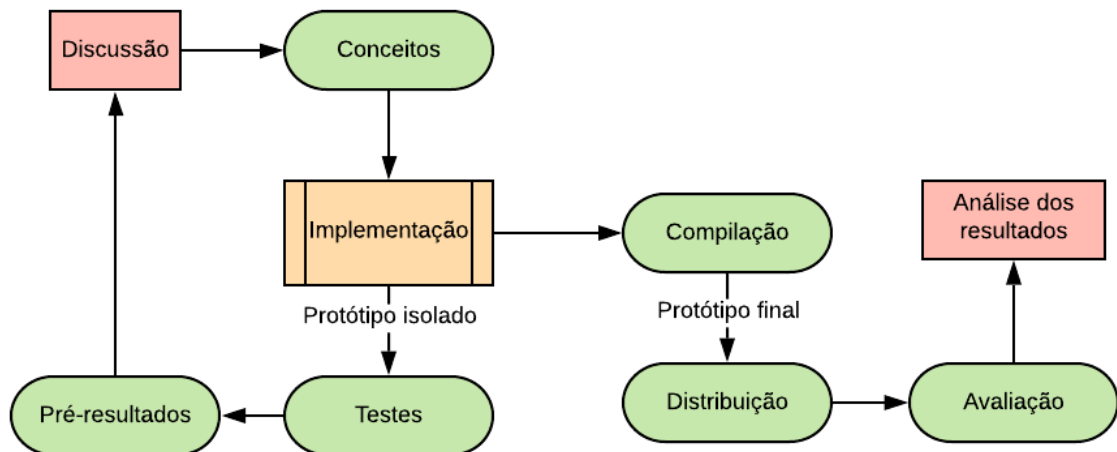
### 3.1.4 Google Forms

O Google Forms é uma ferramenta para criação de formulários em nuvem gratuita e necessita de um endereço de email da Google para a criação e edição. Alguns dos recursos disponíveis da ferramenta são formulários personalizáveis que podem ser distribuídos através de um link de visualização e contam com uma ferramenta de análise através de gráficos gerados automaticamente para o controle das distribuições do formulário.

## 3.2 MÉTODOS

Neste capítulo serão apresentados os materiais e o método utilizado no desenvolvimento deste trabalho. A Figura 1 mostra o fluxo de processos pertinentes para o desenvolvimento do serious game.





**Figura 1 – Diagrama de Metodologia de Desenvolvimento**

**Fonte: Autoria Própria**

### 3.2.1 Planejamento

O conteúdo do jogo é organizado de acordo com o MDA Framework, citado anteriormente para facilitar a organização e entendimento da plataforma como um todo.

O método utilizado para definir os conceitos iniciais do jogo foi o *Brainstorm*, onde foram apresentadas ideias e informações para dar início ao processo de estruturação do jogo, onde as possibilidades foram armazenadas para uma análise de viabilidade da implementação das ideias apresentadas em um protótipo isolado.

Com o protótipo em funcionamento foram realizados testes para avaliar a integridade da implementação e a facilidade de entendimento dos processos necessários para o avanço no jogo, com isso foi possível identificar as melhorias a serem realizadas e discutir os resultados com o objetivo de construir o protótipo final.

Com os protótipos isolados validados foi definido o escopo do protótipo final e realizada a integração dos mesmos para a compilação e distribuição do projeto para a avaliação pelo público. Esse processo está ilustrado pela Figura 1.

Os conteúdos apresentados no serious game seguem uma ordem lógica de aprendizado que foi baseada na ementa da disciplina de fundamentos de programação. Os tópicos explorados são, respectivamente, armazenamento e atribuição de valor, variáveis, tipos de dados, condicionais e laço de repetição.

### 3.2.2 Desenvolvimento

No desenvolvimento foram implementados protótipos isolados onde algumas ideias foram testadas e avaliadas pelos desenvolvedores e mantidos caso o resultado antigisse o objetivo esperado. Algumas ideias contudo foram descartadas por apresentarem resultados inesperados, insatisfatórios ou confusos.

Nas análises dos protótipos foram avaliados os aspectos intuitivos da jogabilidade, tanto como a estética apresentada e a dificuldade de avanço e assimilação dos conceitos. A cada teste e análise os recursos e implementações foram aprimorados visando a independência do jogador, de forma que ele possa progredir sem uma influência externa, os testes preliminares foram realizados sem que houvessem explicações de como o jogo deveria funcionar, passando apenas as informações dos comandos possíveis e analisando as dificuldades que os jogadores enfrentavam ao tentar avançar durante os níveis presentes. O modo como as informações eram apresentadas foram mantidas ou descartadas de acordo com a receptividade dos jogadores. Os testes preliminares foram realizados com diversos jogadores universitários na faixa de 18 a 25 anos, baseando-se na ementa apresentada nos cursos presente no câmpus que variam entre ciência da computação e as engenharias ambientais, de alimentos, elétrica e de produção, assume-se que todos tiveram algum contato com lógica de computação ou até dominam esses conceitos com excelência.

### 3.2.3 Avaliação

Após as implementações necessárias contemplando todo o conteúdo educativo previsto foi gerada uma versão alpha do projeto que foi utilizada para avaliação final. A avaliação foi realizada via Internet através da ferramenta Google Forms de forma anônima onde os voluntários realizaram uma inscrição informando o nome completo, idade e email. Para todos os incritos foi enviado uma cópia da versão alpha junto com um link para o formulário de avaliação contendo perguntas referentes aos aspectos de entretenimento e educacionais do serious game, no qual os avaliadores poderiam dar uma nota linear escalar numa faixa entre 1 e 5, simbolizando péssimo para a menor nota e incrível para a maior nota, respectivamente. As avaliações foram realizadas de forma voluntária e anônima para que a identidade do avaliador não estabeleça nenhuma influência sobre as notas do serious game.

## 4 RESULTADOS

Nesse capítulo serão detalhados os resultados do projeto do serious game detalhando a lógica de implementação.

### 4.1 MECÂNICAS

Nessa seção serão apresentadas as mecânicas presentes no serious game e uma breve explicação de como elas funcionam. Neste trabalho foi utilizado o modelo tridimensional presente na ferramenta Unity, que é baseado em GameObjects, onde cada GameObject possui um componente denominado Transform que possui diversos componentes que podem ser usados para controlar um GameObject.

As propriedades do componente Transform mais pertinentes para este trabalho são as denominadas Position e Rotation. Elas são usadas para armazenar informações da posição e rotação do objeto dentro do ambiente virtual tridimensional e serão explicados com mais detalhes posteriormente.

#### 4.1.1 Movimentação

A movimentação ocorre através da aplicação de conceitos geométricos aplicados ao GameObject utilizando os componentes do Transform, Position e Rotation, e também com a utilização do componente Rigidbody, que tem como objetivo aplicar conceitos físicos como a força gravitacional e atrito.

O jogador pode movimentar o personagem em quatro direções: norte, sul, leste e oeste.

Para realizar a movimentação ele pode utilizar as teclas W, A, S e D, respectivamente, ou pode utilizar as setas direcionais do teclado para mover-se para a direção equivalente a seta pressionada.

Para que a movimentação ocorra na direção desejada, foi necessário atribuir um vetor de três dimensões ao componente EulerAngels que é usado para definir a direção do personagem junto com o componente Rotation, que possui três eixos que podem ser rotacionados, onde somente o parâmetro que representa o ângulo no eixo Y foi alterado, o resultado da variação do ângulo neste eixo faz com que o personagem gire para a direita ou para a esquerda dependendo do valor do ângulo. Por padrão o ângulo zero aponta para a direção norte no plano em que o personagem está e então foram adicionados noventa graus ao ângulo padrão para definir as demais direções desejadas, leste, sul e oeste, que resultaram nos ângulos 90, 180 e 270, respectivamente.

O componente Transform possui um método denominado Translate que é utilizado para alterar as coordenadas de posição do personagem e é armazenada no componente Position, onde, após a direção ser definida, é realizado uma translação linear no eixo Z de um valor pré-definido que resulta no incremento da posição atual na direção definida pelo componente Rotation.

Para que a movimentação ocorra de forma natural foi necessário ajustar as propriedades do componente Rigidbody, para que os objetos tenham gravidade e atrito aplicados em sua mecânica. Para que o personagem conseguisse empurrar um cubo sem esforço aparente foi necessário que a propriedade que representa a massa corporal do personagem fosse dez vezes maior do que a massa corporal definida para o cubo.

#### 4.1.2 Interação

O jogador pode adquirir as informações necessárias interagindo com os objetos de interesse, isso ocorre quando a colisão entre o personagem e um objeto resulta no surgimento de uma figura vermelha com a forma de um ponto de exclamação, a interação ocorre quando o jogador aperta a tecla “Espaço” ou a tecla “0” e o resultado de uma interação pode variar de acordo com o objeto. Um aspecto comum entre todos os objetos é o surgimento de uma “Caixa de Diálogo”, que consiste em um painel contendo um texto explicativo sobre como o objeto funciona ou um pseudo-código com uma expressão equivalente ao funcionamento do objeto de interesse, que ocorre como resultado de uma interação com o personagem.

Para que o jogador consiga entender como funcionam os objetos encontrados ele precisa interagir com cada um individualmente e através da informação contida na caixa de diálogo deduzir a forma necessária de manipular os objetos para progredir para o próximo nível.

#### 4.1.3 Batalha

Uma batalha ocorre sempre que o personagem toca em um inimigo. Todos os inimigos tem um raio de alcance de detecção. Quando o personagem entra no raio de alcance, o inimigo começa a perseguí-lo e não irá parar até que encoste no personagem ou caso o personagem consiga sair do raio de alcance.

A batalha é dividida em dois turnos: o turno do inimigo e turno do personagem. O primeiro turno é sempre do inimigo, onde ele avança até o personagem e realiza um ataque. Pouco antes do ataque ser realizado surge um indicador de interação acima da cabeça do personagem por uma curta duração de tempo. Caso o personagem aperte o botão de interação dentro desse período de tempo ele reduzirá a quantidade de dano causado pelo inimigo pela metade, mas caso ele aperte antes do surgimento do indicador ou depois que o indicador desaparecer ele receberá o dano total proporcionado pelo inimigo.

