

# UTILIZAÇÃO DE GAMEFICAÇÃO NA IMPLEMENTAÇÃO DE UMA FERRAMENTA DE APOIO AO APRENDIZADO DE ALGORITMOS

Gustavo Reis Rocha Pitta Travnik, Giuliano P.M. Giglio

Centro de Ensino Superior de Juiz de Fora (CES-JF)  
Juiz de Fora - MG- Brasil

gustavotravnik@outlook.com, giucontato@gmail.com

**Resumo:** *A necessidade de contrapor a tradição com a inovação requer dos profissionais de educação novas abordagens no ensino escolar, devido ao crescimento do desinteresse dos alunos com os métodos tradicionais de ensino. O ensino de algoritmos computacionais não foge à regra, sendo uma disciplina primordial para a área de computação. Os jogos computacionais são um exemplo de tecnologia atrativa na capacitação profissional. Portanto, esse trabalho consiste em utilizar a gameficação na elaboração de um jogo educativo para apoio ao ensino de algoritmos nas disciplinas que abordam o assunto, promovendo uma maior motivação aos alunos na obtenção do conhecimento de seus conceitos básicos.*

## 1. Introdução

Nos dias atuais observa-se um desinteresse constante da juventude ou a pouca motivação pelos métodos tradicionais de estudo e pela própria escola. O real motivo de tal comportamento possui inúmeras causas como: didática dos professores, conteúdo aparentemente desinteressante ou por não terem ideia da importância do conteúdo programático (PEZZINI e SZYMANSKI, 2017). Segundo o economista Marcelo Néri, "se não revolucionarmos a capacidade de atrair o jovem, nada vai mudar esse quadro" (M. MÁIA, 2016).

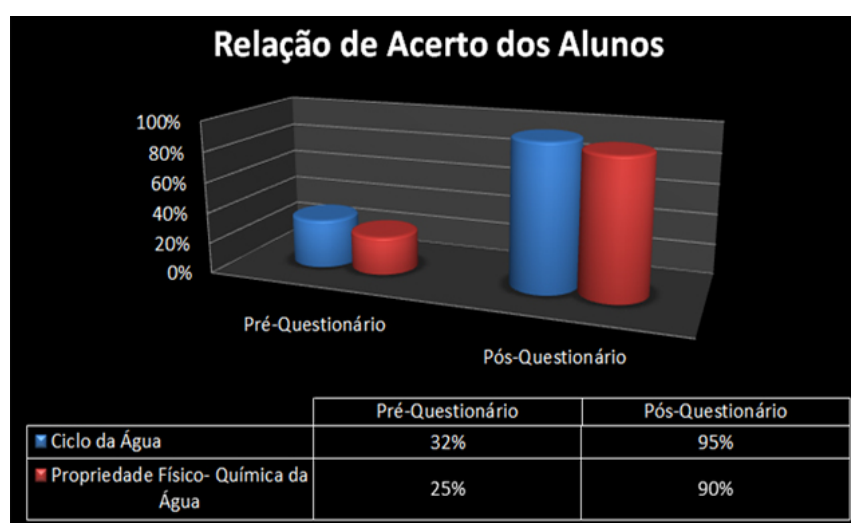
Esse desinteresse se estende também ao ensino fundamental, como pode ser descrito segundo o artigo de Maria Fabiana Nascimento de Carvalho (CARVALHO et Al., 2017). Um dos fatores para esse desinteresse dar-se-á mediante ao fato que os brinquedos iterativos (Brinquedos Iterativos, 2017), estão chamando mais a atenção das crianças, onde as mesmas deixam seu estudo em segundo lugar para poderem interagir com os brinquedos.

Atualmente, observa-se que o método de ensino não chama tanto a atenção comparado com algo que seja mais interativo, como por exemplo jogos eletrônicos e atividades de lazer.

Devido a esse desinteresse, podemos presumir que a forma de ensino deve sofrer alterações para que possa atrair mais os jovens a estudar, e uma das saídas é incentivar os alunos a buscar o conhecimento de maneira autônoma (PEZZINI e SZYMANSKI, 2017), logo, a instituição de ensino deve também implementar métodos para despertar o interesse nos alunos a buscarem conhecimento.

Com a era digital, temos diversas ferramentas de auxílio ao ensino que podem ajudar os alunos a se desenvolverem mais rápido (SILVA e SOUZA, 2016), pois os métodos comuns de ensino podem não suprir as necessidades da nova geração que vem crescendo juntamente com a tecnologia.

Em uma recente pesquisa feita por pesquisadores da Universidade Unigranrio nos cursos de Marketing e Recursos Humanos, foi realizada uma entrevista para coletar opiniões diretamente dos alunos, e na mesma pode-se entender que o conteúdo e a forma com que o mesmo é passado aos alunos durante a aula interfere diretamente no interesse e aprendizado (VIANA, 2016). Para colaborar com esse panorama, há uma experiência utilizando esse tipo de recurso que teve uma taxa de aprovação bem satisfatória, onde foram realizados testes utilizando uma ferramenta simples desenvolvida por Luciane Barro de Santana, chamada “O jogo QuizQuím-Química Divertida” (SILVA et Al., 2016). Consiste em um *Quiz* (jogo de perguntas e respostas) sobre química, onde cada aluno individualmente podia realizar a interação com a ferramenta. Os resultados obtidos podem ser observados na Figura 1 abaixo.



**Figura 1:** Resultado do nível de acerto dos alunos antes e depois da aplicação da ferramenta.  
**Fonte:** QuizQuím- Química Divertida (SILVA et Al., 2016)

De acordo com o gráfico, pode-se perceber que antes da aplicação da ferramenta, a taxa de acerto nas questões sobre o ciclo da água era de 32% e sobre a Propriedade físico-química da água era de apenas 25%, e após a aplicação da ferramenta, o aumento da taxa de acerto foi de 63% no ciclo da água e de 65% nas propriedades. Neste caso a aplicação da ferramenta foi de grande eficácia. Já os métodos de ensino mais interativos, além de chamarem a atenção dos alunos, ainda têm a capacidade de fazê-los aprender mais.

