

# Desenvolvimento de um Jogo Educacional Digital para auxílio à Alfabetização utilizando Redes Neurais

Maik Basso<sup>1</sup>, Silvana Kliszcz<sup>1</sup>, Fábio José Parreira<sup>2</sup>, Sidnei Renato Silveira<sup>2</sup>

<sup>1</sup>Curso de Bacharelado em Sistemas de Informação – <sup>2</sup>Departamento de Tecnologia da Informação - Universidade Federal de Santa Maria (UFSM) – Centro de Educação Superior Norte RS (CESNORS) – Frederico Westphalen – RS – Brasil

maik@maikbasso.com.br, silvana@websetbrasil.com,  
fabiojparreira@gmail.com, sidneirenato.silveira@gmail.com

**Resumo.** *Este artigo apresenta o desenvolvimento de um jogo educacional digital – “Aprendendo com o Zag”, para ser aplicado como ferramenta de auxílio aos processos de ensino e de aprendizagem voltados à alfabetização. O jogo utiliza Redes Neurais Artificiais para realizar a verificação da escrita dos alunos em fase de alfabetização, além de proporcionar um ambiente lúdico que estimula o aprendizado. O jogo foi desenvolvido para alunos do ensino fundamental, na faixa etária entre cinco e sete anos de idade, podendo ser utilizado por meio de um browser web ou dispositivos móveis e dispõe de algumas fases para que a criança possa aprender a escrever e identificar corretamente as vogais do alfabeto.*

**Palavras chaves:** *Jogos Educacionais Digitais, Processo de Alfabetização, Redes Neurais Artificiais.*

**Abstract.** *This paper presents the development of a digital educational game - "Aprendendo com o Zag", to be applied as a tool for teaching and learning processes focused on literacy. The game use Artificial Neural Networks to carry out the writing verification of students in literacy phase, and provide a playful environment that encourages learning. The game was developed for elementary school students, aged between five and seven years old, it can be used through a web browser or mobile devices and have some phases so that the child can learn to write and correctly identify the vowels of the alphabet.*

**Keywords:** *Digital Educational Games, Literacy Process, Artificial Neural Networks.*

## 1. Introdução

Este artigo apresenta o desenvolvimento de um protótipo de jogo educacional digital – *Aprendendo com o Zag*, mais especificamente na área de alfabetização, voltado para crianças de cinco a sete anos de idade, com o intuito de apoiar os processos de ensino e de aprendizagem da escrita. A aplicação deste jogo pode tornar os processos de ensino e de aprendizagem mais agradáveis e menos cansativos, estimulando a aprendizagem das crianças por meio de um mundo lúdico e divertido.

A existência de um grupo de pesquisa na UFSM/Frederico Westphalen, que atua na linha de pesquisa de Inteligência Artificial e Tecnologia Educacional (IATE), também é um dos motivadores para o desenvolvimento deste trabalho. Entre outras ferramentas, este grupo atua no desenvolvimento de jogos educacionais digitais, empregando técnicas de Inteligência Artificial (IA). A aplicação de jogos educacionais digitais pode cativar a atenção das crianças, permitindo com que professores e alunos possuam uma ferramenta que estimule os processos de ensino e de aprendizagem.

Acredita-se que, com o emprego de um jogo educacional digital para apoiar o processo de alfabetização, os alunos se sentirão mais motivados em aprender, não ficando limitados ao uso de papel e de lápis. O jogo implementado também pode auxiliar o professor a desenvolver uma aula mais dinâmica e criativa, promovendo o aumento da atenção dos alunos ao foco central da aula. Seguindo esta linha de pensamento, foram utilizadas imagens de fácil associação e metodologias que facilitam e incentivam os alunos a manterem sua atenção focada no aprendizado da escrita, por meio de uma interface simples e com um jogo dividido em fases não muito complexas. O jogo dispõe de algumas fases para que a criança possa aprender a escrever e identificar corretamente as vogais do alfabeto. A aplicação da IA neste jogo foi realizada por meio de Redes Neurais Artificiais (RNAs) em uma das fases do jogo, na qual o aluno deve escrever e as letras são reconhecidas por meio de RNAs.

Neste contexto, o presente artigo se encontra dividido da seguinte forma: a seção 2 apresenta um breve referencial teórico envolvendo as áreas de estudo relacionadas ao desenvolvimento do protótipo do jogo educacional proposto. A seção 3 apresenta alguns trabalhos relacionados, onde são descritos e comparados trabalhos que envolvem o desenvolvimento de jogos educacionais digitais. A seção 4 apresenta a solução implementada, sendo analisadas as tecnologias e ferramentas usadas para a implantação do protótipo do jogo, bem como a implementação e validação do mesmo. Encerrando o artigo são apresentadas as considerações finais, destacando os resultados obtidos, bem como as referências utilizadas.

## **2. Referencial Teórico**

Esta seção apresenta um referencial teórico sobre as áreas envolvidas no desenvolvimento deste trabalho, destacando conceitos referentes a jogos educacionais, aplicação de IA na Educação, Redes Neurais Artificiais e o processo de alfabetização.

### **2.1. Alfabetização**

O processo de alfabetização, tanto em escolas públicas como em escolas particulares, possui várias formas de abordagem, que são constantemente discutidas, visando garantir a qualidade de ensino aos alunos. Este fato faz parte da realidade educacional do país. Os problemas ou situações que envolvem este processo podem ser contornados com o apoio de equipes de profissionais e educadores interessados, que estejam atentos a toda sistemática que envolve o processo de alfabetização em uma escola. Com isso, é possível minimizar os problemas relacionados a este contexto em uma instituição escolar, construindo uma escola participativa, crítica e que garanta uma eficácia mais elevada na alfabetização dos alunos (PEREIRA et. al., 2013).

Porém, outro ponto importante a ser analisado durante o processo de alfabetização seria o fato de que o mesmo é um processo único, que será vivenciado de forma diferente por cada aluno. Cada aluno possui formas particulares de aprendizagem, o que ressalta a necessidade de que os professores precisam estar sempre inovando, atentos, fazendo uso dos diferentes meios de ensino, visando motivar os alunos a aprender e que, acima de tudo, consigam a sua atenção e dedicação, no decorrer do processo de aprendizagem. Com isso, observa-se que, nesta etapa de aprendizagem deve ser trabalhada não somente a questão da decodificação de sons e letras, mas atingir a questão da escrita de forma ampla e relacionada ao seu uso social (SILVA, 2014).

Os professores precisam tornar suas aulas motivadoras, para assim despertar o interesse dos alunos tanto quanto o gosto pela alfabetização, pela leitura, seguida pela escrita, fazendo com que este se interesse e acompanhe a interpretação, evoluindo de forma prazerosa e eficiente, passando a ter autonomia em agir, inventar, descobrir e ter criatividade. Para isso os professores podem usar de ferramentas para tornar suas aulas interessantes e dinâmicas, sendo que uma destas ferramentas podem ser os jogos educacionais, assunto que é abordado neste trabalho (SILVA, 2014).

## **2.2. Jogos educacionais**

Os jogos educacionais despertam o interesse que as crianças têm em brincar, jogar e, aproveitando essa situação propiciam, por meio de recursos lúdicos, um ambiente agradável para o aprendizado. Como um jogo pode fazer com que o aluno estude e aprenda se divertindo, sem causar estresse, o mesmo pode compreender mais facilmente os conteúdos que estão sendo abordados. Os jogos também ajudam no desenvolvimento da atenção, disciplina, autocontrole, respeito a regras e habilidades perceptivas e motoras. Também podem estimular o trabalho em grupo, já que podem ser jogados de forma colaborativa entre os estudantes. Porém, mesmo com todos estes benefícios proporcionados pelos jogos, ainda cabe ao educador mediar, orientar e problematizar o processo de aprendizagem. Cabe também ao educador avaliar o nível de desenvolvimento dos alunos, diagnosticando possibilidades e necessidades individuais, para conduzir atividades adequadas para cada estudante em particular (SILVEIRA et. al., 2012).