Na sequência o turno do personagem é iniciado, onde ele avança até o inimigo e realiza um ataque. Pouco antes do ataque ser realizado também surge um indicador de interação acima do personagem e se o botão de interação for apertado enquanto o indicador estiver visível, o personagem proporcionará o dobro de dano ao inimigo, que causará a morte do inimigo, caso o jogador falhe em apertar o botão enquanto o indicador estiver visível, ele causará o dano padrão e terá que esperar até seu próximo turno para realizar um novo ataque.

A batalha ocorre em um contexto diferente do principal, o que permite a modularização de funcionalidades de batalha sem que essas modificações afetem as mecânicas no contexto principal.

## 4.2 DINÂMICAS

Aqui serão apresentadas as dinâmicas envolvidas no desenvolvimento do serious game.

#### 4.2.1 Controles de Dados

A principal forma de interação entre os objetos são feitas através de colisões, onde todos os objetos possuem um componente chamado Collider, que calcula se o objeto atual realizou uma colisão com um objeto externo a sua estrutura. O controle dos resultados das colisões é feito através de etiquetas que representam o tipo de um objeto, a partir desse conceito é possível identificar quando foi o personagem ou uma variável que realizou uma colisão com um objeto de interesse. O objeto que representa o jogador possui a etiqueta “Player”, os objetos que representam as variáveis possuem a etiqueta “Var”, os objetos em que o jogador pode interagir, como leitores e operadores, possuem a etiqueta “Item”.

Além do controle de colisões é necessário armazenar o progresso do jogador, isso ocorre através do componente PlayerPrefs, componente do Unity Engine que instancia arquivos binários que armazenam registros em persistência. O armazenamento ocorre sempre que o jogador finaliza um nível o número equivalente ao nível alcançado é armazenado pelo PlayerPrefs, permitindo que o jogador continue do mesmo nível que alcançou caso ele decida continuar o jogo em outro momento. Com um nível armazenado anteriormente é possível acessar-lo ao executar o aplicativo novamente e ao clicar no botão “Continuar”, presente no menu inicial, o jogador será colocado na posição inicial do nível que alcançou.

#### 4.2.2 Dicas - Fonte de Conhecimento

O primeiro objeto visível, além do chão e das paredes que cercam o ambiente em que o personagem está, é uma esfera de luz que representa uma “Dica”, que ao colidir com o personagem surge uma “Caixa de Diálogo”, contendo informações cruciais para o avanço do jogador para os próximos níveis. Em seguida a “Dica” é destruída e a caixa de diálogo permanece até que o personagem se mova. A Figura 2 apresenta a caixa de diálogo informando os comandos básicos do jogo.

No contexto do serious game as dicas são referidas como “Fontes de Conhecimento”, elas contêm todo o conteúdo explicativo necessário para entender os conceitos apresentados e algumas informações que podem ser úteis para a progressão dos níveis.



**Figura 2 – Caixa de diálogo contendo os comandos básicos**

**Fonte: Autoria Própria**

#### 4.2.3 Variáveis

O conceito de variável é representado através de um “Cubo”. O cubo pode ser empurrado pelo jogador para ser posicionado até colidir com um objeto de interesse, que pode ser uma condicional, um operador ou um laço de repetição, o resultado de uma colisão varia de acordo com o tipo de objeto (Figura 3).

Caso a colisão ocorra com uma estrutura condicional ou laço de repetição, ocorre uma verificação para saber se o valor satisfaz a condicional ou a condição de parada do laço de repetição. Para realizar o controle das variáveis foi adotado um padrão nos nomes dos objetos que facilitam a manipulação dos tipos e valores atribuídos. De acordo com o padrão estabelecido as três primeiras letras representam o tipo da variável que podem ser: VAR, STR, NUM e BOL, onde elas representam os tipos: genérico, string, inteiro e booleano, respectivamente. A quarta letra do nome é um caractere de controle que é usado para identificar se a variável possui uma instância no mapa, que é representada por ‘:’, ou se ela é uma variável declarada pelo jogador, que é representada por ‘\_’. A palavra que é composta pelas letras que vem a seguir da quarta letra representam o valor da variável, dependendo do tipo da variável o seu valor admite um vazio, por exemplo, uma string pode ser vazia.



**Figura 3 – Caixa de diálogo citando o termo Variável**

**Fonte: Autoria Própria**

#### 4.2.4 Estruturas Condicionais

As portas funcionam como o resultado de uma condicional e as condicionais são representadas por um objeto semelhante a um poste de luz, que é ativado toda vez que colide com um cubo. Cada porta é acompanhada por uma condicional e uma placa que pode ser realizada uma interação para que a condição seja informada em uma caixa de diálogo, caso o valor do cubo satisfaça a condição escrita na placa a porta é aberta. O progresso do jogo ocorre quando o jogador abre uma porta, mas nem todas as portas são necessárias para avançar de nível, algumas portas abrem caminhos que possuem diferentes objetos, que podem ser recursos ou até inimigos, esses caminhos podem resultar em um beco sem saída.

A dinâmica das condicionais é uma das mais importantes do jogo, ela é essencial para validar o conhecimento do jogador sobre os termos e conceitos apresentados. O controle das condicionais ocorre de acordo com sua nomenclatura, de forma similar ao das variáveis. Os três primeiros caracteres são usados para identificar se é uma estrutura condicional simples, que contém somente uma porta, ou composta, que contém um poste interligado por duas portas. Quando o nome do objeto começa com a sequência “iff” ou “ifi” significa que a condicional é simples, caso a sequência inicial seja “ife” significa que a condicional é composta.



Quando a luz que se acende em resultado da colisão com um cubo tiver uma cor verde significa que a condicional foi satisfeita e a porta se abrirá, mas quando a luz tem uma cor vermelha significa que o valor do cubo não satisfaz a condicional e, caso a condicional seja simples, a porta apenas não se abrirá e causará dano ao jogador mas caso seja uma condicional composta uma porta continuará fechada e outra porta irá se abrir.

A Figura 4 apresenta um exemplo de interação com uma estrutura condicional simples. A condicional na figura exige que a variável seja do tipo booleano.



**Figura 4 – Interação com estrutura de condicional simples**

**Fonte: Autoria Própria**

Pode-se deduzir pela Figura 4 que o nome do objeto que representa a condicional é “iff:bol=TRUE”, portanto, o resultado da colisão entre o mesmo e um cubo será positivo e a porta se abrirá caso o objeto do cubo tenha o nome “bol:TRUE”, o que indica que o cubo é do tipo booleano e seu valor é TRUE, mas o resultado será negativo caso o cubo tenha um nome diferente, independente do tipo ou valor. Do quinto ao sétimo caractere presente no nome do objeto contém a sequência que identifica o tipo de variável que ativa a condicional, o caractere seguinte identifica o operador lógico da expressão condicional e os caracteres que estão na sequência são usados para identificar a referencia de valor presente na condicional. Uma condicional é ativada sempre que colide com um objeto que contenha a etiqueta “Var”.

O nome do objeto define o funcionamento da condicional, mas a mensagem que aparece para o jogador em forma de caixa de diálogo depende de outros componentes e, por

isso, pode ser usada de maneira independente da sintaxe de controle da condicional. A Figura 5, exemplifica uma forma de uso baseada em uma mensagem descritiva.



**Figura 5 – Interação com estrutura condicional de mensagem descritiva**

**Fonte: Autoria Própria**

Esse recurso demonstrado na Figura 5 permite que a validação dos conceitos informados sejam aplicados para diversos conteúdos, onde a caixa de diálogo pode apresentar, por exemplo, uma pergunta descritiva sobre o conteúdo ou uma expressão lógica ou matemática.

#### 4.2.5 Operadores

A dinâmica do operador tem como função alterar o valor de um cubo, isso ocorre sempre que um cubo, ou com um objeto que tenha a etiqueta “Var”, colida com um operador. O controle do operador também utiliza o padrão de nomenclatura de seu objeto, mas no caso do operador o terceiro caractere representa a operação que será realizada, que é representada pelo símbolo equivalente ao operador lógico, assim como a sequência de caracteres posteriores representam o valor que será utilizado na operação. As operações contempladas pelos operadores consistem nas operações básicas matemáticas, como adição e subtração, e na

operação de atribuição de valor. A Figura 6 mostra um cubo colidindo com um operador, isso resulta na alteração do valor do cubo de acordo com a expressão descrita no operador.



**Figura 6 – Operador Ativado**

**Fonte: Autoria Própria**

O comportamento de cada operação depende do tipo de dado da variável, por exemplo, a operação adição se comporta de uma forma para uma variável do tipo numeral mas é um comportamento diferente em uma variável do tipo string. Esses comportamentos são exemplificados na próxima subseção.

#### 4.2.6 Tipos de Dados

Os tipos de dados definem o comportamento dos valores das variáveis, atualmente existem quatro tipos possíveis.

- VAR: É definido como um tipo genérico de variável, que se comporta como um número inteiro caso realize somente operações com inteiros. Esse tipo só está presente na fase inicial do jogo antes do conceito de tipos de dados for apresentado. As operações que o tipo VAR realiza são as operações matemáticas como de um numeral mas, diferente de

- uma variável do tipo NUM, ela pode ter uma palavra ou caractere atribuído ao seu valor;
- STR: É um tipo que assume o comportamento de uma String. Ela pode realizar uma operação de atribuição para qualquer valor incluindo um valor vazio, além de realizar a operação de concatenação quando o operador apresenta o símbolo de adição. Todas as demais operações além da atribuição e adição resultam no valor “erro”;
  - NUM: É o tipo que assume o comportamento de um número inteiro, realizando as operações matemáticas de atribuição, adição, subtração, multiplicação e divisão. Caso o valor da operação não seja um número inteiro o valor resulta em “erro”;
  - BOL: Assume somente os valores TRUE e FALSE. Todas as demais operações além da atribuição resultam no valor “erro”.

#### 4.2.7 Leitores

Os leitores são objetos utilizados para informar o jogador sobre o valor de um “cubo”. Ele é ativado sempre que acontece uma colisão com um objeto que contenha a etiqueta “Var” e resulta na apresentação de uma caixa de texto informando o valor atual da variável. A Figura 7 mostra o resultado da colisão de um cubo com um leitor.