Devido aos resultados obtidos na aplicação da ferramenta, pode-se notar que a presença desta multidisciplinaridade torna-se uma figura bem importante no ensino em geral, pois com a utilização desse tipo de ferramenta, tende-se a ter um índice de aprendizado mais elevado e com uma curva de tempo reduzida.

Os recursos de informática vêm crescendo nas escolas de nível médio e fundamental com a utilização de ferramentas de apoio, como os jogos e softwares educacionais, pois a exploração de tais ferramentas permite transformar o conteúdo didático em algo mais lúdico e atrativo aos jovens, melhorando assim a qualidade de ensino e o interesse dos alunos em relação a matéria (MELO, et Al., 2016).

### 1.1 Métodos de Ensino de Algoritmos Computacionais

O conceito algoritmo serve como uma ferramenta genérica para a resolução de problemas, onde temos uma sequência de ações que devem ser feitas para que se possa obter o resultado final (PRUSS, 2016). Uma receita de bolo é considerada um algoritmo.

Como citado acima, o algoritmo é usado para a resolução de problemas reais, e também como forma de melhorar o raciocínio. Assim como em qualquer problema, podemos buscar maneiras de resolvê-lo, buscar outros caminhos para completar a tarefa, entre outros. Para isso, deve-se elaborar uma sequência de passos para serem executados. Os algoritmos computacionais são aqueles processados pelo computador, onde temos todos os passos para, sequencialmente, definir o caminho da solução buscada.

O uso de ferramentas para auxílio ao ensino de algoritmos tem se apresentado muito eficaz, pois os mesmos possuem um método de aprendizado diferenciado dos métodos comuns que são muitas vezes rudimentares, como o ensino de algoritmos diretamente através de uma plataforma de desenvolvimento (IDE<sup>1</sup>). Todo esse processo deve ser feito utilizando a própria linguagem de programação e suas abstrações e complexidade.

Há várias propostas na literatura para o apoio ao ensino de algoritmos, e uma das mais conhecidas é o "VisuAlg", que se trata de uma interface de programação onde a linguagem utilizada se deriva do "Portugol", que utiliza o português para criar sua lógica estruturada, sendo assim, o aluno desenvolve o sistema totalmente em português, o que facilita o entendimento da estrutura do algoritmo (Ferramenta VisuAlg, 2016).

Há outras ferramentas mais interativas graficamente e elas serão apresentadas na seção 2, de modo que o usuário possa fazer o uso de elementos visuais para a manipulação dos dados na tela, além da beleza a atratividade dos detalhes gráficos, assim como grupos de estudos e movimentos de aprendizado.

Tem-se atualmente uma geração social onde os elementos audiovisuais, tecnologias e métodos estão muito diversificados e, além de avançados, continuam em rápido avanço, o que faz com que as coisas mais antigas ou tradicionais fiquem mais desinteressantes aos olhos da nova geração, ou geração Y. Com isso o método de ensino também deve sofrer modificações para que se possa atender os requisitos de atenção dos alunos, que desejam coisas mais práticas, mais interessantes e mais dinâmicas, entre outras. O que pode ser feito para tentar contornar isso é adicionar ou substituir alguns métodos por técnicas mais dinâmicas e divertidas, que fazem com que os alunos mantenham uma motivação alta em se aprender e praticar várias disciplinas, sobretudo algoritmos e suas ramificações, que ainda hoje possuem métodos considerados rudimentares ou tradicionais de ensino.

E um dos meios que temos para implementar essas técnicas, é a gamificação, do inglês *Gamefication*, que é um conceito cujo papel é transformar uma atividade rotineira em um jogo (USP, 2016), onde temos os mesmo elementos que um jogo teria, como por exemplo:

**Distração:** Jogos geralmente são atividades mais livres, onde nem sempre se tem obrigação de cumprir um roteiro fixo, o que faz com que o jogador (aluno) não se sinta preso àquilo, e possa fazer o trabalho de forma mais divertida e tranquila.

**Metas:** que é o objetivo principal da sua atividade, o resultado esperado de todo o seu esforço.

---

<sup>1</sup> Que seria o local onde se é digitado os códigos e manipulado os objetos para que o mesmo gere um resultado final

**Regras:** são os limites impostos pela atividade em fim de conseguir cumprir as metas, como por exemplo, qual o caminho que deve ser seguido ou as atividades permitidas em cada momento do jogo.

**Feedback:** é o resultado das atividades feitas pelo jogador (aluno), fornecidas como pontuação, prêmios, incentivos, entre outros.

A proposta deste trabalho é desenvolver uma ferramenta alternativa de ensino de algoritmos e programação para alunos de algoritmos computacionais, utilizando o conceito de gamificação, possuindo como diferencial vários desafios e objetivos integrados aos conceitos da disciplina, contendo características interativas e de entretenimento, a fim de incentivar o aluno nas práticas de seu aprendizado em programação.

Para isso, o trabalho está estruturado, em sua primeira unidade, contextualizar os aspectos de aprendizado atuais de algoritmos computacionais e a necessidade de maior interação a fim de aumentar a motivação e rendimento do aluno. Na Seção 2, será apresentada a gamificação e suas principais características, além de sua aplicação no suporte ao aprendizado em geral. Para atender aos objetivos gerais deste trabalho, será desenvolvido e apresentado na Seção 3 e na Seção 4 uma proposta de gamificação para o aprendizado de algoritmos, através de uma ferramenta interativa e motivacional. Nas sessões cinco e seis, serão analisados os resultados alcançados, delineadas as vantagens e desvantagens de seu uso, além dos trabalhos futuros propostos a partir de então.