Como já abordado anteriormente, os jogos possuem a vantagem de se tornar agentes de socialização uma vez que aproximam os alunos jogadores, gerando competitividade ou cooperativismo, tanto em ambientes virtuais ou no próprio ambiente físico de uma escola ou universidade. Outra vantagem é que os alunos podem compartilhar suas informações e experiências, expondo suas dificuldades relacionadas aos jogos, gerando um ambiente de ajuda mútua, resultando em um contexto de aprendizagem distribuída (SAVI; ULBRICHT, 2008).

### **2.2.1. Jogos educacionais digitais**

Como consequência do avanço tecnológico cada vez mais constante e em grande escala de atualização, a sociedade atual vive na chamada "era da informação" e, como resultado disso, a experiência educacional deve acompanhar essas etapas para ser diversificada e oferecer multiplicidade de tarefas. Os alunos precisam aprender de forma simples e conseguir dominar

o processo de aprendizagem para que seja possível formar suas competências, e não mais ficar restritos à absorção de conteúdo sem colocar em pauta suas ideias. Tem-se a necessidade de que o processo educativo seja mais dinâmico e desafiador, visando o desenvolvimento de habilidades para a obtenção e utilização das informações (MORATORI, 2003).

Com o acesso fácil a dispositivos como computadores, celulares, *tablets* entre outros, se torna viável a utilização de jogos educacionais digitais, já que os mesmos podem unir a curiosidade pela tecnologia e se tornar motivadores para o desafio, a fantasia e a curiosidade. Os jogos digitais podem fornecer alguns resultados bastante interessantes e que são tão importantes quanto os previamente determinados. Esses jogos fornecem a oportunidade para que os alunos descubram, inovem e construam suas soluções, criem suas capacidades de forma a estar sempre conectados com as últimas novidades e atualizações do mundo. Além disso, oferecem a oportunidade de despertar o seu raciocínio lógico, suas habilidades de resolução de problemas de forma que sejam mais interessantes do que seriam expostos em um exercício ou jogo comum. Um jogo computacional, mesmo que seja simples, pode ensinar várias habilidades e conceitos, de maneira sofisticada, interativa e inovadora para que os alunos não cansem rapidamente da mesmice dos jogos não computacionais (SAVI; ULBRICHT, 2008).

Segundo esse contexto, os jogos de computador educacionais, ou jogos educacionais digitais, devem explorar, de forma mais completa possível, os processos de ensino e de aprendizagem. Mesmo tendo-se em vista que os jogos de maior sucesso comercial atualmente são os que combinam violência com efeitos visuais sofisticados, um ponto positivo nesta área é a possibilidade de se combinar entretenimento com conteúdo educacional. Uma das maiores vantagens desta categoria de jogos é integrar a tecnologia ao desenvolvimento do aprendizado infantil, com o uso equilibrado e coerente dos recursos disponíveis (RIEDER et. al., 2004).

Outro ponto a ser levado em consideração, quando se pretende utilizar um jogo educacional digital dentro dos processos de ensino e de aprendizagem, é que, além de ser considerado o conteúdo, deve-se considerar a maneira como o jogo se apresenta de forma relacionada à faixa etária que irá compor o público alvo da aplicação. É importante um plano de jogo, ou seja, as metas a serem alcançadas, como: memória, orientação temporal e espacial, coordenação motora, percepção auditiva, percepção visual, raciocínio lógico-matemático, expressão linguística, planejamento e organização (RIEDER et. al., 2004).

### **2.3. Inteligência artificial**

A Inteligência Artificial (IA) é uma área da ciência computacional que surgiu por volta da década de cinquenta, logo após a segunda guerra mundial. Não existe apenas uma única definição para a área de IA, mas generalizando os conceitos, pode-se dizer que a inteligência artificial caracteriza-se por dispor de racionalidade, ou seja, é capaz de fazer com que o computador aprenda com os dados que dispõe. Em outras palavras, armazenar o conhecimento e executar as atividades corretamente a partir do conhecimento armazenado (RUSSELL; NORVIG, 2004).

Outra visão um pouco diferente é estabelecida por Luger (2004), que define a IA como sendo o ramo da ciência da computação que se ocupa com a automação do

comportamento inteligente. O autor ainda afirma que a área é fundamentada por alguns princípios que se baseiam nas estruturas de dados usadas na representação do conhecimento, tais como os algoritmos usados para aplicar o conhecimento às linguagens de programação propriamente ditas.

O nome Inteligência Artificial foi sugerido por John McCarthy em um seminário que aconteceu em *Dartmouth* no verão do ano de 1956, sendo que o mesmo surgiu como substituto para o termo usado até então: “racionalidade computacional” (RUSSELL; NORVIG, 2004).

Em 1987 a IA se torna de fato uma ciência, após adotar com firmeza o método científico. Para serem aceitas, as hipóteses da IA eram submetidas a rigorosos experimentos empíricos, sendo que os resultados eram avaliados estatisticamente levando em consideração sua relevância. Após esse fato, os estudos de IA avançaram mais rapidamente, e, o processo de compreensão da inteligência caminha lado a lado com os sistemas reais (RUSSELL; NORVIG, 2004).

Algumas das preocupações dos pesquisadores de IA são como representar o conhecimento e como buscar a informação. A primeira preocupação refere-se a como capturar o conhecimento de forma adequada para que possa ser manipulado pelo computador, e a segunda refere-se a uma técnica de solução que explore os problemas em todos os seus estágios possíveis para que se possa chegar a uma solução (LUGER, 2004).

Existem diversas arquiteturas dentro da área de IA para que se possam resolver diferentes tipos de problemas. Uma arquitetura que se destaca nesse sentido é a arquitetura neural, por conseguir capturar o conhecimento por meio de um grande número de unidades distribuídas em uma rede, unidades estas chamadas de neurônios, e que possibilitam o reconhecimento de dados incompletos e ruidosos, chegando assim à solução do problema de forma mais eficiente. Esta arquitetura, que envolve o desenvolvimento de RNAs, é um dos tópicos deste trabalho, já que o jogo apresentado neste artigo foi implementado empregando-se esta técnica de IA (LUGER, 2004).

### **2.3.1. Inteligência artificial aplicada ao desenvolvimento de jogos**

A utilização de técnicas de IA no desenvolvimento de jogos digitais está se tornando cada vez mais comum, devido ao aumento da complexidade dos jogos e de seus recursos cada vez mais bem elaborados e realistas. A aplicação dessas técnicas e/ou algoritmos em jogos pode ser dividida em três grandes blocos relacionados com a sua respectiva área de atuação, sendo: movimento, tomada de decisão e estratégia de jogo (MILLINGTON; FUNGE, 2009).

Os algoritmos de movimento implicam no cálculo do percurso, ou seja, ajudam a traçar rotas e a evitar obstáculos enfrentados por elementos de um jogo. O grupo de algoritmos de estratégias de jogo é um grupo centralizador, que ajuda um conjunto ou equipe de elementos e/ou personagens de um jogo a tomar um rumo em comum. Como exemplo dessa categoria de algoritmo pode-se citar uma equipe de soldados: cada soldado toma suas próprias decisões e traça seus próprios movimentos, mas a equipe possui um objetivo em comum (MILLINGTON; FUNGE, 2009).

O último grupo de algoritmos de IA utilizada em jogos são os de tomada de decisão. Cada elemento do jogo possui normalmente um conjunto de estados ou atividades possíveis, seja manter-se parado, movimentar-se, aumentar, reduzir, entre outras. Para que seja possível uma tomada de decisão é preciso analisar o contexto e executar uma determinada ação. Sendo assim, esse conjunto de algoritmos implica na alteração de estados dos elementos, causando repercussões visuais ao jogador (MILLINGTON; FUNGE, 2009).