O leitor tem uma luz que acende quando ocorre uma colisão com um cubo, semelhante a forma como ocorre com o operador. Assim como podemos ver na Figura 7 sempre que ocorre uma colisão com um cubo é apresentada uma caixa de diálogo para o jogador informando o valor atual do cubo.

Os leitores funcionam independente do tipo do cubo e sempre mostrará todo e qualquer valor que esteja na sequência de caracteres a partir da terceira posição da string que compõe o nome do objeto do cubo.

#### 4.2.8 Declaradores

Um declarador é representado por um objeto que cria um novo cubo quando é realizada uma interação com o mesmo pelo jogador. Essa funcionalidade permite que o jogador crie a quantidade de cubos que desejar porém todos os cubos serão criados por padrão com o valor



**Figura 7 – Leitor Ativado**

**Fonte: Autoria Própria**

“NULL”. Cada declarador pode ser usado para declarar somente um único tipo de variável, que é descrito em uma caixa de diálogo sempre que uma variável é declarada. A Figura 8 mostra a criação de um cubo e uma mensagem descrevendo a declaração.

Esse tipo pode ser identificado pela sequência de caracteres que são posteriores ao quarto caractere do nome do objeto do declarador. Quando o personagem colide com o botão verde ele tem a possibilidade de realizar uma interação, que resulta como a criação de um novo cubo que surge na área demarcada posicionada abaixo do botão verde.

#### 4.2.9 Inimigos

Um inimigo é um objeto que permanece em um estado de espera até que seja detectado um personagem, objeto com a etiqueta “Player”, em um raio de distância definido. Quando esse objeto é detectado, o inimigo rotaciona o vetor que representa a frente do objeto até a posição do personagem e movimenta-se para a frente até que o personagem consiga sair do raio de detecção ou haja uma colisão entre o inimigo e o personagem. O inimigo pode ser visto ao lado



**Figura 8 – Jogador interagindo com o Declarador**

**Fonte: Autorial Própria**

direito da Figura 9, preparando-se para se movimentar em direção ao jogador.



**Figura 9 – Inimigo em estado de espera**

**Fonte: Autorial Própria**

Quando ocorre uma colisão com o personagem a cena de batalha é carregada, se o personagem vencer o objeto do inimigo é destruído, mas caso o personagem seja derrotado o

jogo se encerra e a cena de encerramento do jogo é carregada.

#### 4.2.10 Laços de Repetição (Loops)

Os loops são os objetos equivalentes a um laço de repetição, eles possuem uma estrutura composta por uma área interna demarcada em vermelho, uma estrutura condicional simples cercada por uma estrutura de limitação em uma área quadrada e pelo menos uma área destinada a alocação de um operador móvel como pode-se ver na Figura 10 .



**Figura 10 – Interação com uma estrutura de loop**

**Fonte: Autoria Própria**

Na Figura 10 nota-se que há um seta no sentido anti-horário dentro da área vermelha. A seta indica que se um cubo que for colocado na área vermelha ficará realizando movimentos giratórios no sentido indicado.

Os Loops apresentam uma estrutura condicional interna e que aplica os mesmos conceitos de implementação das estruturas condicionais citadas anteriormente. Entretanto, os loops possuem um mecanismo que faz com que a variável se movimente em círculos e colida com a estrutura condicional e com qualquer operador móvel alocado nas estruturas periféricas



do loop, nota-se que na Figura 10 há somente um operador alocado na extremidade sul da estrutura do loop. Esse mecanismo faz com que um cubo que for inserido na área vermelha do loop realize diversas interações com um ou mais operadores e com a estrutura condicional, o mecanismo só é interrompido quando o valor do cubo não satisfaz a estrutura condicional alocada no loop.



**Figura 11 – Robô emitindo luzes azuis e operador de comando no loop**

**Fonte: Autoria Própria**

Os loops podem ser usados para enviar ordens de comando para um Robô, que pode ser observado na Figura 11. Os robôs são objetos que interagem somente com um tipo específico de operador que é usado para movimentar um único robô vinculado a ele. Isso ocorre quando uma colisão entre o operador de comando e um cubo é realizada e resulta na movimentação do robô em uma curta distância na direção determinada pelo operador, somente uma das quatro direções é possível por operação: Norte, Sul, Leste ou Oeste. Cada operador tem como valor a direção na qual o robô será movimentado e o jogador será informado dessa direção caso realize uma interação com o operador de comando.



## 4.3 ESTÉTICA

Nessa seção serão apresentados conceitos estéticos que compõe o serious game.

### 4.3.1 Iluminação

As luzes são utilizadas como uma tentativa de controlar os pontos de atenção do jogador. Por esse motivo o ambiente é predominantemente escuro. Os objetos de interesse emitem uma luz para guiar o jogador até a informação necessária, que pode auxiliar o jogador a avançar durante os níveis ou a entender alguns conceitos.

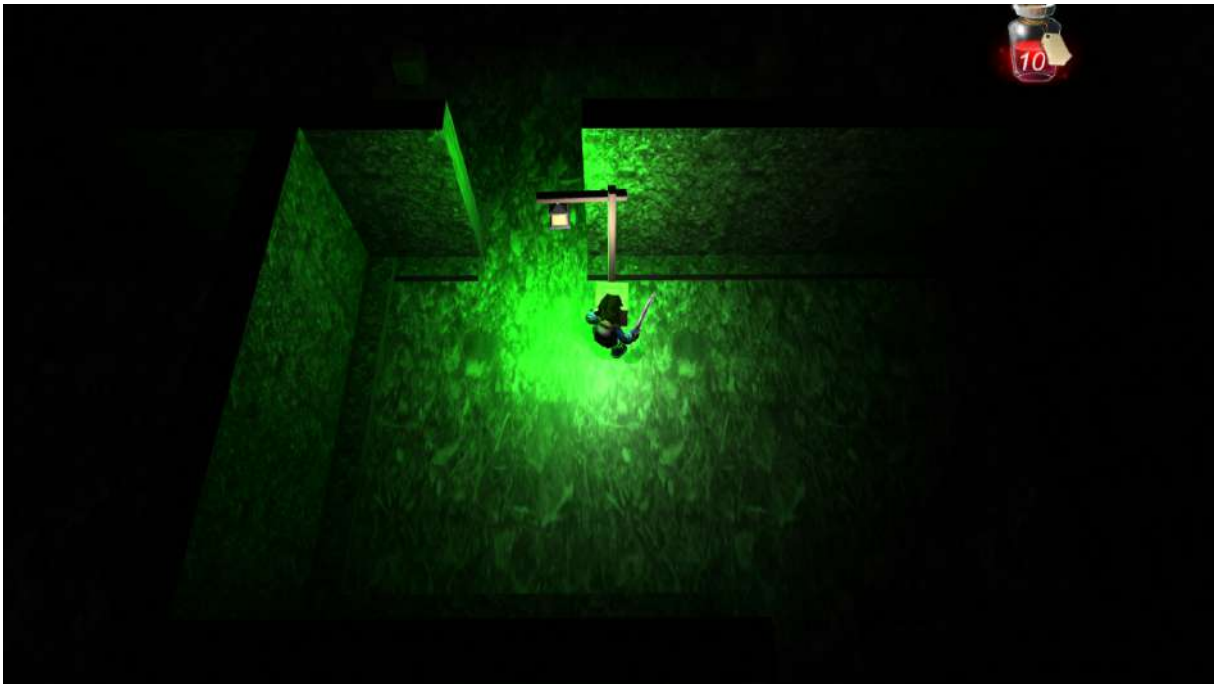
As luzes também são usadas para apresentar um retorno visual para que o jogador saiba o resultado de uma interação. Na maioria das vezes esse retorno pode ser identificado através das cores, sendo verde para positivo e vermelho para negativo.

Com a imagem apresentada na Figura 12 é possível deduzir que o cubo que está tocando a estrutura condicional possui um valor que satisfaz a condição abrindo caminho para o jogador.

### 4.3.2 Batalha

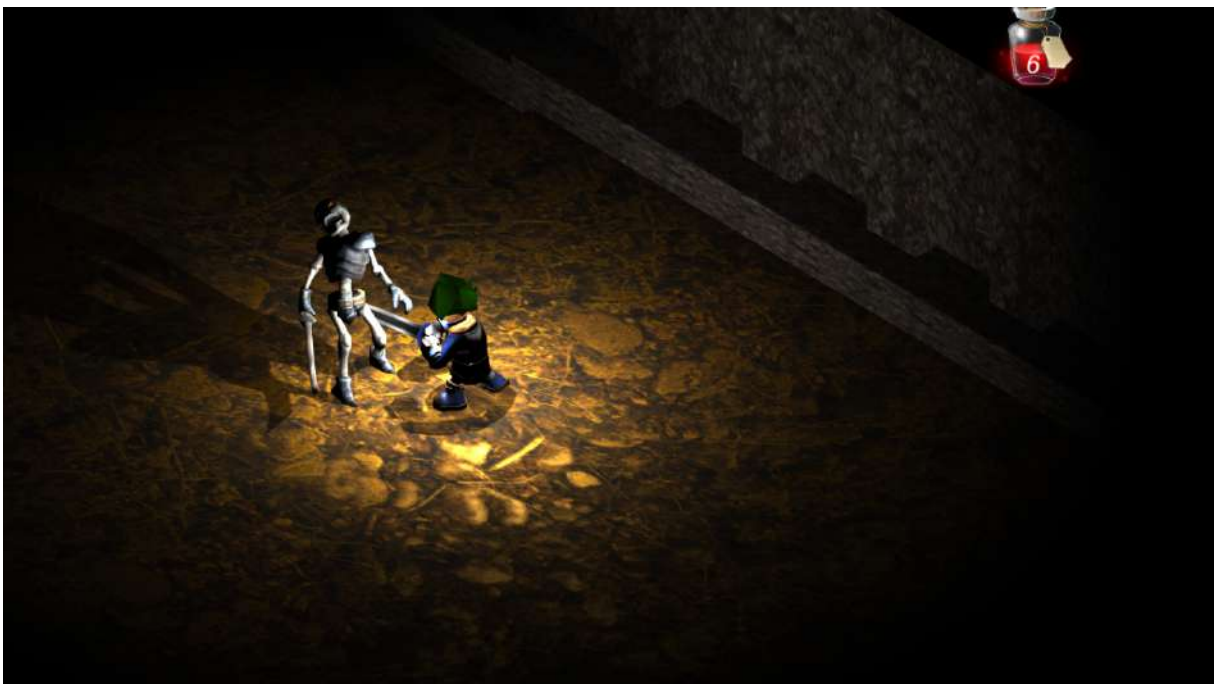
A estética da batalha é muito comum nos jogos do gênero de RPG clássicos. O conceito de mudar-se o foco de maneira repentina e apresentar somente o jogador e o inimigo como foco da cena é amplamente utilizado por permitir uma estética mais agradável que permite que o personagem e o inimigo sejam visualizados com maiores detalhes. A Figura 13 ilustra um momento da batalha quando o personagem realiza um ataque ao inimigo.