## **2. A Utilização da Gamificação na Construção de Aplicações de Suporte ao Aprendizado**

Apesar do nome gamificação lembrar “Jogo”, não se trata especificadamente da criação de um, mas sim o uso de seus elementos interativos e sua capacidade de prender a atenção do usuário (FARDO, 2016). Basicamente, a gamificação é o uso de elementos encontrados nos jogos, como a dinâmica e o poder de prender a atenção do usuário, em diversas outras aplicações (LIVRO GAMIFICATION, 2016). Com frequência cada vez maior, esse conjunto de técnicas tem sido aplicado em campos variados, tais como saúde, educação, políticas públicas e esportes.

Uma escola pública de Nova York, chamada *Quest to Learn*, tem como proposta estruturar o funcionamento das aulas presenciais transformando a rotina da escola em um grande jogo com regras e metas bem definidas. De acordo com o manual para pais e alunos da escola, o estudante é inserido em uma realidade de desafios, tendo como plano de fundo o conhecimento visto durante as aulas, a colaboração entre os alunos para chegar aos resultados esperados e a organização sistemática do raciocínio (NAVARRO, 2016). O objetivo é estimular a necessidade de aprender, motivando e criando no estudante o interesse em adquirir conhecimento.

De acordo com Huizinga (HUIZINGA, 2017), é possível elencar oito características principais e comuns a todas as atividades consideradas jogos:

1. **Participação voluntária:** o jogo é uma atividade na qual todos os participantes são livres para fazer parte, ou não.

2. **Distração:** o jogo não é uma obrigatoriedade, portanto, não pode ser considerado uma tarefa. Dessa forma, é praticado nas horas de ócio, como distração.
3. **Exterior à “realidade”:** o jogo é a evasão da vida real para uma esfera paralela de tempo e espaço, não podendo ser considerado parte integrante do cotidiano.
4. **Limites espaciais e temporais:** como o jogo é a evasão da vida real, requer espaço e duração delimitados para a sua prática. Essas limitações são responsáveis por deslocar o participante para a realidade paralela do jogo.
5. **Meta:** o jogo possui objetivo definido e claro para todos os participantes.
6. **Regras:** para alcançar a meta, deve-se agir de acordo com as regras determinadas, com o objetivo de inserir os participantes na realidade paralela do jogo.
7. **Sistema de feedback (resultados):** considerando a existência de uma meta a ser atingida, conseqüentemente, haverá um resultado, representando o alcance, ou não, dessa meta. Portanto, é necessário determinar um sistema de contagem de pontos ou avaliação de feedback, a fim de definir claramente o resultado do jogo entre os participantes. Em um jogo, não existe dúvida quanto ao alcance, ou não, do objetivo final por parte dos seus jogadores.
8. **Término:** o jogo sempre acaba.

Nota-se uma semelhança com o algoritmo, pois ambos possuem uma seqüência de passos a serem seguidos e interpretados, fazendo com que o usuário final seja estimulado a fazer o uso de lógica para conseguir resolver os desafios propostos pelo sistema.

Segundo Huizinga (HUIZINGA e LUDENS, 2017), a gameficação adiciona características que alteram e aprimoram as tarefas a serem realizadas, colocando na mesma uma meta/objetivo, proporcionando uma motivação para o qual o usuário realizará aquela tarefa, e logicamente, adicionando uma recompensa a dedicação do participante àquele trabalho, como um placar onde ficam seus pontos ou até mesmo prêmios físicos, como uma corrida, onde os três primeiros colocados ganham uma medalha.

Considerando o ensino voltado à informática, o nível de interatividade e eficácia da gameficação na maioria das vezes é mais fácil de ser implementada, pois o ambiente disciplinar já possui uma estrutura computacional mais avançada, e seus alunos já têm um contato muito maior com os meios eletrônicos.

Com o avanço das linguagens de programação e marcação, como o HTML5, ficou mais fácil a criação e implementação de ferramentas que fazem o uso da gameficação, pois é possível a criação de um ambiente mais amigável e fácil de ser modificado, como por exemplo, o uso de um conceito na programação chamado *Reflection* (Revista Devmedia, 2017), que por sua vez é capaz de interpretar um código digitado pelo usuário sem a necessidade do uso de uma plataforma tradicional como uma IDE.

Num levantamento realizado, encontrou-se diversas ferramentas que foram feitas para esse propósito, sendo algumas delas:

- **Scratch:** Essa ferramenta é uma das mais conhecidas por ser muito interativa e atrativa para o público mais jovem, geralmente entre 8 a 16 anos, a mesma permite a criação de pequenos jogos e animações utilizando um sistema de programação com interface amigável, pois os comandos são representados como peças que o aluno pode encaixar umas na outra formando um sistema

que pode manipular imagens e sons. Como ela é uma plataforma *online*, a mesma pode ser aplicada em escolas e faculdades, sem a necessidade de instalação ou preparação de ambiente. (Ferramenta Scratch, 2016)

- **RoboCode:** É um programa pequeno desenvolvido em Java para usuários que querem aprender um pouco desta linguagem brincando. Trata-se de uma arena de combate onde blindados de guerra se enfrentam até que reste apenas um sobrevivente, ou um time. O detalhe está no controle, ao invés de se manipular com teclado e mouse, deve-se programá-los para combater por conta própria, e como a lógica é mais simples e fácil de aprender, a pessoa se diverte e passa se interessar em aprender a linguagem proposta. (Ferramenta RoboCode, 2016)
- **Code Academy:** A Codeacademy é uma instituição educacional que disponibiliza totalmente gratuito o conteúdo de ensino de algoritmos e de diversas linguagens de programação. Pode-se ter todo o conteúdo online com exemplos e tutoriais disponibilizados pela própria Codeacademy. Podemos citar alguns cursos disponíveis, dentre eles HTML e CSS, JQuery, PHP, Python e Ruby. (Instituição Code Academy, 2016)
- **Alice:** É um ambiente 3D, onde o aluno cria um algoritmo para manipular os objetos que são adicionados na tela através de uma palheta de recursos, somente é necessário arrastar e soltar o objeto desejado para o local preferido, após isso, pode-se usar Java, C++ ou C# para a programação do seu sistema. Com Alice, é possível criar animações e jogos simples, que chamam muito a atenção dos jovens, apesar de requerer um pouco mais de conhecimento do aluno, ele estimula o mesmo a se desempenhar, pois as possibilidades de criação são diversas e simples. (Ferramenta Alice, 2016)

Diferente das ferramentas citadas acima, a proposta deste artigo é criar um jogo, onde o aluno possuirá uma imersão igual a qualquer outro jogo não educativo, fazendo-o esquecer que aquele sistema tem o objetivo de ensinar algo. Devido a isso, não basta somente ter o conhecimento da lógica computacional para conseguir passar pelos desafios, também é necessário a habilidade de jogador para perseverar.