A aplicação de um algoritmo de IA no desenvolvimento de jogos deve ser estudada com cautela, já que é um processo demasiadamente complicado e que gasta demasiados recursos do computador ou dispositivo móvel. Na grande maioria das vezes, esses algoritmos mais complexos não trazem grandes vantagens à jogabilidade e nem ao desempenho do jogo (MILLINGTON; FUNGE, 2009).

Variadas técnicas de IA podem ser aplicadas no desenvolvimento de jogos, isoladamente ou de forma combinada, tais como algoritmos genéticos e sistemas multiagentes, entre outras (SILVEIRA; BARONE, 1998). Neste trabalho utilizou-se a técnica de RNAs, que serão apresentadas na próxima seção.

### **2.3.2. Redes Neurais Artificiais**

As Redes Neurais Artificiais (RNAs) são algoritmos utilizados no processamento de informação, que seguem como base para a sua construção o modelo de sistema neural biológico humano. Essas redes são sistemas que se apresentam de forma paralela e distribuída, e são formados por unidades de processamento mais simples, chamados neurônios, que calculam determinadas funções, geralmente não lineares. Esses neurônios situam-se em camadas, sendo interligadas por conexões, as quais são associadas a pesos. Estes pesos armazenam o conhecimento representado na rede, servindo para fazer um balanço entre as entradas recebidas por unidade constituinte da rede (HAYKIN, 2001).

Uma RNA possui a capacidade de aprender por meio de exemplos, que são conhecidos por conjuntos de treinamento, sendo estes dados aplicados durante a fase de treinamento da rede. Mediante a fase de treinamento, a rede armazena um padrão de aprendizado capaz de generalizar um determinado problema. Finalizando a aplicação da RNA, passa-se por uma etapa de teste, cuja mesma utiliza dados dos pesos armazenados na etapa de treinamento, e então cruza determinados dados produzindo respostas satisfatórias para dados ainda desconhecidos, porém que fazem parte de um mesmo problema para o qual a rede foi treinada (HAYKIN, 2001).

Além de entender o funcionamento de uma RNA, também é importante estudar sua arquitetura, que é o que determina a capacidade de processamento da rede. Escolher corretamente o número de conexões pode ser um passo decisivo para que o processo de treinamento seja bem sucedido. As redes podem ser classificadas em recorrentes ou não recorrentes (VELLASCO, 2007).

As RNAs recorrentes, ou redes do tipo *feedback*, possuem realimentação das saídas para as entradas, tendo como saída os valores determinados por entradas atuais e pelas saídas anteriores. Sua estrutura não é obrigatoriamente organizada em camadas; essas redes

também podem apresentar interligações entre neurônios da mesma camada e entre camadas não subsequentes (VELLASCO, 2007).

As redes não recorrentes, ou do inglês *feedforward*, são aquelas que não possuem realimentação de suas saídas para as suas entradas, por isso, são conhecidas como “sem memória”. Esse tipo de rede pode ser de camada única ou pode ser formada por múltiplas camadas. Essa arquitetura de redes em camadas é formada por um conjunto de neurônios de entrada, uma camada de saída e uma ou mais camadas intermediárias ou ocultas. Nas redes não recorrentes, não existem conexões ligando um neurônio de uma camada a outro neurônio de uma camada anterior nem a um neurônio de uma mesma camada. Esse tipo de rede *feedforward* é o mais utilizado atualmente, chamando a atenção para a distinção para as suas saídas que são sinal funcional e sinal de erro (VELLASCO, 2007).

Os sinais funcionais são sinais de entrada que se propagam para frente (neurônio a neurônio) através das camadas da rede e terminam na saída da rede como um sinal de saída. São chamados assim porque em cada neurônio da rede pelo qual o sinal passa, esse sinal é calculado como uma função das entradas pelos pesos associados aquele neurônio. Já os sinais de erro originam-se nos neurônios de saída e se propagam para trás (camada por camada) através da rede (VELLASCO, 2007).

Após os sinais de entrada terem se propagado por todas as camadas da rede, chegamos ao final de uma iteração, onde ocorre a atualização/ajuste dos pesos, que contém o conhecimento da rede. Segundo Tissot (TISSOT et. al., 2012), quando se trabalha com os pesos a atenção deve ser voltada para a inicialização e ajuste dos mesmos, sendo esse um processo fundamental para a convergência e para evitar *overfitting* (excesso de ajuste nos pesos), problema que geralmente acontece em padrões de treinamento. Ainda com relação ao ajuste dos valores dos pesos na rede, existem vários algoritmos já consagrados, tais como o *perceptron*, que será descrito na próxima seção.

### **2.3.3. Algoritmo *Perceptron***

O *perceptron* foi desenvolvido por Frank Rosenblatt por volta do ano de 1958. Este algoritmo usa uma forma simples de aprendizado supervisionado, baseando-se na modificação dos pesos de modo a reduzir a taxa de erro da rede neural. O *perceptron* é composto pelas camadas de entrada e de saída, possuindo ainda um componente conhecido como nível de ativação. A camada de entrada e o nível de ativação do *perceptron* trabalham com os valores de -1 representando a não ocorrência do fator analisado e de +1 quando há a ocorrência do padrão a ser analisado (LUGER, 2004).

O conhecimento gerado pelo algoritmo é armazenado em pesos, que são representados por conjuntos numéricos do tipo real. Os níveis de ativação do *perceptron* são obtidos pela soma dos valores ponderados das entradas. Depois de obtidos estes valores, os mesmos passam por uma função limiar que determina o valor de sua saída, que quando ficar próximo de +1 representa forte ocorrência de acerto, ou pelo contrário -1, que representa baixa ocorrência de acerto (LUGER, 2004).

Este algoritmo trabalha sobre a premissa de que os padrões de entrada sejam necessariamente padrões lineares separáveis, ou seja, conjuntos que possam ser separados por uma ou mais características (HAYKIN, 2001).

A rede neural *perceptron* está classificada como sendo uma rede *feedforward*, cujo fluxo de informações em sua estrutura reside sempre no sentido da camada de entrada para a camada de saída, não existindo qualquer tipo de realimentação dos valores produzidos para a sua camada de entrada (SILVA et. al., 2010).

Considerando-se a arquitetura de uma rede neural do tipo *perceptron*, como mencionado anteriormente, devido as suas características estruturais, as funções de ativação utilizadas são as funções degrau, ou seja, possuem apenas duas possibilidades de saída (SILVA et. al., 2010). Na implementação da RNA aplicada no jogo apresentado neste trabalho, utilizou-se uma função degrau que resulta em saídas -1 ou +1 sendo que -1 representa que a letra não foi identificada pela RNA e +1 indica que a letra foi identificada pela RNA.

### **3. Estado da Arte**

Nesta seção serão apresentados alguns trabalhos correlacionados ao apresentado neste artigo, bem como a comparação entre estes jogos e o protótipo do presente artigo.

#### **3.1. Daphnia World**

Bispo (et. al., 2012) apresenta um jogo educacional digital criado com o objetivo de ensinar o ciclo de vida e o uso de micro crustáceos do gênero *Daphnia*, auxiliando no aprendizado de conceitos de toxicologia ambiental para os alunos de Saneamento Ambiental da Faculdade de Tecnologia da UNICAMP.

O jogo foi criado em HTML 5 (*HyperText Markup Language*), juntamente com a linguagem de programação *Javascript*, e está disponível na *web*, com a possibilidade de ser jogado diretamente por meio de um navegador (*browser*). O processo de desenvolvimento do protótipo do jogo foi dividido em duas partes: *engine* que é uma biblioteca com elementos e funcionalidades que auxiliam no desenvolvimento de jogos, porque controla o tempo e o espaço no jogo; e a interface jogo, que apresenta os elementos específicos do *Daphnia World* (BISPO et. al., 2012).

A Figura 1 apresenta a interface do jogo, que foi concebida por meio de um processo de edição gráfica onde foram utilizados dois softwares livres, o *Inkscape* e o *Gimp*, para a edição e criação das imagens e vetores (BISPO et. al., 2012).



Figura 1. Interface do jogo Daphnia World. Fonte: (Bispo et. al., 2012).