Pode-se perceber que a batalha segue uma estética única onde não há presença de objetos interativos, apenas o personagem e o inimigo estão presentes. Nesta cena o foco de atenção deve ser totalmente voltado aos momentos de interação que são identificados pelo surgimento do ponto de exclamação sobre a cabeça do personagem.



**Figura 12 – Estrutura de condicional simples satisfeita**

**Fonte: Autoria Própria**



**Figura 13 – Personagem principal atacando o inimigo**

**Fonte: Autoria Própria**

### 4.3.3 História

A história se passa em um mundo onde a tecnologia foi esquecida. Devido a falta de consciência das pessoas para a utilização dos recursos naturais, ocorreu um desastre ambiental que acabou com toda a forma de tecnologia baseada na eletricidade.

Os recursos tecnológicos passaram a emitir uma radiação intensa que causou a morte de muitas pessoas e acabou com uma parte significativa da humanidade. Somente os que passaram a viver longe da tecnologia sobreviveram. Nesse contexto países com uma paisagem natural preservada, como o Brasil, prosperaram e lentamente foram se tornando as novas potências mundiais.

Essas potências mundiais estabeleceram o monopólio sobre o conhecimento tecnológico que restava na humanidade e se tornaram uma soberania mundial.

Depois de séculos foram descobertas algumas formas de tecnologias no subterrâneo que não foram alteradas pela radiação, por esse motivo iniciaram-se campanhas de mineração para buscar as tecnologias do passado e descobrir seu segredo. Muitos grupos de busca foram bem sucedidos em encontrar artefatos do passado na antiga região de Minas Gerais, e muitos caçadores de recompensa foram encorajados para buscar novos artefatos por essa área, mas muitos não voltaram dessa jornada.

O nosso herói parte para descobrir o que está acontecendo na região de Minas Gerais. Acredita-se que existem mortos vivos atacando todos que buscam o conhecimento das tecnologias perdidas. A missão do herói é descobrir o segredo dos mortos, enquanto encontra tecnologias nunca vistas por ele no caminho.

A contextualização da história foi apresentada por meio de textos explicativos que se iniciam na cena de introdução contando a situação atual do personagem e sua motivação para ingressar na jornada. Para apresentar os textos foram utilizados componentes de interface gráfica do Unity. O principal componente é chamado de Canvas que é a base para toda a interface. Dentro do Canvas pode ser adicionado um componente Panel que possui objetos de texto que podem ser adicionados como filhos. Os objetos de texto que compõe a cena de introdução e encerramento são controlados através de um *Script* que segue uma ordem para apresentar os textos tornando-os visíveis por um breve momento até que o próximo objeto de texto possa ser apresentado na sequência.

#### 4.3.4 Personagens e Assets

O personagem principal é inspirado em ícones da cultura pop japonesa. As características presentes como a aparência inofensiva infantil e os cabelos pontiagudos e coloridos são predominantes nessa cultura, que serviu como influência para muitos jogos do gênero. O personagem principal pode ser observado na Figura 14.



**Figura 14 – Modelo do personagem principal.**

**Fonte: Autoria Própria**

Para permitir essa caracterização foi necessário alterar o modelo original adquirido do personagem principal. A edição foi realizada utilizando um asset chamado “MeshEditor” que permite a edição de modelos tridimensional a partir do controle da posição das arestas dos polígonos que compõe a estrutura do modelo tridimensional.

A estética do personagem principal foi desenvolvida inspirada nos perfis de personagens que representam os heróis em suas obras, que é comum em jogos RPG clássicos e desenhos japoneses. O objetivo é apresentar uma estética na qual um aluno consiga identificar facilmente quem representa o herói e quem representa o vilão da história.

Os inimigos apresentam um estética predominante na fantasia de seres malignos o que pode sugerir ao jogador o pensamento de perigo ao encontrá-lo. Em alguns testes notou-se que alguns jogadores evitaram o contato a qualquer custo dos inimigos por acreditar que a colisão com um inimigo causaria uma derrota imediata. O inimigo pode ser visualizado na Figura 15.



**Figura 15 – Modelo dos inimigos**

**Fonte: Autoria Própria**

Todos os Assets utilizados neste trabalho estão disponíveis na AssetStore de forma gratuita, são de livre distribuição e possuem uma descrição de como devem ser utilizados, além de exemplos práticos de implementação. Os Assets podem ser encontrados na AssetStore pelos nomes listados a seguir:

- RPG Hero PBR HP Polyart;
- Fantasy Monster - Skeleton;
- Medieval Town Exteriors;
- Action RPG Music Pack v1.2;
- Blade Sounds;
- Electric Sfx;
- Simple Lightning Bolt;
- MeshEditor;
- RPG Pack Free;
- RPG Album;
- RPG Inventory Icons;
- Starfield Skybox;
- VRInteractions.

#### 4.4 ESTRUTURA DE NÍVEIS

O serious game é estruturado em cinco níveis, cada nível tem como objetivo apresentar um conceito da lógica computacional de forma prática, sendo que o primeiro nível foi construído com o objetivo de ser uma espécie de tutorial, onde o jogador recebe instruções sobre como o jogo funciona e como é possível avançar durante os níveis.

Ao início do primeiro nível o jogador se encontra em uma sala contendo somente uma porta com um poste ao lado, que é usado para representar uma condicional, e um cubo, que é usado para representar uma variável. O primeiro desafio é empurrar o cubo de maneira que o mesmo encoste no poste que fica posicionado próximo a porta, a Figura 16 mostra a instrução para que o jogador saiba qual é o objetivo.



**Figura 16 – Primeiro desafio do jogador**

**Fonte: Autoria Própria**

Ao avançar o jogador encontra uma sala contendo uma porta e dois cubos, nesse cenário somente um dos cubos pode abrir a porta, caso o jogador coloque o cubo errado no poste ele perderá um ponto de vida e receberá um retorno visual e auditivo e então terá que remover esse cubo e colocar o outro no poste.

Avançando para a próxima sala o jogador encontra duas portas, cada uma com um poste e uma placa indicando o valor que abre a porta, mas somente um cubo na sala, essa sala

também tem o objetivo de introduzir um Leitor que pode ser usado para que o jogador saiba qual o valor presente no cubo e, conseqüentemente, qual a porta que ele pode abrir.

Na seqüência existe uma sala com uma porta e um cubo contendo um valor que não abre a porta, nessa sala é introduzido o operador, que, ao ser encostado pelo cubo, altera o valor do cubo para que o mesmo possa abrir a porta e avançar.

As demais salas no primeiro nível irão explorar diferentes usos dos operadores de forma que o jogador precise utilizá-los para alterar o valor dentro de um cubo com o objetivo de abrir as portas que estiverem pelo seu caminho enquanto se depara com inimigos que estarão prontos para iniciar uma batalha caso detectem e encontrem no personagem do jogador, esse padrão de cenário se repete até o final de todos os níveis.

O segundo nível possui instruções que esclarecem que o cubo se comporta como uma variável, nesse nível também é introduzido um declarador, que é necessário para criar novos cubos para abrir portas em salas que não possuem cubos presentes.

O terceiro nível apresenta Declaradores e Operadores que contemplam os demais tipos de dados. Os primeiros desafios envolvem portas que se abrem com valores de *String*, como , por exemplo, "aba", e ao final do nível é necessário utilizar um cubo que seja do tipo Lógico para avançar.

O quarto nível apresenta o conceito de estrutura condicional usando as portas como referência para o entendimento desse conceito. Nos níveis anteriores as placas apresentavam um texto descritivo para que o jogador pudesse entender que ela só seria aberta com um determinado valor, mas nesse nível é apresentado um pseudo-código que representa uma estrutura condicional e ao longo das salas é possível encontrar instruções que expliquem o significado dos pseudo-códigos e como eles funcionam.

O quinto e último nível apresenta a estrutura de laço de repetição que deverá ser usada de duas formas, para alterar o valor de um cubo até que o mesmo satisfaça a estrutura condicional em uma porta, e para movimentar um robô até que ele empurre um cubo com o objetivo de abrir uma porta onde o jogar não teria acesso.

#### 4.5 AVALIAÇÃO

A avaliação foi dividida em três etapas: inscrição por meio de formulário eletrônico, teste do jogo e validação por meio de formulário eletrônico, no total foram realizadas vinte e

três avaliações por um público misto.

Para a primeira etapa da avaliação os voluntários realizaram uma inscrição através do preenchimento de um formulário com as informações de nome completo, idade e email, além de um termo de responsabilidade. No total foram realizadas 80 inscrições para avaliar o protótipo. Após a primeira etapa concluída foi enviado um email contendo um *link* para *download* do protótipo e também um *link* para o formulário contendo as perguntas de avaliação.

Os inscritos tiveram o período de uma semana para testar o protótipo e realizar a avaliação sobre o mesmo. No total foram recebidas 23 avaliações, assume-se que os inscritos remanescentes não conseguiram finalizar o protótipo e concluir o teste a tempo e, por consequência, não entregaram a avaliação preenchida.

Todas as respostas referentes a avaliação são medidas por uma escala linear (escala de Likert) de 1 a 5, onde os números representam as notas: péssimo, ruim, indiferente, bom e incrível, respectivamente. Estas notas podem ser visualizadas no eixo horizontal dos gráficos apresentados nessa seção enquanto que o eixo vertical representa a porcentagem de votos atribuídos pelos avaliadores.