### 3. Uma Proposta de Gamificação para Aprendizado de Algoritmos

Com base nessas pesquisas e com a possibilidade da gamificação ser algo útil e produtivo, será desenvolvido uma plataforma utilizando os conceitos com a proposta de agilizar e incentivar o estudo de algoritmos computacionais.

A proposta baseia-se na criação de um jogo de alta interatividade com o aluno, onde cada objetivo do jogo, realiza a introdução e treinamento de uma tarefa da disciplina, como por exemplo, comandos de condição, onde determinado evento somente acontece se alguma condição imposta pelo jogo for verdadeira. O sistema de recompensa do jogo é dividido em dois módulos:

- **Módulo 01:** O aluno ou grupo possui uma pontuação que pode ser comparada com a dos demais que fazem o uso do sistema, criando assim um clima de competição, onde o aluno com maior pontuação terá mais itens no ranque, como medalhas, troféus, pontuação, entre outros.
- **Módulo 02:** A cada desafio conquistado pelo aluno ou grupo, novos caminhos no jogo são abertos, como novos desafios, ou áreas de lazer,

onde o aluno poderá ter um tempo de distração, fazendo assim o aprendizado mais divertido e menos linear, voltado somente para a matéria.

Cada fase do jogo será um exercício diferente da disciplina, e para o avanço no jogo, o mesmo deve vencer os desafios até chegar ao final do mesmo. A forma escolhida para a apresentação da disciplina dentro do jogo é um instrutor virtual, que possuirá uma inteligência artificial razoável para ajudar os participantes com dicas e explicação sobre o jogo.

Esta ideia é muito parecida com os princípios do jogo *LightBot* (LightBot, 2017), onde podemos encontrar uma série de exercício interativos, onde seu objetivo é fazer um robô percorrer uma pista em busca de blocos que deverão ser acesos, utilizando de instruções montadas pelo aluno.

Uma das diferenças, é que a proposta terá como base o sistema RPG (Techopedia, 2017), que simboliza um ambiente onde temos uma missão principal que devemos cumprir para terminar o jogo, um sistema de batalha padrão e seu personagem evolui durante o jogo em termos de força, inteligência, entre outros atributos.

### **3.1 Sistema *Ingame* (dentro do jogo)**

O sistema *Ingame* simboliza como o sistema se comporta durante o jogo em seu ambiente virtual. Como a ideia do jogo é ser um auxiliar e não um substituto do professor, o sistema de interação do jogo será o seguinte:

1. A turma poderá se dividir em grupos ou individualmente. Futuramente cada grupo ou pessoa poderá escolher o nome do seu time e a logo (imagem)
2. Caso seja formado um grupo, todos devem usar o mesmo computador, sendo assim uma pessoa fica responsável pela mecânica do jogo e as outras ajudam a pensar.
3. O jogo será separado por fases, no estilo jogos planos RPG, como por exemplo o jogo *Zelda Classic* (Wikipedia, 2017) Cada fase terá um desafio diferente, podendo ser ele de lógica, ação, ou os dois.
4. Para derrotar os inimigos e avançar, é necessário resolver questões de lógica, que terão suas dificuldades aumentadas de acordo com a progressão no jogo.
5. O jogo possui início e fim, caso o aluno perca antes de chegar no final, o mesmo volta ao começo do jogo, porém ao perder, sua pontuação é gravada, podendo sempre ser superada pelo aluno.
6. Ao final do jogo, são mostradas as estatísticas de cada grupo ou aluno e quem se sagrou o vencedor.

Como incentivo, a instituição de ensino que implementa a gamificação ou o professor, pode fornecer uma recompensa para a equipe ou aluno vencedor, podendo ser algo material ou alguma coisa que indique que a equipe foi a vencedora, como medalhas ou indicação em um cartaz.

### **3.2 Tecnologias Empregadas**

Como citado anteriormente, as atuais plataformas tendem a ser visualmente agradáveis, e para isso, usaremos dois *Frameworks* (Bibliotecas de código) para a

criação do jogo, que nesse projeto trata-se do XNA (ANDRADE, 2017), que vem a ser um Framework criado pela Microsoft para o desenvolvimento de jogos para PC, XBOX e Celular. O segundo é a NovaDII, um Framework criado pelo aluno para auxílio ao desenvolvimento de jogos.

A IDE utilizada será o Visual Studio 2015 (Visual Studio, 2017), uma ferramenta desenvolvida pela Microsoft para a criação de programas, onde utilizaremos da linguagem de programação C# (C-Sharp) (OFICINADANET, 2017) para a criação do jogo. O mesmo possui licença *Community*<sup>2</sup>, tornando assim todo o desenvolvimento legal e reconhecido.

Os recursos do jogo serão adquiridos por meio de compra e através de sites que disponibilizam conteúdos para jogos gratuitos, como o site *OpenGameArt* (Open Game Art, 2017). São eles:

- **Texturas e Imagens:** Necessárias para compor a parte visual do sistema, como os personagens, o mapa, o plano de fundo, entre outros.
- **Músicas e Efeitos Sonoros:** Usados para criar a trilha sonora do jogo, fazendo-o ficar mais interessante e possibilitando melhor imersão do jogador.