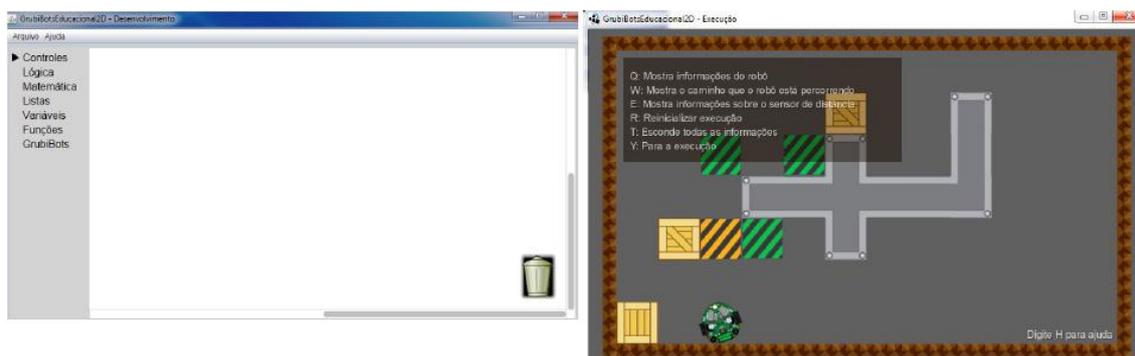
Ainda segundo os autores do projeto, a primeira versão do jogo atingiu os objetivos planejados. Os resultados confirmam a possibilidade de se desenvolver jogos educacionais utilizando HTML5, uma plataforma livre, aberta e padronizada internacionalmente. O jogo foi inicialmente concebido para ser usado com jovens em idade de ingresso em universidades, para ser uma ferramenta na apresentação do conceito de ecotoxicologia. Entretanto, foram feitos outros testes com crianças em idade pré-escolar que também apresentaram resultados promissores: com um uso bastante pontual, houve um claro interesse das crianças por conceitos como poluição, o que pode ser um uso interessante na formação dos valores dessas crianças. Esta nova possibilidade de uso traz consigo novos desafios para a continuidade do projeto (BISPO et. al., 2012).

### 3.2. GrubiBots Educacional

Oliveira (et. al., 2014) propõe o desenvolvimento de um jogo educacional digital para o ensino de algoritmos na educação básica, utilizado como ferramenta de apoio ao estudo de robótica. O seu protótipo de jogo possui, como diretrizes principais, o desenvolvimento de habilidades lógicas computacionais pelos alunos e a difusão do ensino de algoritmos de forma lúdica.

O projeto foi desenvolvido em conjunto pelo Departamento de Ciência da Computação e pelo Departamento de Educação da Universidade Federal de Lavras, e tem o intuito de resolver problemas reais das diversas áreas do conhecimento, através do desenvolvimento de modelos algorítmicos aplicados à robótica (OLIVEIRA et. al., 2014).

A Figura 2 demonstra as duas telas principais do jogo: a primeira, do lado esquerdo, demonstra o painel de controle do jogo onde se tem a opção de construir os algoritmos, e do lado direito, o ambiente de execução, onde as propriedades do algoritmo refletem-se nas ações do robô (OLIVEIRA et. al., 2014).



**Figura 2. Painel de controle e de execução do jogo GrubiBots. Fonte: (Oliveira et. al., 2014).**

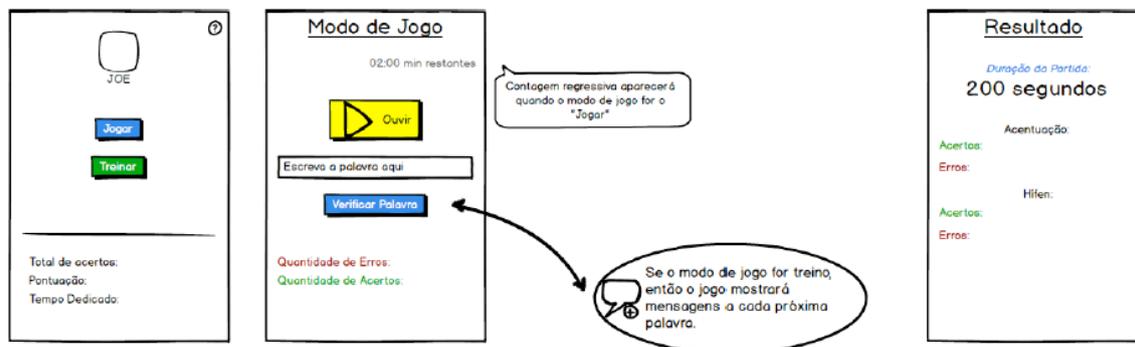
Segundo os autores do projeto, o jogo apresenta um método de programação inovador em relação aos tradicionais métodos de ensino e aprendizagem desta área, uma vez que, tal ferramenta possibilita a construção de algoritmos a partir da interconexão entre blocos de programação formados por instruções e estruturas de programação representadas por ícones, que representa a área de desenvolvimento do programa (OLIVEIRA et. al., 2014).

O jogo foi desenvolvido com a linguagem de programação *Java*, e está disponível atualmente para o ambiente *desktop*. Além disso, é importante ressaltar que o jogo destaca-se por não necessitar do desenvolvimento de códigos de programação e ainda pode ser utilizado para resolver problemas de programação de diversas áreas de conhecimento. (OLIVEIRA et. al., 2014).

### 3.3. JOE: Jogo Ortográfico Educacional

Segundo Paschoal (et. al., 2014), o JOE foi desenvolvido para possibilitar o aprendizado de conceitos importantes do novo acordo ortográfico da língua portuguesa, em um ambiente que busca estimular o aluno a continuar dedicado em seus estudos. O projeto é uma ferramenta de prática e treino que exercita as regras ortográficas necessárias para uma escrita adequada à norma padrão da língua portuguesa, verificando, a partir de uma avaliação, onde estão ocorrendo os erros, reforçando, assim, o aprendizado.

A Figura 3 apresenta o projeto da interface do jogo, demonstrando, consecutivamente, a tela principal do jogo, a tela do jogo em execução propriamente dita e, por fim, a tela que exhibe os resultados obtidos na partida.



**Figura 3. Interface do jogo JOE. Fonte: (Paschoal et. al., 2014).**

O jogo foi desenvolvido para a plataforma de dispositivos móveis *Android* e possui três telas principais. Foi construído usando a ferramenta *Android Studio*, desenvolvida pela *Google*, utilizando a linguagem de programação *Java*. As informações do jogo como as palavras e as mensagens estão armazenadas utilizando o banco de dados *SQLite* (PASCHOAL et. al., 2014).

O jogo está disponível por meio da *Play Store* do *Google*, de forma gratuita. O jogo incentiva o aprendizado de uma forma agradável, desafiando o jogador a acertar as palavras e, ao mesmo tempo, leva ao aprendizado de forma divertida e descontraída.

### 3.4. Estudo Comparativo

Por meio da análise e apresentação das características dos trabalhos correlacionados estudados, elaborou-se um quadro comparativo (Quadro 1) entre eles e o protótipo de jogo apresentado neste trabalho, com a intenção de visualizar os pontos fortes e limitações de cada trabalho, bem como suas características em comum.

**Quadro 1. Comparativo entre protótipos de jogos**

<b>Características Analisadas</b>	<i>Daphnia World</i>	<i>GrubiBots Educacional</i>	<b>JOE: Jogo Ortográfico educacional</b>	<b>Solução Implementada: Aprendendo com o Zag</b>
Faixa etária	Jovens com idade de ingresso em universidades	Estudantes de ensino básico	Pessoas interessadas em aprender o novo acordo ortográfico	Crianças de 5 a 7 anos de idade
Plataformas	<i>Web</i>	<i>Java (desktop)</i>	<i>Android</i>	<i>Android, Windows, Linux, Firefox OS</i>
Disponibilidade	<i>Web</i>	---	<i>Google Play</i>	<i>Web, Google Play, Firefox Marketplace</i>
Utilização de técnicas de IA	NÃO	NÃO	NÃO	SIM
Baseado em uma estória	NÃO	NÃO	NÃO	SIM
Linguagens e recursos de implementação	HTML5, Javascript, CSS	<i>Java</i>	<i>Java, SQLite</i>	HTML 5, CSS3, Javascript

Como visto no Quadro 1, o protótipo de jogo educacional digital apresentado neste trabalho assemelha-se a algumas características de outros trabalhos desenvolvidos, porém destaca-se por ser o único a estar disponível na maioria das plataformas (plataformas de dispositivos móveis), podendo também ser acessado diretamente via *web*, por meio de um *browser*.