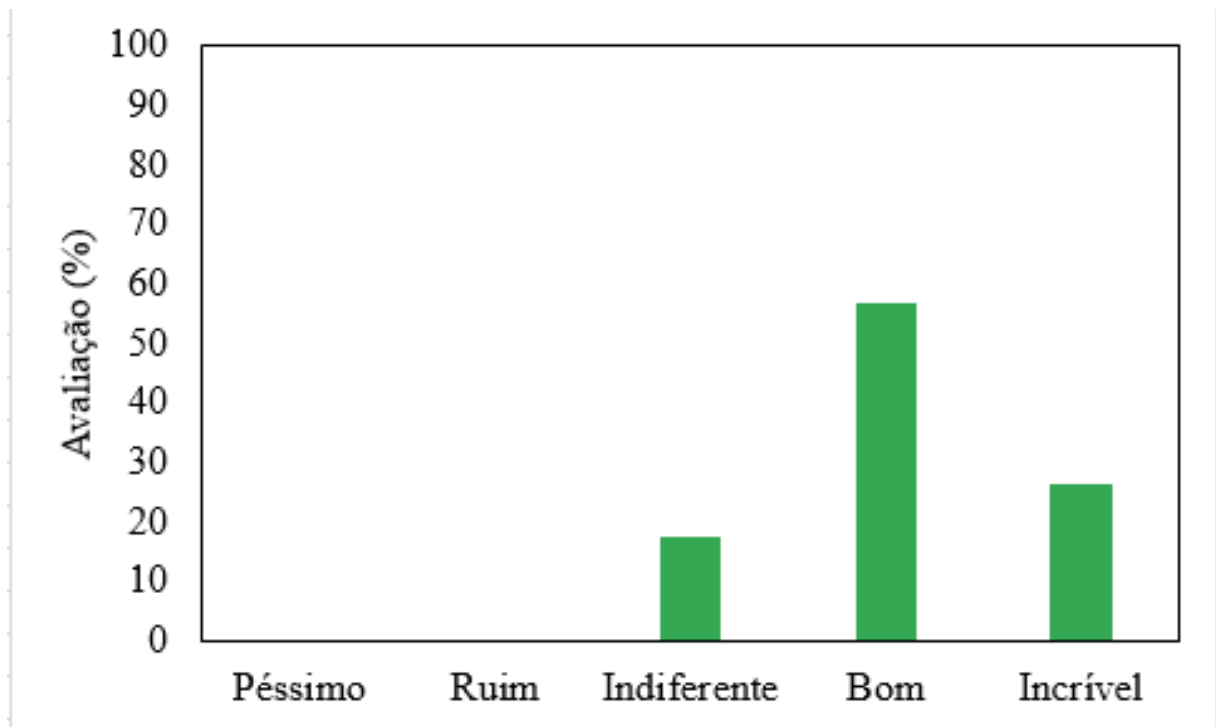
A Pergunta 1 tem como objetivo avaliar a receptividade do contexto da história do jogo e da maneira como foi apresentado. O texto apresentado no formulário de avaliação foi: “Qual nota você daria para a história e contextualização do jogo?”. Na Figura 17 pode-se observar o gráfico de avaliações relativo a pergunta 1. Como pode ser percebido no gráfico o resultado apresentou-se positivo, sendo que aproximadamente 60% das avaliações classificam a história do jogo como “Bom”.

A segunda pergunta tem como objetivo avaliar a possibilidade em aplicar a plataforma como ferramenta de auxílio no ensino de fundamentos de programação. Para essa pergunta o texto apresentado foi: “O que você acha de utilizar esse jogo para auxiliar no aprendizado de lógica de programação?”.

Pode ser observado na Figura 18 que aproximadamente 60% dos avaliadores reconheceram a possibilidade de utilizar a ferramenta como “Incrível”, enquanto aproximadamente 20% avaliaram a possibilidade como “Bom”. Analisando o gráfico fica evidente que a utilização do jogo como ferramenta educacional é interessante .

Os resultados da pergunta de número dois superam as expectativas por não apresentarem nenhum resultado de natureza negativa, visto que parte dos avaliadores pode nunca ter tido contato com fundamentos de programação. Essa pergunta é fundamental para a avaliação do protótipo pois demonstra a receptividade do público sobre a viabilidade na utilização do serious game como recurso educacional, que é o objetivo principal do desenvolvimento da plataforma.





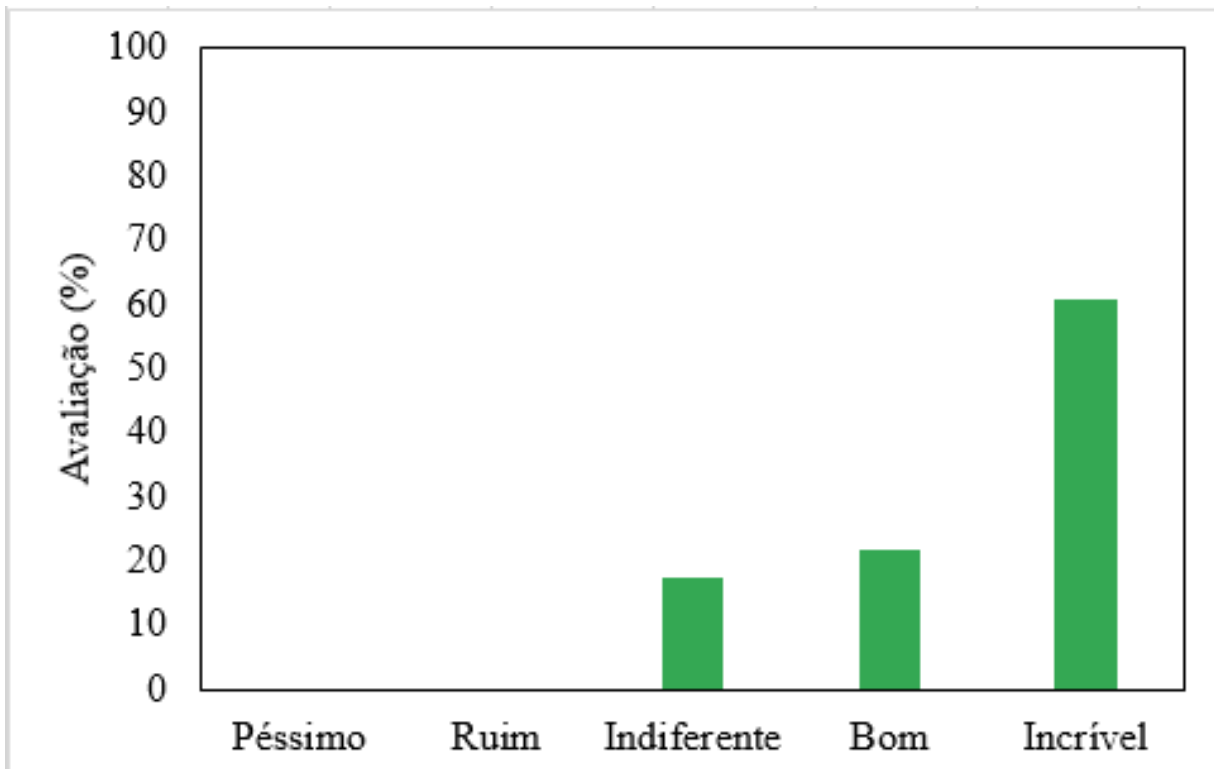
**Figura 17 – Gráfico de Resultados da Pergunta 1**

**Fonte: Autoria Própria**

A terceira pergunta avalia a percepção dos jogadores em relação a movimentação dos personagens, incluindo o comportamento dos objetos e a mecânica de batalha. No formulário de avaliação o texto apresentado para a pergunta foi: “Como você avaliaria as mecânicas do jogo? (Movimentação, Interação com Objetos, Batalha)”. Os resultados obtidos pela pergunta 3 são ilustrados pela Figura 19.

Essa pergunta apresentou alguns resultados negativos que eram esperados, já que o objetivo era tornar o jogo, de certa forma, desafiador e por consequência alguns jogadores podem apresentar frustrações por não serem familiarizados com jogos do gênero.

Durante os testes preliminares notou-se uma dificuldade para realizar o transporte dos cubos de forma precisa, o que resultou em frustrações nos jogadores, esses resultados preliminares podem ser o motivo das avaliações negativas. Apesar dos resultados dos testes preliminares, optou-se por manter a mecânica de transporte dos cubos para evitar que jogadores com maior familiaridade com jogos eletrônicos do gênero se sentissem entediados. No entanto, aproximadamente 40% dos avaliadores classificaram a mecânica do jogo como “Boa”, que apresenta uma porcentagem de votos muito próxima a “Indiferente”, sugerindo que a mecânica possa ser aprimorada para que obtenha uma maior receptividade.