A forma para implementação do jogo será através de sua instalação nos computadores da sala de aula, aos quais serão necessárias pequenas configurações básicas para o funcionamento, como por exemplo, a instalação dos componentes necessários para o funcionamento do jogo, componente da Microsoft Net Framework 4.0 e XNA 4.0, e como forma de aumentar a interatividade, os computadores devem possuir sistema de áudio, pois o jogo possuirá efeitos sonoros.

Com a aplicação do jogo, pretende-se alcançar uma boa imersão do aluno no próprio, com isso, espera-se que o mesmo consiga ter seu aprendizado acelerado, consistente e motivador, fazendo com que o aluno se interesse pela matéria de uma forma divertida e o faça buscar de forma autônoma por diferentes soluções.

#### 4. Implementação da Proposta do Jogo

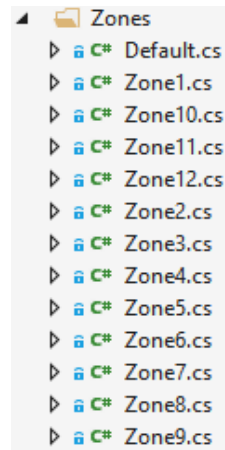
Como citado anteriormente, o jogo proposto, cujo nome é *Linn*, foi desenvolvido para fins educativos, mas mantendo aquele ambiente de um jogo de entretenimento normal. Para facilitar o desenvolvimento do jogo, o mesmo faz o uso de um padrão de projeto, que é um modelo padrão com diversos passos para se realizar um projeto, que no caso deste jogo, é o *flyweight* (Ribeiro, 2017), que foi desenvolvido para auxiliar aplicações que possuem um grande número de objeto, que em um jogo é um cenário bem comum.

O mapa do jogo foi dividido em zonas (regiões), onde cada uma é isolada das demais, e a visibilidade do jogador se limita a uma região específica. Por exemplo, se o usuário estiver na região 1, a região 2 fica invisível. Para isso cada uma delas é uma classe diferente de acordo com a Figura 2.

---

<sup>2</sup> Licença Community: é a permissão para uso da ferramenta contanto que não seja usado para fins comerciais.

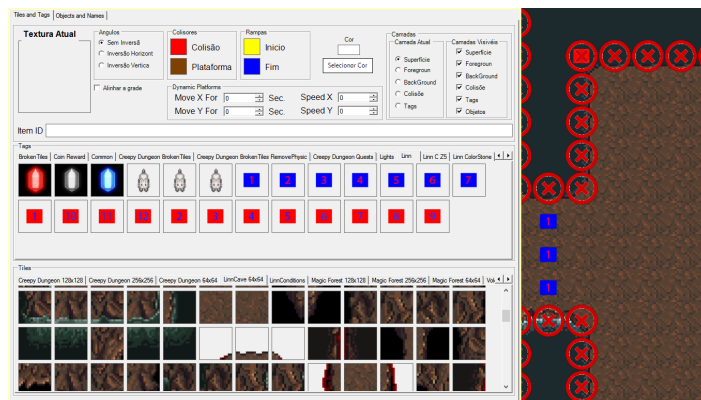




**Figura 2:** Imagem mostrando a estrutura das classes das regiões.  
**Fonte:** Do Autor

Cada região foi criada utilizando o criador de mapa chamado “Nova MapCreator” que foi construído utilizando o Framework “NovaDII” que foi desenvolvido pelo autor.

O criador de mapa possui uma interface gráfica onde o usuário monta o mapa em modo desenho, como se fosse montar um quebra cabeça. Na Figura 3, temos à esquerda as peças necessárias para montar o mapa, e do lado direito o resultado em si.



**Figura 3:** Criador de mapa.  
**Fonte:** Criador de mapa Nova Map Creator

Devido ao uso do padrão *flyweight*, existe uma classe chamada de “StaticData.cs”, que guarda as referências dos objetos criados, para que seja possível o acesso a qualquer objeto do jogo.

```
public static class StaticData
{
    public static Matrix Matrix;
    public static KeyboardState keyboardNew, keyboardOld;
    public static Boolean LightsOn { get; set; }
    public static Map.MainMap MainMap { get; set; }
```

**Figura 4:** Trecho de código da classe “StaticData”

**Fonte:** Do Autor

O trecho do código da Figura 4 exemplifica a classe “StaticData.cs”, onde por exemplo, observa-se o código “public static Map.MainMap [...]”, que seria a declaração do mapa principal, logo, através dessa variável, qualquer classe tem acesso a todo o conteúdo do mapa.

#### 4.1 Características de Jogabilidade

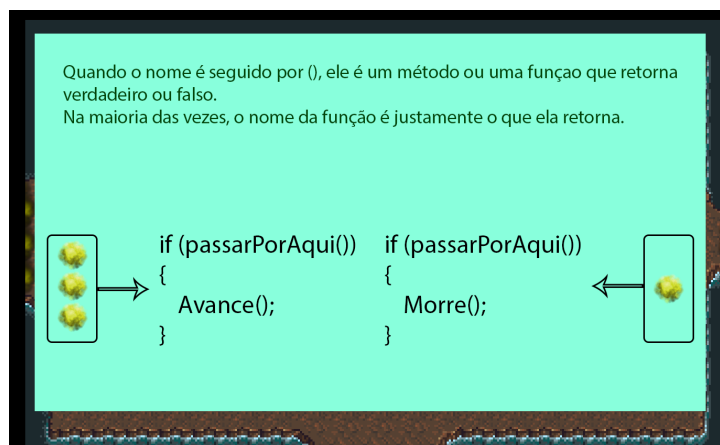
O jogo inicia em um lugar escuro, onde o personagem principal encontra-se desorientado e começa a falar consigo mesmo, assim como mostrado na Figura 5. Neste momento deve-se procurar a saída do mapa.