Outro ponto bastante interessante é que o projeto apresentado neste artigo é o único, entre os projetos estudados, que utiliza técnicas de IA para analisar as respostas fornecidas pelos jogadores, no caso, as letras que serão escritas pelos jogadores (alunos em processo de alfabetização). A utilização de técnicas de IA no desenvolvimento do jogo educacional digital aqui apresentado mostra que é possível utilizar tais técnicas mesmo em jogos mais simples, diferentemente dos jogos apresentados nesta seção.

#### **4. Solução Implementada**

O jogo *Aprendendo com o Zag* foi desenvolvido com o intuito de auxiliar alunos do ensino fundamental, com idade entre cinco e sete anos, no seu processo de alfabetização. O projeto envolveu o desenvolvimento de um jogo educacional digital codificado utilizando tecnologias *web* tais como HTML5, CSS3 e *Javascript*. O jogo dispõe de algumas fases para que a criança possa aprender a escrever e identificar corretamente as vogais do alfabeto. Acredita-se que este jogo possa auxiliar o professor durante os processos de ensino e de aprendizagem envolvendo a alfabetização.

No decorrer do desenvolvimento do projeto foram consideradas as opiniões de professores convidados a participar da validação do jogo, além de alunos que foram convidados a realizar testes no mesmo, de forma a ajustar o jogo e torná-lo menos cansativo e repetitivo.

Todo o processo de desenvolvimento do protótipo foi realizado com base em estudos de tecnologias atuais, trabalhos relacionados na área e acompanhamento de professores do ensino fundamental, que atuam no processo de alfabetização, pretendendo-se alcançar resultados satisfatórios no interesse ao aprendizado, no correto emprego da tecnologia entre outras.

##### **4.1. Tecnologias Empregadas na Implementação do Protótipo**

Como mencionado anteriormente, o projeto foi desenvolvido utilizando tecnologias *web* de última geração, que disponibilizam os recursos necessários para a implementação do protótipo de jogo educacional digital *Aprendendo com o Zag*. Definiu-se a utilização do HTML5 por conter diversos recursos essenciais que possibilitam a construção de jogos digitais. O *canvas*, um dos recursos do HTML5, considerado por Meyer (2011) um dos mais poderosos da linguagem, foi utilizado no projeto.

O *canvas* possibilita a construção de imagens por meio de um contexto de duas dimensões (2D). Estas imagens, denominadas *frames*, podem ser modificadas dinamicamente sendo controladas por tempo ou eventos ocorridos no elemento. O elemento *canvas*, permite que *scripts* modifiquem a tela *bitmap* formando animações. O fator determinante é a resolução, ou seja, as medidas de altura e largura que definem o tamanho do *bitmap* do

*canvas*. A utilização do elemento ainda é altamente recomendada para a renderização de gráficos do jogo, ou outras imagens visuais em tempo real (MEYER, 2011; SHANKAR, 2012).

O HTML5 auxiliou na definição da estrutura do jogo ou projeto. Sendo assim, tem-se a necessidade de utilização de outras duas linguagens fundamentais em projetos *web*, o CSS (*Cascading Style Sheets*, ou em português, folha de estilos em cascata) para compor os estilos dos elementos do jogo e a linguagem *Javascript*, para a produção dos *scripts* que comporão a lógica de jogo, controle de tempo e espaço, validação das respostas, entre outros aspectos. É importante ressaltar que, como o elemento *canvas* não permite a utilização direta de folhas de estilos em cascata (CSS) em seus elementos internos, a mesma foi realizada por meio da utilização de *Javascript* (SHANKAR, 2012).

*Javascript* é uma linguagem de programação leve, interpretada pelos *browsers* (navegadores de internet) e que ainda dispõem de recursos de orientação a objetos. Por meio desta linguagem podem ser manipulados quaisquer elementos de um documento HTML, definindo estilos, mudando suas propriedades, definindo ações, entre outros aspectos (FLANAGAN, 2004).

A aplicação de RNAs compõe um ponto em destaque no jogo, já que o mesmo conta com um módulo onde são explorados todos os recursos do algoritmo *perceptron*, com a finalidade de validar as respostas dos jogadores de maneira correta e precisa, gerando resultados condizentes com as respostas fornecidas pelo jogador. Basicamente este módulo é utilizado nas fases do terceiro nível do protótipo do jogo implementado, onde o jogador deve escrever as vogais na tela.

## 4.2. Modelagem do Jogo

Primeiramente foram definidos os personagens, elementos e estória do jogo. O personagem de nome Zag (um “robozinho”), é o protagonista do jogo, denominado “Aprendendo com o Zag”. O jogo é ambientado em uma cidade, onde *Zag* se depara com diversas situações, contadas em formato de estória por sua professora, que faz a narração e conduz o jogador pelas fases do jogo. Durante o jogo, o aluno deve ajudar o personagem em certas atividades, como por exemplo, escrever sobre letras, procurar e encontrar as letras, e, ajudar *Zag* a fazer a sua primeira prova na escola.

A modelagem do jogo foi realizada com a técnica conhecida como *storyboard*, que é uma ferramenta criada para o desenvolvimento de quadros (frames) que compõem uma animação. Um *storyboard* pode representar um esboço do modelo de uma aplicação e mostrar como seus elementos estarão organizados, além disso, ajuda no planejamento do conteúdo de cada unidade, na disposição das mídias. O *storyboard* é o “rascunho” da aplicação permitindo aos responsáveis pelo projeto visualizarem sua estrutura de navegação, ou seja, discutirem a sequência do conteúdo e fazerem as revisões e o acompanhamento necessários (FALKEMBACH, 2005). Esta técnica foi adotada para demonstrar a estrutura e organização do protótipo do jogo *Aprendendo com o Zag*. A Figura 4 apresenta o *storyboard* do protótipo.