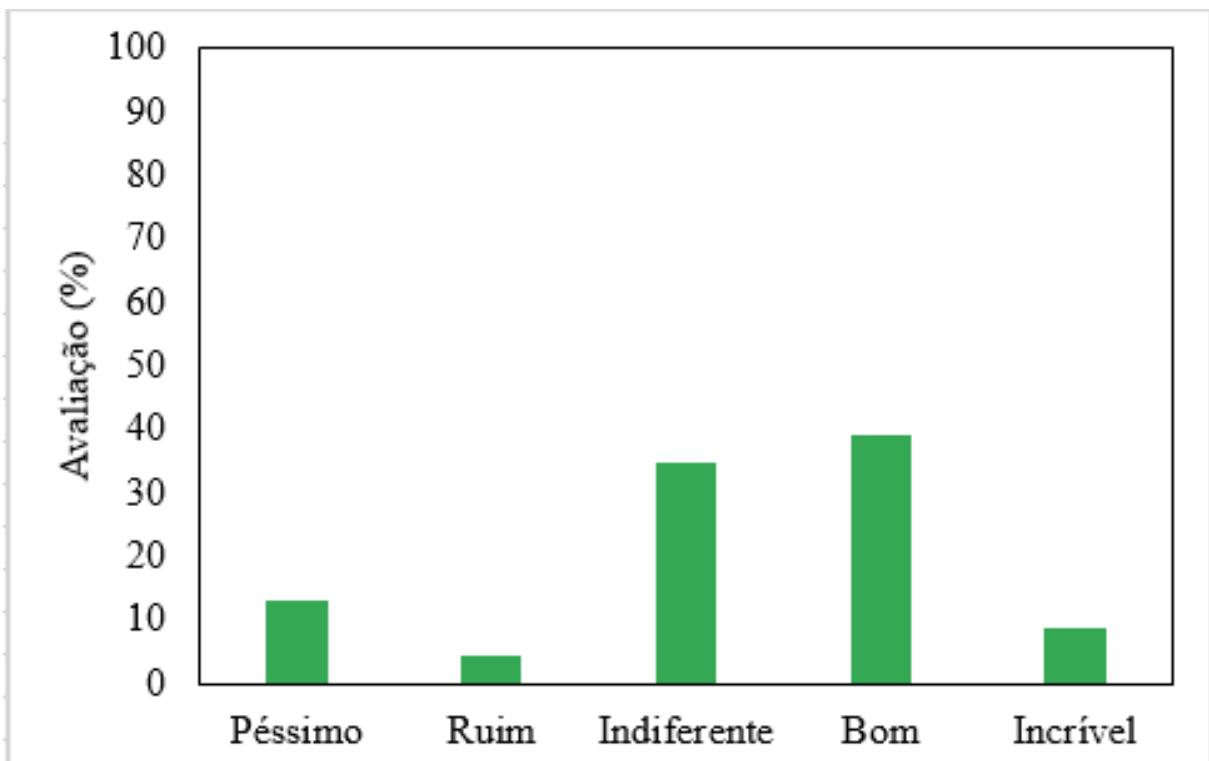


**Figura 18 – Gráfico de Resultados da Pergunta 2**

**Fonte: Autoria Própria**

A questão de número quatro tem como objetivo avaliar o aspecto didático sobre as funcionalidades do serious game e os conceitos necessários para progredir durante os níveis. O texto apresentado na pergunta de avaliação foi: “Como você avaliaria a explicação da forma como o jogo funciona e como é possível avançar durante os níveis?”.

Os resultados são apresentados na Figura 20 e são em sua maioria positivos. Nota-se que aproximadamente 40% dos avaliadores consideraram que o serious game foi apresentado de forma satisfatoriamente ou incrivelmente didática, sugerindo que a maior parte dos avaliadores tiveram facilidade em progredir durante o serious game sem ajuda externa. Esse aspecto é importante porque indica que é possível utilizar a plataforma para os alunos que possuem dificuldades comunicativas e preferem não se expôr para o professor ou para os colegas de classe.



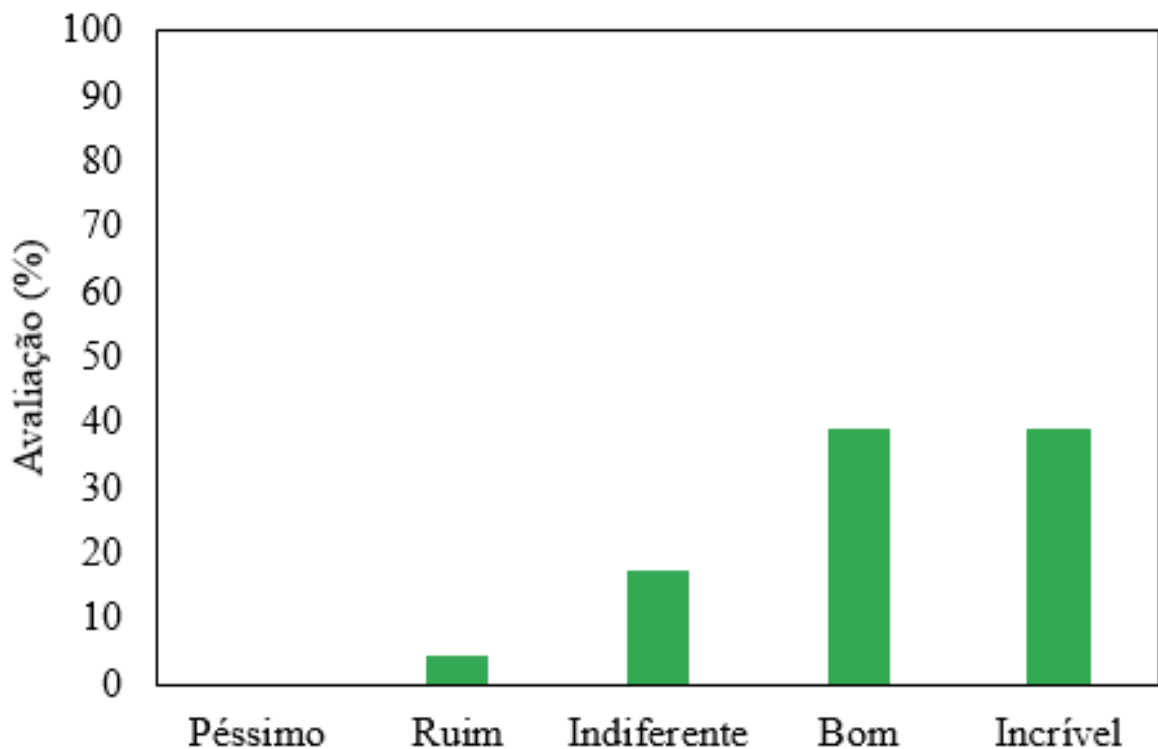
**Figura 19 – Gráfico de Resultados da Pergunta 3**

**Fonte: Autoria Própria**

A partir da quinta até a oitava questão, tem-se como objetivo identificar os níveis com maior e menor receptividade entre os avaliadores de acordo com os conceitos de fundamentos de programação que foram apresentados. Os quatro gráficos foram unificados na Figura 21 para facilitar a análise dos conceitos avaliados. Texto apresentado na pergunta: “Como você avaliaria a forma como os conceitos de lógica computacional foram apresentados?”. Este texto foi repetido nestas quatro perguntas, apenas variando o conteúdo, sendo o conteúdo “variável”, “Condicional”, “Laço de repetição” e “Tipos de dados”, para a pergunta 5, 6, 7 e 8, respectivamente.

Os níveis iniciais apresentaram um tempo maior de trabalho e aprimoramento, portanto, esperava-se que teriam uma receptividade maior em relação aos níveis finais. No entanto, observando-se o gráfico, percebe-se que todos os níveis obtiveram notas positivas.

De acordo com o gráfico é possível concluir que os avaliadores foram receptivos à forma como os conceitos foram apresentados para a maioria dos casos considerando que a maior parte das avaliações tendem a um aspecto positivo. O valor péssimo não foi atribuído para nenhum dos conteúdos, já os resultados de ruim para variável e condicional foram muito baixos podendo ser classificados como pontos discrepantes.



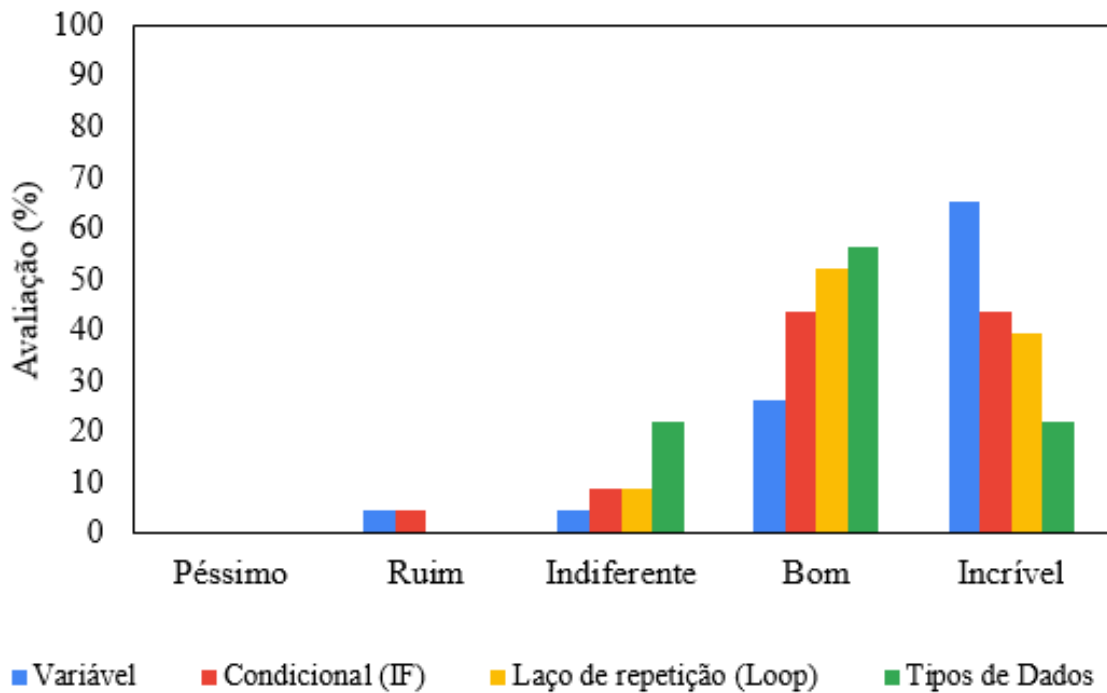
**Figura 20 – Gráfico de Resultados da Pergunta 4**

**Fonte: Autoria Própria**

A pergunta de número nove tem como o objetivo avaliar a teoria do fluxo, onde pretende-se deixar o jogador motivado utilizando a oscilação entre desafio e progresso, para evitar o estado de tédio e frustração durante as seções do jogo.

O texto apresentado na pergunta foi o seguinte: “Você acredita que o progresso do jogo possui um equilíbrio de dificuldade bem aplicado? (Nem muito fácil, nem muito difícil)”.

Os resultados das avaliações podem ser observados na Figura 22. De acordo com o gráfico nota-se que 50% do público que avaliou o protótipo manteve-se em estado de fluxo durante os testes, já que classificaram esse quesito como incrível. Esperava-se que houvessem resultados negativos devido a alguns dos jogadores possuírem muita ou nenhuma familiaridade com os conceitos apresentados, principalmente em relação as mecânicas, e essa característica pode influenciar muito na dificuldade para progredir no jogo. Contudo, aproximadamente 50% do público classificou o aspecto questionado como “Incrível”, o que superou as expectativas.