**Figura 5:** Exibição do primeiro mapa.

**Fonte:** Do Autor

No próximo mapa, o protagonista se depara com um gato falante, que logo diz que ele está preso em um sonho, e a única forma de acordar é chegando ao final do jogo. Para isso, deve-se resolver quebra cabeças que a sua resolução depende se o jogador conseguir interpretar alguns códigos escritos de acordo com a Figura 6, podendo ser visualizados pressionando a tecla M do teclado. Que no caso desse mapa, é uma condição “IF” simples, onde o mesmo mostra 2 caminhos, um que faz perder o jogo, e o outro que continua, e tudo isso é explicado ao jogador pelo gato.



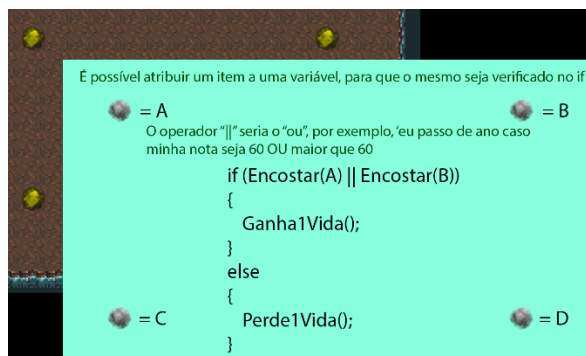
**Figura 6:** Utilização de condicionais no jogo  
**Fonte:** Do Autor

Caso o jogador passe do primeiro desafio indicado pela imagem acima, a próxima sala possui inimigos normais, como mostrado na Figura 7, como em qualquer outro jogo. Que é exatamente o diferencial deste, pois o mesmo possui elementos de ação que podem ser sanados interpretando códigos.



**Figura 7:** Inimigos e a condição de passagem do terceiro mapa  
**Fonte:** Do Autor

Já no próximo mapa, o jogo dá uma ideia sucinta do que seria uma variável, onde o mesmo atribui cada item do mapa a uma letra e as usam na condição "IF" da fórmula para passar de fase, o que pode ser visto na Figura 8.



**Figura 8:** Atribuição de letra ao elemento do mapa  
**Fonte:** Do Autor

Logo após esse mapa, no mapa seguinte, o gato aparece novamente para explicar o jogador como montar o próprio "IF", que no jogo é retratado como um

quebra cabeça que as variáveis são elementos do próprio jogo, como por exemplo, a quantidade de vidas do jogador.

Na montagem do “IF” é usado o *Reflection*, pois a função que checa os valores corretos nos lugares certos precisa saber qual o tipo daquela classe a qual seu valor se refere, como ilustra a Figura 9 a seguir:



**Figura 9:** Montagem do “IF” dentro do jogo  
**Fonte:** Do Autor

A Figura 10 abaixo representa a condição que verifica se o jogador completou corretamente a condição do “If” no mapa, a mesma recupera a lista de encaixes e de peças (*Slot* e *Element*), e caso todos os encaixes estejam preenchidos com suas respectivas peças, o retorno da função é verdadeiro.

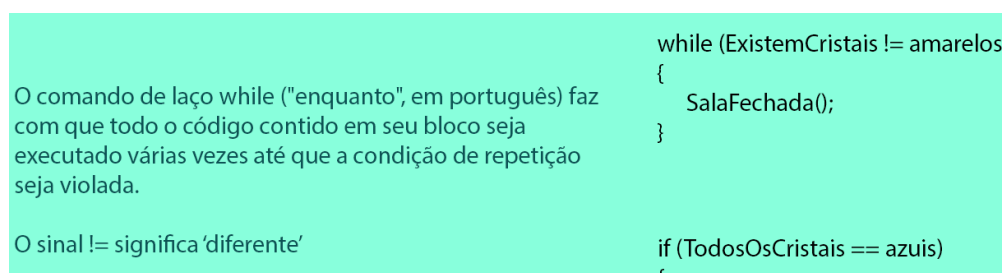
```
List<Localizables> source = GetEntitiesOfCurrentZone();
int count = source.Count(k => k.GetType() == typeof(Entities.Conditional.Slot));
for (int i = 0; i < source.Count; i++)
{
    if (source[i].GetType() == typeof(Entities.Conditional.Slot))
    {
        Entities.Conditional.Slot s = (Entities.Conditional.Slot)source[i];
        int id = s.numericId;
        for (int j = 0; j < source.Count; j++)
        {
            if (source[j].GetType() == typeof(Entities.Conditional.Element))
            {
                Entities.Conditional.Element e = (Entities.Conditional.Element)source[j];
                if (s.rectangle.Intersects(e.GetCollisionRectangle()))
                {
                    if (s.numericId == e.numericId && s.varType == e.varType)
                    count--;
                }
            }
        }
    }
}
return count == 0;
```

**Figura 10:** Trecho de código que verifica se a condição foi sanada  
**Fonte:** Do Autor.

Existe um inimigo no jogo que não pode ser derrotado com o ataque normal, somente se o mesmo for atraído para um espinho que ele morre, porém o jogador só saberá disso se entender o código mostrado ao apertar a tecla M.

Seguindo em frente, o jogador também se depara com a estrutura de laço conhecida como “WHILE”, que basicamente é explicado ao usuário quando o mesmo acessa a tela de código do jogo.

O “WHILE” é encontrado em duas telas, e seu código é mostrado de acordo com a Figura 11. Na primeira tela, o jogador somente deve interpretar o que a estrutura faz, e no segundo, assim como no mapa do “IF”, deve-se montar o próprio “WHILE” antes que os espinhos do mapa alcancem o jogador.



**Figura 11:** Primeira aparição do “WHILE no jogo  
**Fonte:** Do Autor

O último desafio do jogo é derrotar o gato, que usa como base a quantidade de vidas do jogador para atacar, o que também é mostrado na tela de códigos do jogo. Após derrotar o gato, o jogo termina.

O tamanho total que o jogo ocupa em disco ficou em aproximadamente 5 MB, e é compatível com Windows XP e superiores. Ocupa em média 20 MB de memória e menos de 2% de processador, considerando um Processador da marca Intel modelo i3 6100u. Logo, o mesmo não necessita de um computador específico para ser executado, qualquer computador acadêmico é capaz de iniciar a aplicação.