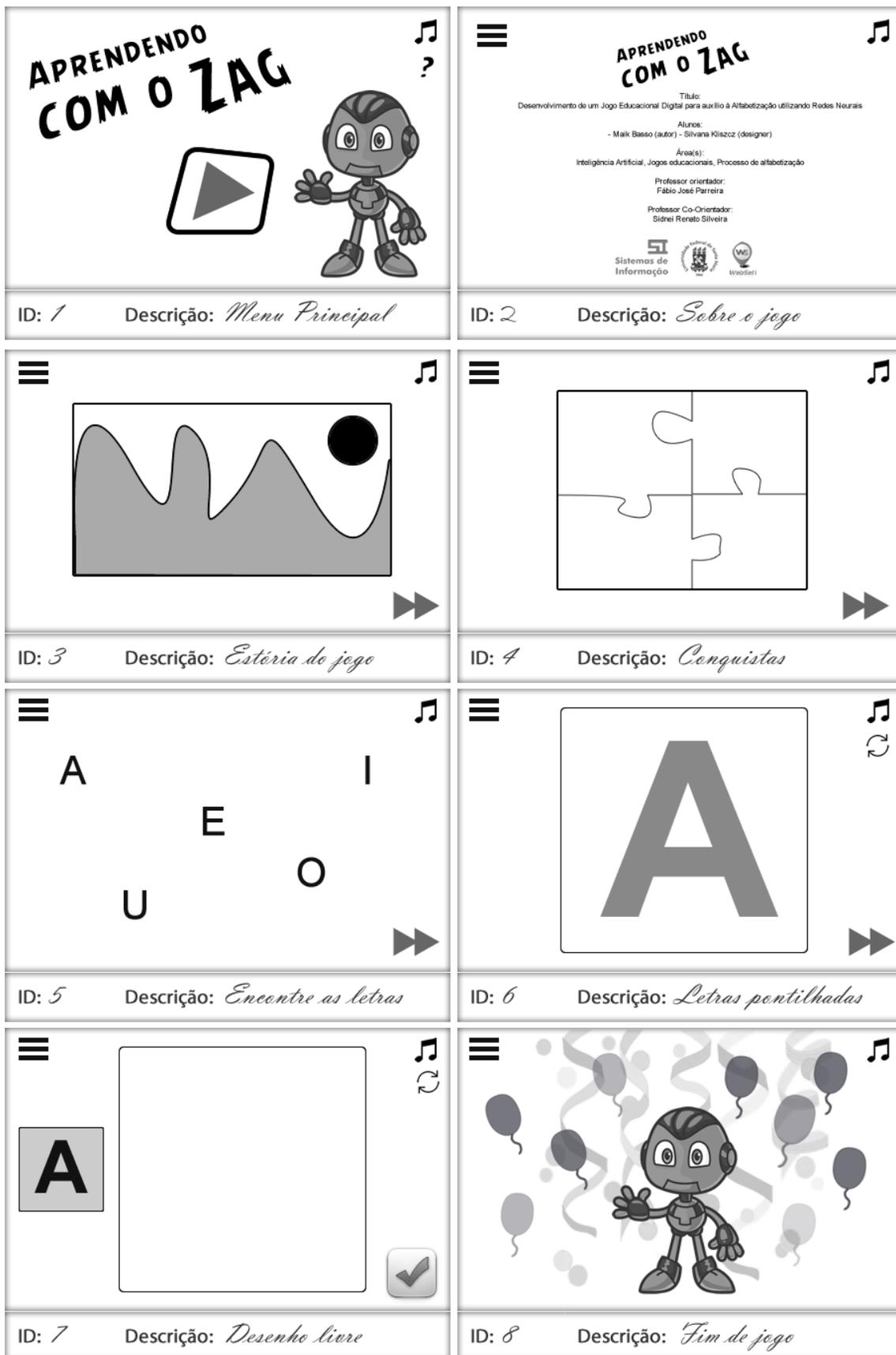


Figura 4. Storyboard do protótipo desenvolvido. Fonte: dos autores, 2015.

No decorrer do jogo, *Zag* acompanha o aluno por três níveis diferentes. No primeiro nível de jogo, o aluno deve ajudar *Zag* a encontrar todas as vogais do alfabeto, que estão “escondidas” em uma interface que representa a cidade de Robonópolis (cidade fictícia onde se passa a estória do jogo).

Já no segundo nível, a professora do *Zag*, pede ajuda ao aluno para que auxilie *Zag* em sua primeira tarefa na escola. A primeira tarefa é escrever todas as vogais em letra de “forma”, escrevendo em seu dispositivo (computador, *tablet* ou *smartphone*) em um quadro branco, desenhando sobre a letra em vermelho, que serve de referência à escrita nesta fase. Após escrever com o auxílio de uma espécie de caligrafia digital, o aluno chega ao último nível do jogo.

No último nível do jogo *Zag* encara seu primeiro dia de prova na escola. Neste nível o aluno deve escrever as vogais do alfabeto no quadro branco, porém agora sem o auxílio da letra em vermelho, aumentando consideravelmente o nível de dificuldade do nível. Neste nível as respostas são validadas por uma RNA, que realiza a comparação dos dados de treinamento, que são as respostas desejadas para as fases, com a resposta do jogador, definindo se a resposta do jogador está certa ou errada, demonstrando o resultado ao jogador.

O importante a ressaltar nesta etapa é que, a base de treinamento disponível para a fase de teste da RNA já se encontra armazenada no protótipo, não necessitando o treinamento a cada execução. Além disso, os dados contidos nesta base foram obtidos por meio de tentativa e erro, onde o conjunto de dados que se mostrou mais relevante perante os resultados desejados, foi utilizado para compor a base de treinamento.

Com o último nível de jogo concluído, a tela de créditos é apresentada ao aluno, finalizando o jogo.

### **4.3 Protótipo Implementado**

Logo no início deste projeto sentiu-se a necessidade de implementar uma *engine* para a construção do jogo, que resultou em um artigo publicado no Encontro anual de Tecnologia da Informação (EATI), realizado no Instituto Federal Farroupilha campus de Frederico Westphalen. O projeto da *engine* foi intitulado de *MB Engine* e foi a partir desta lógica e recursos de construção de interfaces que o jogo *Aprendendo com o Zag* foi construído. A *MB Engine* disponibilizou os recursos necessários como a criação de elementos gráficos do jogo, tal como sua estrutura e manipulação, controle de espaço e controle temporal, recursos estes fundamentais para o desenvolvimento da lógica de jogo. Outros recursos também de suma importância utilizados no jogo e que também foram disponibilizados pela *engine* foram os efeitos gráficos, utilizados para compor as animações do jogo, e os efeitos sonoros, disponibilizando controle total sobre os áudios que compõe o jogo (BASSO et. al., 2015).

Como mencionado anteriormente, o jogo implementado possui três níveis, onde foram utilizadas diferentes formas para que o jogador possa aprender a identificar e escrever as vogais. O primeiro nível chama-se de “busca a letra”, nele o aluno deve encontrar as letras escondidas em uma imagem. Para esta fase foi desenvolvida uma imagem que representa a cidade onde se passa a história do jogo, onde foram dispostas as letras a serem encontradas

de uma forma em que elas fiquem camufladas, porém a disposição das letras e a imagem não mudam no decorrer da fase, este recurso estará disponível em futuras versões do jogo.

Como descrito acima, foram dispostos vários elementos gráficos pela tela e em meio a estes foram dispostas as vogais. Cada elemento que representa uma vogal do alfabeto contém um evento *onclick*. Quando o jogador clica sobre a letra escondida, uma função é disparada tocando o áudio que diz ao jogador que a letra que foi encontrada e, por fim, a letra é removida da tela, sendo colocada no painel localizado na parte de baixo da tela, contendo todas as letras que devem ser encontradas.

A Figura 5 representa o primeiro nível do jogo, que também contém dois botões padrões em todas as interfaces do jogo, que são o botão de voltar ao menu principal e o botão de desabilitar a trilha sonora do jogo.



Figura 5. Primeiro nível do jogo. Fonte: Dos autores, 2015.

O segundo nível do jogo, dispõe de um quadro contendo uma letra em vermelho e o jogador deve escrever sobre a mesma, para que possa memorizar como se escreve a letra. Nesta tela as cores contrastantes, o vermelho da letra, o azul do traço e o branco do quadro, não estão dispostos desta forma por acaso. As cores contrastantes possibilitam que, a partir de uma função do contexto do *canvas* conhecida como *getImageData()*, seja possível recuperar uma matriz numérica contendo o tamanho do contexto (*height* e *width*) e também o vetor *data* que, em cada quatro posições, armazena uma informação que representa a cor de um *pixel* da tela. As cores estão representadas no formato RGB (*Red Green Blue*), sendo a primeira posição representando o nível de vermelho no *pixel*, a segunda o nível de verde, a próxima posição o nível de azul e por último o alfa, que representa o nível de transparência do *pixel*.

Uma função que faz a análise destes padrões de cor de cada *pixel* do *canvas* foi implementada e, a partir desta, tem-se à disposição um mapeamento de cores da tela. Então, utilizando os eventos da linguagem de programação *Javascript* correspondentes às ações do *mouse* de pressionar, arrastar e soltar, pode ser iniciada a escrita no *canvas* e a definição das ações do nível. Quando ocorre o evento que representa pressionar o botão do *mouse* (*onmousedown*), inicia-se a escrita do traço no *canvas*.