**Figura 21 – Gráfico de Resultados das Perguntas 5 a 8**

**Fonte: Autoria Própria**

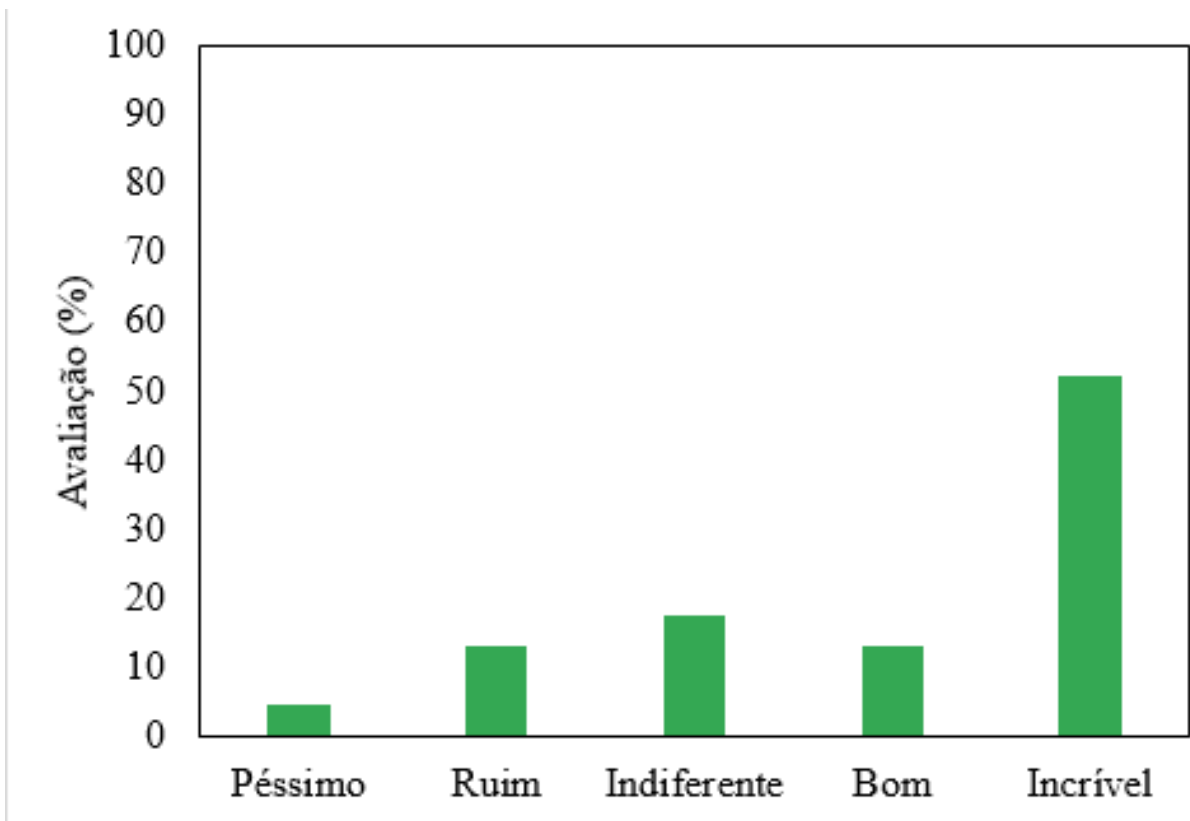
A décima pergunta busca avaliar a estética presente no serious game, relativo as escolhas dos gráficos, sons e contextualização. Os resultados são apresentados na Figura 23. Pode-se concluir que a receptividade foi positiva em sua parte marjoritária, variando entre “Bom” e “Incrível”.

A pergunta de número onze também tem conexão com a teoria do fluxo, considerando que, se os avaliadores acharem o jogo divertido a chance de se sentirem entediados ou frustrados é reduzida. Nessa pergunta o texto apresentado foi: “Qual nota você daria para a diversão do jogo?”.

Os resultados da pergunta onze podem ser observados na Figura 24 e demonstram que a maioria dos avaliadores encontrava-se em estado de entretenimento, já que aproximadamente 60% avaliou a diversão do serious game como “Bom”, e em segundo lugar, cerca de 30% como “Incrível”.

A pergunta de número doze busca avaliar a composição do jogo por completo considerando todos os aspectos apresentados anteriormente. Texto apresentado na pergunta: “De uma maneira geral, que nota você daria para o jogo?”.

É possível visualizar pela Figura 25 que não houve nenhum resultado considerado negativo, onde em torno de 50% dos avaliadores considerou o jogo “Bom” e aproximadamente



**Figura 22 – Gráfico de Resultados da Pergunta 9**

**Fonte: Autoria Própria**

40% considerou o jogo “Incrível”.

Levando em consideração todos os conceitos apresentados pelo serious game, pode-se perceber que foram bem recebidos pelo público. Isso é comprovado pelos resultados da pergunta doze onde o jogo foi classificado, de forma geral, como “Bom” pela maioria dos avaliadores.

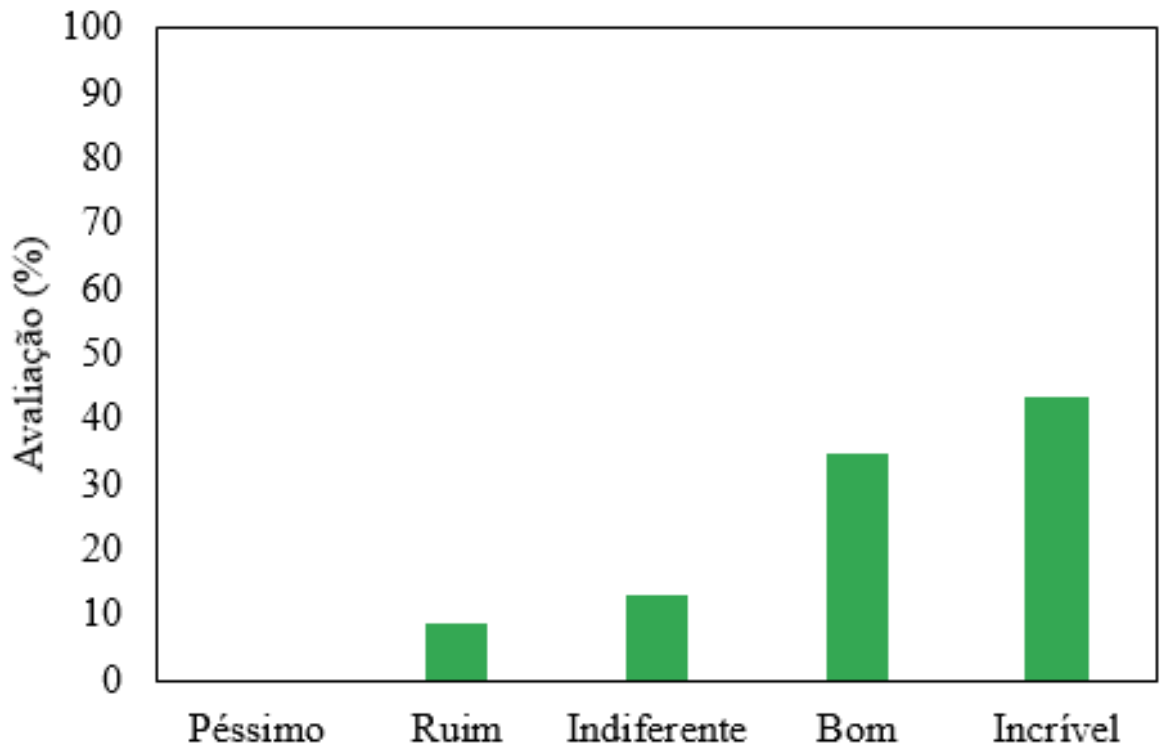


Figura 23 – Gráfico de Resultados da Pergunta 10

Fonte: Autoria Própria

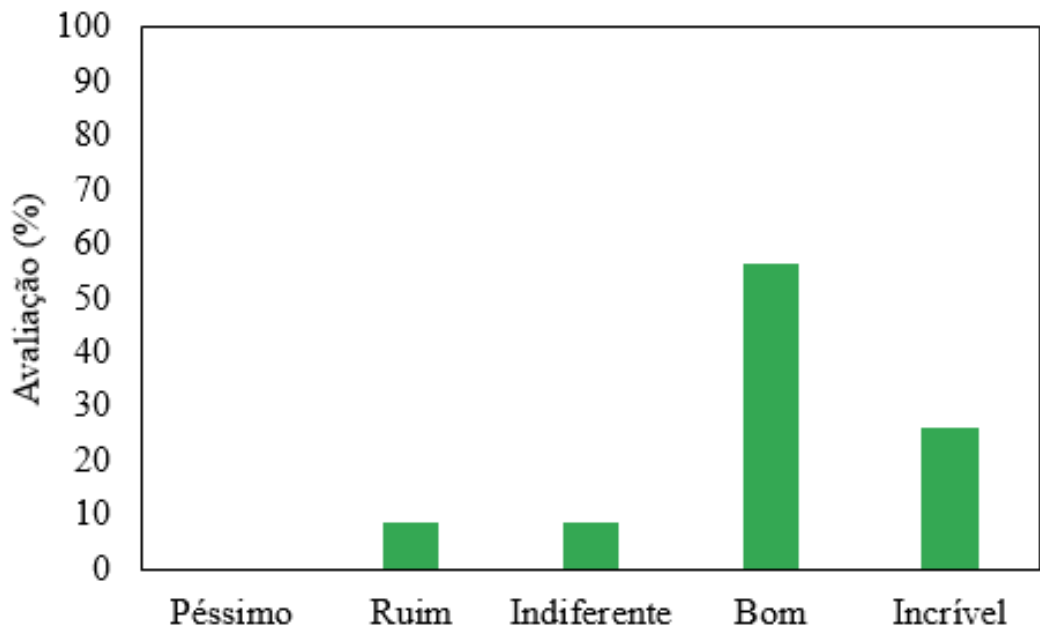
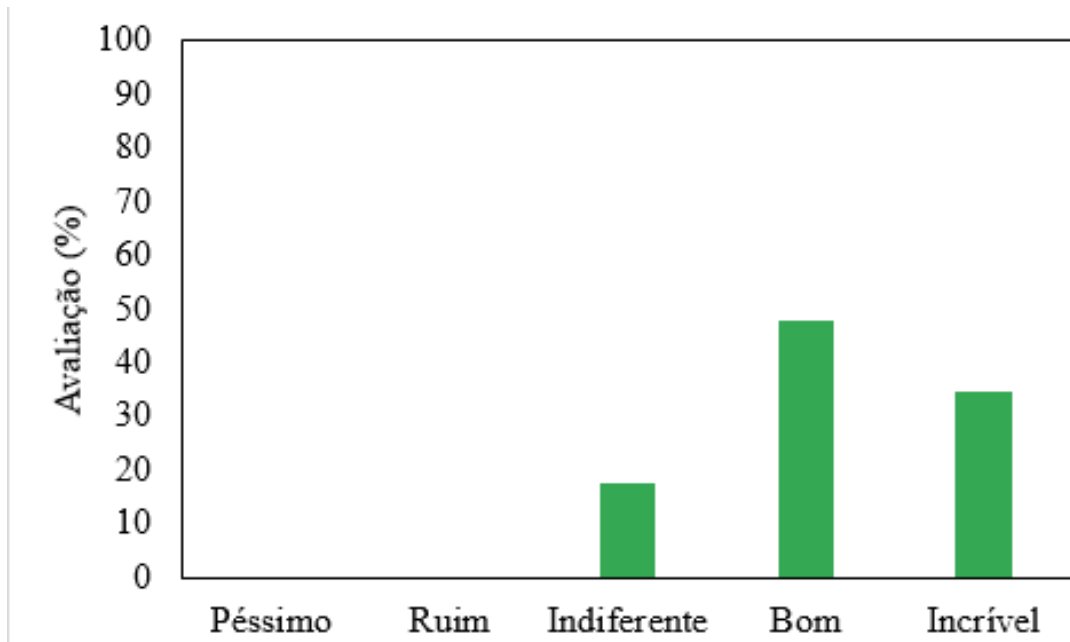