## 5. Avaliação dos Resultados Obtidos na Aplicação do Jogo Proposto

Para medir a eficácia da aplicação do jogo, foi criada uma avaliação para verificar se o aluno teve uma melhora em sua parte técnica a respeito de programação. Para isso, foram levantados os quesitos mais comuns de um jogo tradicional e um jogo educacional (MEDEIROS e SCHIMIGUEL, 2017).

Tem-se várias questões que devem ser levadas em consideração, como a imersão do jogador no ambiente do jogo, o grau de dificuldade e também questões mais pessoais, como, por exemplo, a opinião do jogador sobre o jogo.

### 5.1 Método de Avaliação

A avaliação consiste na aplicação de um formulário antes e depois do usuário executar e jogar o jogo. Inicialmente as questões foram mais relacionadas à matéria de programação em geral, possuindo mais questões fechadas do que questões abertas, exemplificadas na Figura 12.

Você está respondendo as questões \*

Antes de jogar

Depois de jogar

Qual é o comando que verifica se uma condição é verdadeira ou falsa?

WHILE

FOR

IF

**Figura 12:** Questões do início do formulário de avaliação  
**Fonte:** (TRAVNIK, 2017)

Logo após essas questões mais técnicas observadas na Figura 12, foram questionados pontos mais pessoais sobre o jogo, como mostra a Figura 13 a seguir.

O jogo te ajudou a aprender algo? Se não, porque?

Sua resposta

---

Por quanto tempo você gastou jogando?

Sua resposta

---

Qual foi sua maior dificuldade em terminar o jogo?

Sua resposta

---

**Figura 13 -** Questões abertas do formulário  
**Fonte:** (TRAVNIK, 2017)

Com isso, pode-se saber o que o jogo conseguiu passar para o usuário e quais os pontos que o mesmo pode melhorar. Os resultados esperados são bem positivos, visto que em uma primeira avaliação aplicada a uma pessoa que não tinha nenhuma noção a respeito de programação, a mesma conseguiu responder todas as questões do formulário após ter jogado.

## 5.2 Resultados Alcançados

O jogo foi aplicado a 5 pessoas, com os seguintes atributos:

1. Sem nenhum conhecimento de programação e sem pretensão de aprender.
2. Sem conhecimento de programação com vontade de aprender.
3. Com um pequeno conhecimento de lógica computacional porém sem prática na programação.
4. Com conhecimento de lógica cursando o cursos de sistemas de informação.

5. Com grande conhecimento de lógica, já trabalhando no ramo de programação.

Essa diversidade foi justamente para ter uma opinião mais variada se o jogo realmente pode ajudar pessoas que não tiveram contato com algoritmos computacionais.

Em geral os resultados foram bem satisfatórios, visto que o jogo foi aplicado a pessoas de diversas áreas. A totalidade de 100% das pessoas (3) que não sabiam responder o formulário antes do jogo, conseguiram responder pelo menos uma questão de programação e as que já sabiam (2) conseguiram responder todas, isso considerando que o tempo médio de jogo das pessoas foi de 53 minutos.

A maior dificuldade relatada pelos usuários foram as partes mecânicas do jogo, como por exemplo, derrotar um inimigo usando o ataque normal. Devido a isso, foi feito um ajuste para deixar as partes mecânicas mais simples e focar mais nas partes que usam lógica.

Nas questões abertas do formulário, uma aluna do curso respondeu na questão que pergunta se o usuário acha válido o jogo ser aplicado na sala de aula, o seguinte: "Sim, claro. O jogo correspondeu as minhas expectativas e creio que será uma importante ferramenta nas aulas."

## 6. Conclusão e Considerações Finais

Apesar de simples, o jogo passa uma ideia do que seria a interpretação dos algoritmos, pois a aplicação implementa justamente o que está escrito na tela de códigos do jogo.

Inicialmente o jogo é bem difícil para quem nunca jogou algum jogo parecido ou não tenha nenhum conhecimento de lógica computacional. Porém com persistência do usuário, o mesmo pode ter uma noção básica do que é programação de uma maneira mais dinâmica.

Os resultados preliminares foram bem satisfatórios, visto que o mesmo foi aplicado a pessoas que não tinham nenhum contato com programação. Sendo assim, o sistema não tem um requerimento formal a ser cumprido, qualquer pessoa que tenha interesse em ter uma noção de programação, pode usar esse recurso.

O jogo que foi desenvolvido nesse trabalho, ainda é muito pequeno e pouco abrangente. Nossa intenção futura é de realizar mais aplicações do mesmo em salas de aula e realizar o aprimoramento do jogo em questões por exemplo de adicionar mais desafios, estrutura de dados, tipos de dados e afins.

## REFERÊNCIAS

Andrade, K.; **Conceitos e Instalação do XNA 4.0**; Disponível em: <http://www.pontov.com.br/site/xna/53-xna/259-conceitos-e-instalacao-do-xna-40>. Acesso em 10/02/2017

Brinquedos Iterativos; **Entenda a diferença entre brinquedos educativos e pedagógicos**

Disponível em: <http://blog.abaratadizqtem.com.br/entenda-a-diferenca-entre-brinquedos-educativos-e-pedagogicos/>. Acesso em 22/06/2017

Carvalho, M. F. N.; Pereira, V. C.; FERREIRA, S. P. A. **A (DES) Motivação da Aprendizagem de Alunos de Escola Pública do Ensino Fundamental I: Quais os Fatores Envolvidos?**; Disponível em: [https://www.ufpe.br/ce/images/Graduacao\\_pedagogia/pdf/2007.2/a%20desmotiva%20da%20aprendizagem%20de%20alunos%20de%20escola.pdf](https://www.ufpe.br/ce/images/Graduacao_pedagogia/pdf/2007.2/a%20desmotiva%20da%20aprendizagem%20de%20alunos%20de%20escola.pdf). Acesso em 28/03/2017

Ferramenta **VisuAlg**; Disponível em: <http://www.apoioinformatica.inf.br/produtos/visualg>. Acesso em 24/10/2016.