Quando ocorre o evento ao mover o *mouse* (*onmousemove*), é capturada a cor do *pixel* inicial da escrita, então é definido se o jogador está escrevendo dentro ou fora da letra. Caso esteja escrevendo dentro da letra em vermelho, a ação de escrever continua, senão a tela é limpa e uma mensagem sonora dizendo ao jogador para tentar novamente é disparada. Por fim, quando ocorre o evento de soltar o *mouse* (*onmouseup*) nesta interface, é disparada a função que verifica se o jogador terminou de escrever letra no *canvas*. Esse processo é executado para todas as vogais que são escolhidas de forma aleatória para que o jogador escreva todas as vogais e, quando concluí-las, passe para a próxima fase.

Outro recurso que ainda não foi mencionado nesta fase é o botão reiniciar. Se o jogador optar por começar novamente a escrever a letra atual, pode clicar no botão reiniciar para começar novamente. A Figura 6 apresenta o segundo nível do jogo.



**Figura 6. Segundo nível do jogo. Fonte: Dos autores, 2015.**

Por fim, tem-se o terceiro e último nível do jogo, que também é o mais importante porque é o foco deste trabalho. Este nível segue a mesma lógica do nível anterior, porém possui algumas particularidades. Nesta fase a professora de Zag diz uma letra ao jogador que deve escrevê-la no quadro em branco, sem auxílio de nenhuma imagem, e clicar no botão verde para verificar a sua resposta. A Figura 7 apresenta a interface do terceiro nível.

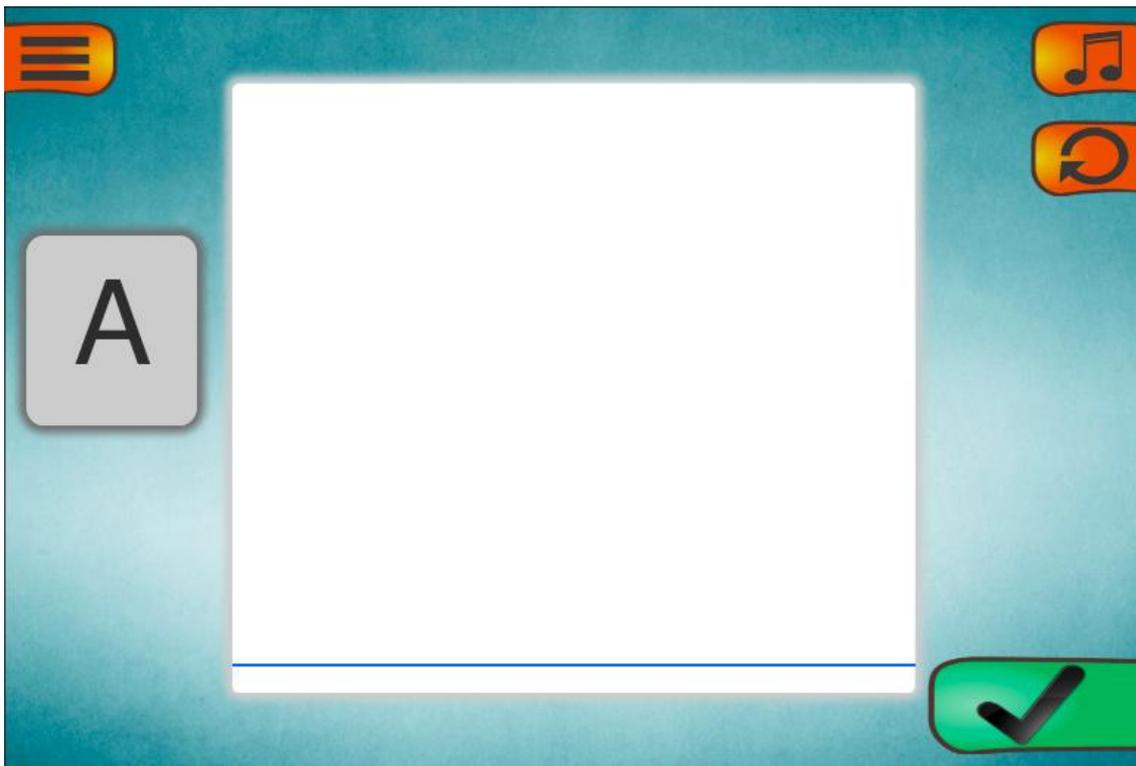


Figura 7. Terceiro nível do jogo. Fonte: Dos autores, 2015.

A resposta (escrita da letra) nesta fase é avaliada por uma RNA baseada em padrões. Para tanto, precisou-se desenvolver uma classe chamada de PIC (Processamento de Imagens em *Canvas*) que gera os padrões de treinamento e de teste da RNA, para que se possa ter uma unificação no padrão de coleta dos dados que a rede neural irá trabalhar.

A classe PIC possui dois *canvas* auxiliares ocultos (não são exibidos ao jogador). O primeiro passo da PIC é fazer o recorte do desenho feito pelo jogador, eliminando os espaços em brancos que são desnecessários e duplicar a imagem para o primeiro *canvas* auxiliar que recebe o nome de *canvasRecorte*. Esse processo de redimensionamento é feito pelo cálculo das dimensões *width* (largura), *height* (altura) e também as posições *x* e *y* do plano cartesiano, no conjunto de dados obtidos pelo método *getImageData()* do contexto do *canvas*. Após ter recuperado no *canvas* de recorte a parte relevante da imagem, recupera-se novamente, por meio do método *getImageData()* a matriz de dados atualizada. A partir desta atualização, utiliza-se a função *putImageData()*, que foi reescrita para atender as necessidades do projeto, para desenhar os dados no segundo *canvas* auxiliar que possui um tamanho padrão de sete *pixels* de largura por nove de altura. Então se recupera novamente a matriz de dados que terá exatamente as dimensões sete por nove e gera-se a matriz *pixelada* (matriz que possui conjuntos de valores em que cada posição da mesma representa um pixel da tela), atribuindo-se “1” para a posição da matriz onde o *pixel* possuir cor e “-1” para onde o *pixel* for branco ou não possuir cor. Então, após ter realizado todo o processamento descrito, obtêm-se uma matriz *pixelada*, pelo método *getMatriz()* da classe PIC.

No jogo implementado, é importante ressaltar que a saída da classe PIC foi ajustada para uma matriz *pixelada* de sete colunas por nove linhas, por ser a menor matriz *pixelada* que pode representar uma letra do alfabeto com precisão. Esta medida foi obtida de forma empírica, na base de tentativa e erro, a partir da realização de vários testes, começando com medidas maiores para a matriz *pixelada*, chegando-se ao menor número possível da matriz em que não havia perda na taxa de reconhecimento. Outro ponto relevante é que uma matriz de proporção sete colunas por nove linhas representa um total de 63 posições ou mapeamentos de *pixels* do *canvas*. Sendo assim, cada uma destas posições corresponde a um neurônio de entrada na rede neural. Quanto maior a quantidade de informações processadas pela rede neural, mais lenta será sua execução, como desempenho é fundamental em um jogo optou-se por adotar o menor conjunto de entradas.

Uma das vantagens da utilização da classe PIC para a recuperação dos dados de treinamento e teste da RNA é que, como o jogo é responsivo, ou seja, atende a várias resoluções de tela, a matriz *pixelada* no final do processo PIC sempre terá o mesmo tamanho, o que possibilita que se tenha somente um conjunto de pesos de treinamento gerados para a execução dos testes no RNA em diversas resoluções de tela.

Então, com a classe PIC funcionando corretamente na interface do jogo, coletam-se as matrizes *pixeladas* das vogais escritas por várias pessoas diferentes e, juntamente com a matriz *pixelada* de vogais digitais, forma-se a base de treinamento da RNA. Esta base de treinamento foi composta por 17 amostras para cada letra vogal, totalizando um total de 85 amostras, onde cada uma destas corresponde a uma saída da classe PIC. Estas amostradas foram geradas a partir da escrita de algumas pessoas que se dispuseram a contribuir com o projeto (17 voluntários). Não há um número correto de amostras para a RNA funcionar perfeitamente. Acredita-se que, quanto mais amostras existirem, melhor será o reconhecimento. Após a formação da base de treinamento, ela foi devidamente armazenada na classe *DadosRNA*, bem como os alvos do treinamento e do teste, e posteriormente os pesos gerados pelo treinamento da RNA.