Figura 24 – Gráfico de Resultados da Pergunta 11

Fonte: Autoria Própria



**Figura 25 – Gráficos de resultados da pergunta 12**

**Fonte: Autoria Própria**



## 5 CONCLUSÕES

Este trabalho apresentou uma estrutura de desenvolvimento de um serious game do gênero RPG com foco no auxílio da aprendizagem de lógica computacional, um protótipo funcional aplicável e a avaliação do mesmo. A estrutura construída para o desenvolvimento do jogo possibilita que o mesmo seja aplicado para a aprendizagem de diversos conteúdos. O serious game foi desenvolvido visando sanar algumas das dificuldades apresentadas das abordagens tradicionais de ensino, entre elas, a discrepância entre os níveis de conhecimento dos alunos.

Com base nas avaliações, percebeu-se que o protótipo foi bem recebido pelo público. Em todas as perguntas de avaliação, com exceção da pergunta de número três, mais de 60% das avaliações tiveram resultados positivos, deixando claro que sua aplicação como recurso educacional é viável.

### 5.1 PRINCIPAIS CONTRIBUIÇÕES

- Auxiliar na aprendizagem de lógica computacional;
- Apresentar alternativas de metodologias de ensino;
- Introduzir conceitos de desenvolvimento de jogos;
- Explorar métodos motivacionais para a aprendizagem;
- Facilitar a aplicação de metodologias ativas.

## 5.2 LIMITAÇÕES

- A plataforma é limitada apenas para o sistema operacional Windows e para quem sabe operar um computador, além de ter como requisito o domínio da língua portuguesa e de matemática básica;
- A avaliação limitou-se a 23 voluntários com uma faixa etária entre 18 e 50 anos.

## 5.3 TRABALHOS FUTUROS

- Implementar uma fase contextualizada que exemplifique o conceito do Laço de Repetição “Para”;
- Implementar uma fase contextualizada que exemplifique o conceito de “Funções”;
- Introduzir o conceito do tipo de dados “Decimal”;
- Adaptar os controles para dispositivos móveis;
- Armazenar o tempo de duração dos níveis com objetivo de avaliar o desempenho do jogador;
- Implementar um contador de fontes de conhecimento adquiridas.

## REFERÊNCIAS

ABT, C. C. Serious games. **Serious games**, 1987.

AGUIAR, E. V. B. As novas tecnologias e o ensino-aprendizagem. **Vértices**, v. 10, p. 63–72, 2008. ISSN 14152843. Disponível em: <<http://essentiaeditora.iff.edu.br/index.php/vertices/article/viewArticle/34>>.

BASTOS, S.; ADAMATTI, D. F.; CARVALHO, F. A. H. D. “O Ensino de Lógica de Programação no Ensino Médio e suas aplicações na Neurociências”. n. Sbie, p. 459–468, 2015.

CAMBRUZZI, E.; SOUZA, R. M. D. Robótica Educativa na aprendizagem de Lógica de Programação : Aplicação. n. Wie, p. 21–28, 2015.

CASSENOTE, M. R. S.; ANTONIAZZI, R. L. O Ensino De Lógica Para Programação a Partir De Jogos Educativos : Um Relato De Experiência. v. 3, p. 269–280, 2015.

FARDO, M. L. A Gamificação Aplicada Em Ambientes De Aprendizagem. **RENOTE - Revista Novas Tecnologias na Educação**, v. 11, n. 1, p. 1–9, 2013. ISSN 16791916. Disponível em: <<http://seer.ufrgs.br/renote/article/view/41629>>.

FRANÇA, R.; TEDESCO, P. Desafios e oportunidades ao ensino do pensamento computacional na educação básica no Brasil. n. Cbie, p. 1464, 2015. ISSN 2316-8889. Disponível em: <<http://br-ie.org/pub/index.php/wcbie/article/view/6331>>.

GARCIA, R. E.; CORREIA, R. C. M.; SHIMABUKURO, M. H. Ensino de Lógica de Programação e Estruturas de Dados para Alunos do Ensino Médio. **Xiii Workshop sobre Educação em Computação**, p. 246–249, 2008.

GOMES, T. C. S.; MELO, J. C. B. D. O Pensamento Computacional no Ensino Médio : Uma Abordagem Blended Learning. p. 651–660, 2005.

GUEDES, L.; SILVA, J. B. L. da. Aluna do 7 Semestre do Curso de Pedagogia do campus Universitário de Sinop – UNEMAT. Pertence ao Grupo de Orientação do professor Me. João Batista Lopes da Silva. Graduado em Licenciatura Plena em Educação Física pela Universidade Federal de Cruz Alta, R. p. 161–171, 2012.

JÚNIOR, J. P.; RAPKIEWICZ, C. Ensino de Algoritmos e Programação: Uma Experiência no Nível Médio. **XXV Congresso da Sociedade Brasileira de Computação**, p. 2351–2362, 2005. Disponível em: <<http://scholar.google.com/scholar?hl=en\&btnG=Search\&q=intitle:Ensino+de+Algoritmos+e+Program>>.

KOSLOSKY, M. Aprendizagem baseada em casos um ambiente para ensino de lógica de programação. 1999. Disponível em: <<https://repositorio.ufsc.br/handle/123456789/81012>>.

LIMA, J. W. ferreira; MEIRELES, A. M. R. Lima2014.pdf. Revista Expressão Católica, p. 5, 2014.

MANZANO, J. A.; OLIVEIRA, J. **Algoritmos: Lógica para desenvolvimento de programação de computadores**. 2005. 221 p.

MELO, L. D. A.; KELLY, T.; COSTA, D. L.; CAROLINY, A. Pense bem : proposta e desenvolvimento de jogo digital para ensino de computação na educação básica. n. Cbie, p. 346–355, 2013.

MIZUKAMI, M. d. G. N. ENSINO: As abordagens do processo. **Editora Pedagógica e Universitária**, p. 1–13, 1986. Disponível em: <<http://nead.uesc.br/arquivos/Biologia/mod4bloco4/ep4/ABORDAGENS-DO-PROCESSO.pdf>>.

MONTESSORI, M. **The Montessori Method : the origins of an educational innovation: including an abridged and annotated edition of Maria Montessori's The Montessori Method**. [s.n.], 2004. 295 p. ISSN 1098-6596. ISBN 9788578110796. Disponível em: <<http://www.arvindguptatoys.com/arvindgupta/montessori-new.pdf>>.

NAVARRO, G. A transformação do conceito do termo jogo no contexto da pós-modernidade. **Biblioteca Latino-Americana de Cultura e Comunicação**, v. 1, 2013.

NETO, W. C. B.; SCHUVARTZ, A. A. Ferramenta Computacional de Apoio ao Processo de Ensino-Aprendizagem dos Fundamentos de Programação de Computadores. **SBIE - Simpósio Brasileiro de Informática na Educação**, v. 17, p. 520–528, 2007. ISSN 2316-6533. Disponível em: <<http://br-ie.org/pub/index.php/sbie/article/view/601>>.

OLIVEIRA, M. L. S. de; SOUZA, A. A. de; BARBOSA, A. F.; BARREIROS, E. F. S. Ensino de lógica de programação no ensino fundamental utilizando o Scratch : um relato de experiência. v. 12, p. 1493–1502, 2014.

PICADO, L. A indisciplina em sala de aula: Uma abordagem comportamental e cognitiva. **O Portal dos Psicólogos**, p. 1–14, 2009. Disponível em: <<http://www.psicologia.pt/artigos/textos/A0484.pdf>>.

RANGEL, M. **Métodos de ensino para a aprendizagem e a dinamização das aulas**. 2006. 94 p.

RITTERFELD, U.; CODY, M.; VORDERER, P. **Serious games: Mechanisms and effects**. [S.l.: s.n.], 2009. 1–530 p. ISSN 1431-9276. ISBN 0203891651.

SCHELL, J. **The art of game design**. [S.l.: s.n.], 2008. 489 p. ISSN 1098-6596. ISBN 9788578110796.

ZANETTI, H. A. P.; BORGES, M. A. F.; LEAL, V. C. G.; MATSUZAKI, I. Y. Proposta de ensino de programação para crianças com Scratch e Pensamento Computacional. v. 4, p. 43–58, 2017.

ZICHERMANN, G.; CUNNINGHAM, C. **Gamification By Design**. [S.l.: s.n.], 2011. 208 p. ISSN 14602180. ISBN 9781449397678.