Ferramenta **Scratch**; Disponível em: <https://scratch.mit.edu/about>. Acesso em 17/10/2016.

Ferramenta **RoboCode**; Disponível em: <http://robocode.sourceforge.net/>. Acesso em 27/10/2016.

Fardo, M, L. **A Gamificação Aplicada em Ambientes de Aprendizagem**. Disponível em: <http://seer.ufrgs.br/renote/article/viewFile/41629/26409>. Acesso em 06/12/2016

Ferramenta **Alice**; Disponível em: [http://alice.org/index.php?page=what\\_is\\_alice/what\\_is\\_alice](http://alice.org/index.php?page=what_is_alice/what_is_alice). Acesso em 25/10/2016.

LightBot; **LightBot - Solve Puzzles using Programming Logic**; Disponível em: <https://lightbot.com>. Acesso em 22/03/2017

Huizinga, J.; **Homo Ludens - O jogo como elemento da cultura**. São Paulo: Perspectiva, 1980. Leitura em 10/01/2017.

Instituição **Code Academy**; Disponível em: <https://www.codecademy.com/pt>. Acesso em 16/10/2016.

Livro gamification. **(Gameificação para empresas)**; Disponível em: <http://www.livrogamification.com.br/#what>. Acesso em 10/12/2016.

Melo, J. C. B., Silva, T. S. C. da Silva, Silva, A. S. C., **Ensino de Algoritmos a Nível Médio Utilizando Música e Robótica: Uma Abordagem Lúdica**; Disponível em: <http://www.lbd.dcc.ufmg.br/colecoes/wei/2011/0024.pdf>. Acesso em: 25/10/2016.

Medeiros, M. O; Schimiguel, J. **Uma Abordagem para Avaliação de Jogos Educativos: Ênfase No Ensino Fundamental**; Leitura em 03/06/2017

M. Maia. **Desinteresse afasta jovem da escola**. Disponível em: <http://oglobo.globo.com/sociedade/educacao/desinteresse-afasta-jovem-da-escola-4204782>. Acesso em 20/10/2016.

Navarro, G; **Gamificação: a transformação do conceito do termo jogo no contexto da pós-modernidade**. Disponível em: [https://disciplinas.stoa.usp.br/pluginfile.php/125459/mod\\_resource/content/1/gamificacao.pdf](https://disciplinas.stoa.usp.br/pluginfile.php/125459/mod_resource/content/1/gamificacao.pdf). Acesso em 10/12/2016.



Oficinadanet; **C# (CSharp) o que é esta linguagem?**; Disponível em: [https://www.oficinadanet.com.br/artigo/526/c\\_sharp\\_csharp\\_o\\_que\\_e\\_esta\\_linguagem](https://www.oficinadanet.com.br/artigo/526/c_sharp_csharp_o_que_e_esta_linguagem). Acesso em 22/06/2017

Open Game Art; **Site de Recursos gratuitos para jogos**; Disponível em: <https://opengameart.org/>. Acesso em 08/06/2017

Pruss, E. M. **Algoritmos: A Lógica da Automatização de Tarefas**. Revista das Faculdades Santa Cruz, v. 7, n. 1, Disponível em: <http://www.santacruz.br/v3/revistaacademica/12/cap3.pdf>. Acesso em 24/10/2016.

Pezzini, C. C.; Szymanski, M. L. S. **Falta de Desejo de Aprender Causas e Consequências**; Disponível em: <http://www.diaadiaeducacao.pr.gov.br/portals/pde/arquivos/853-2.pdf>. Acesso em 29/02/2017

Revista Devmedia; **Reflection**. Disponível em: <http://www.devmedia.com.br/reflection-artigo-net-magazine-84/20409>. Acesso em 01/05/2017

Ribeiro. A. A.; **Padrão de projeto Flyweight**; Disponível em: [http://www.dpi.ufv.br/projetos/apri/?page\\_id=722](http://www.dpi.ufv.br/projetos/apri/?page_id=722). Acesso em 01/05/2017

Silva, L.B.; Tavares, J.S.; et Al. **QuizQuím - Química Divertida**. Disponível em: [http://www.abq.org.br/cbq/trabalhos\\_aceitos\\_detalhes,10167.html](http://www.abq.org.br/cbq/trabalhos_aceitos_detalhes,10167.html). Acesso em 12/10/2016.

Silva, V., Souza, A. Moraes, D. **Pensamento Computacional no Ensino de Computação em Escolas: Um relato de Experiência de Estágio em Licenciatura em Computação em Escolas Públicas**; Disponível em: [http://ceur-ws.org/Vol-1667/CtrlE\\_2016\\_AC\\_paper\\_55.pdf](http://ceur-ws.org/Vol-1667/CtrlE_2016_AC_paper_55.pdf). Acesso em 26/10/2016.

Travnik, G; Formulário feito para o jogo Linn Disponível em: <https://goo.gl/forms/tfZsvLQkWneBnPbm2>. Acesso em 18/05/2017

Techopedia; **Role-Playing Game (RPG)**. Disponível em: <https://www.techopedia.com/definition/27052/role-playing-game-rpg>. Acesso em 28/03/2017

Visual Studio; **IDE do Visual Studio para Windows e Mac**; Disponível em: <https://www.visualstudio.com/pt-br>. Acesso em 15/03/2017

Viana. G; **A tecnologia invade a sala de aula**; Disponível em: <http://www.techtudo.com.br/noticias/noticia/2012/10/tecnologia-invade-sala-de-aula-veja-recursos-que-auxiliam-o-ensino.html>. Acesso em 20/10/2016.

Wikipedia. **O Jogo The Legend of Zelda Classic**; Disponível em: [https://pt.wikipedia.org/wiki/The\\_Legend\\_of\\_Zelda\\_\(jogo\\_eletr%C3%B4nico\)](https://pt.wikipedia.org/wiki/The_Legend_of_Zelda_(jogo_eletr%C3%B4nico)). Acesso em 11/03/2017