Para a etapa de treinamento da RNA e geração dos pesos que contém o conhecimento da rede, foram executados basicamente seis passos de forma consecutiva. O primeiro deles é o passo 0, onde inicializam-se a matriz de pesos. No próximo passo, o passo 1, define-se a condição de parada da rede neural, que é uma *flag* que recebe o valor *false*, e também é feita a inicialização do contador de passos necessários para o treinamento da RNA e de uma taxa de aprendizagem no valor de "0.005", valor este obtido de forma empírica (tentativa e erro). Esta taxa ajuda a controlar a velocidade e a qualidade do conhecimento gerado pela rede neural.

No passo 2, implementa-se uma estrutura de repetição do tipo *for* que executa, para cada amostra de treinamento ( $x_1..x_n$ ) ( $t_1..t_n$ ), os passos 3, 4 e 5 do treinamento. Dentro deste laço de repetição, no passo 3 da etapa de treinamento, calcula-se o potencial de ativação das saídas (*Yin*) para cada neurônio de saída. Após isso, no passo 4, calcula-se a resposta do neurônio de saída (*y*), aplicando a função de ativação, que define o valor da saída (*y*). No passo 5, atualizam-se os pesos se caso a saída (*y*) for diferente do valor do *target*/alvo (*t*). Executa-se esta estrutura de repetição até que a condição de parada seja

verdadeira, então a função de treinamento retorna a matriz de pesos com o conhecimento da rede.

Para o jogo implementado foram necessários um total de 63 neurônios de entrada ( $x$ ), já que utilizou-se uma matriz  $7 \times 9$ , e de 5 neurônios de saída ( $y$ ), cada um representando a saída de uma vogal do alfabeto. Para o processo de treinamento foram necessários aproximadamente 23 passos de execução das instruções da RNA para obter o conhecimento necessário para a etapa de teste, os pesos ( $w$ ). A Figura 8 representa as fases de execução da RNA, desde a coleta de dados da classe PIC até a finalização do treinamento da rede.

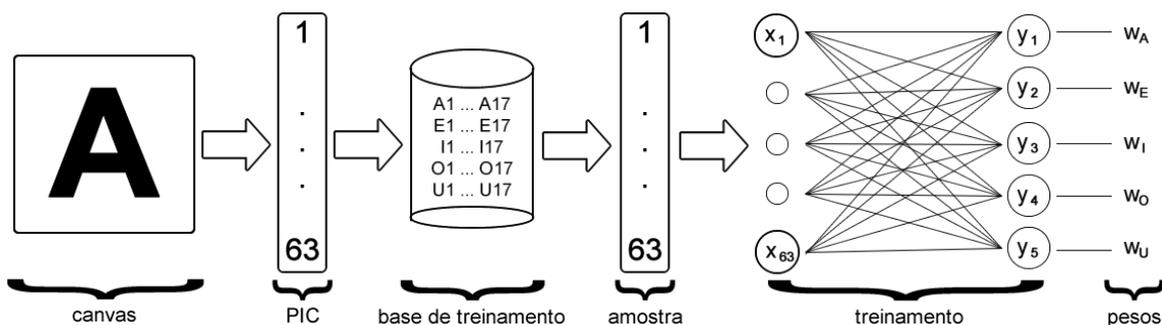


Figura 8. Coleta de dados PIC e treinamento da RNA. Fonte: Dos autores, 2015.

Para evitar que a rede tenha que executar a etapa de treinamento toda a vez que o botão verificar seja pressionado, os pesos resultantes do treinamento foram armazenados na classe de *DadosRNA* em um arquivo *Javascript*, para que possam ser recuperados rapidamente para a execução da etapa de teste.

Neste mesmo nível, como mencionado anteriormente, toda a vez que o botão verde, verificar a resposta, for pressionado, a etapa de teste da RNA é executada. Para que isso aconteça, primeiramente os dados da base de treinamento são recuperados, e a classe PIC gera em tempo real a matriz *pixelada* que irá compor a base de teste da RNA, da mesma maneira que ocorre na etapa anterior ao treinamento da rede. Executados estes procedimentos, tem-se os dados necessários para a realização do teste.

Na etapa de teste, a RNA executa novamente os passos 3, 4 e 5 da rede com algumas adaptações. No passo 3, criou-se novamente a estrutura de repetição do tipo *for* que calcula o potencial de ativação da saída ( $Y_{in}$ ) para cada neurônio de saída. Dentro desta estrutura de repetição, no passo 4, calcula-se a resposta do neurônio de saída ( $y$ ), aplicando a função de ativação obtendo-se, assim, a resposta do limiar da rede. Em uma rede neural o limiar é o número central, que fica entre a resposta considerada correta e a incorreta. O limiar que foi definido para a rede neural do protótipo deste artigo foi o valor de "0.5", valor obtido de forma empírica (tentativa e erro).

No passo 5 da rede realiza-se o teste propriamente dito, por meio da comparação entre os dados do treinamento e os dados processados no teste. Então é feita a identificação da letra, se a letra solicitada for igual à letra identificada, a função de teste da rede neural retorna verdadeiro, caso contrário retorna falso. A Figura 9 representa os procedimentos que ocorreram na etapa de teste da RNA.

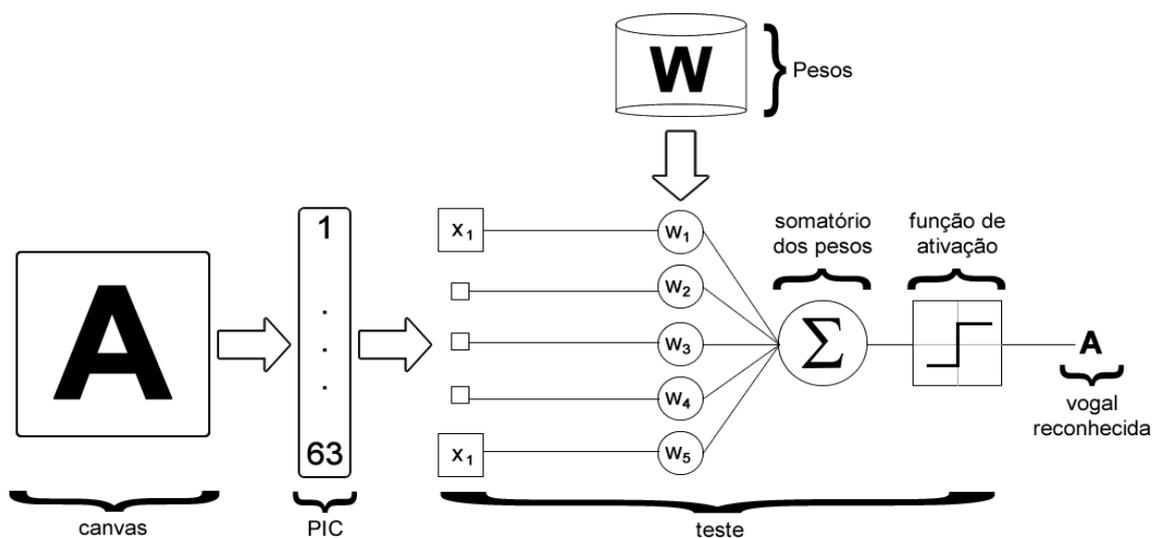


Figura 9. Coleta de dados PIC e teste da RNA. Fonte: Dos autores, 2015.

Quando a RNA identificar a letra, e esta for igual à letra solicitada é disparado um sinal sonoro mostrando que o jogador acertou a letra. Posteriormente, é requisitado que o mesmo desene outra letra aleatória. Quando o jogador escrever corretamente todas as vogais e a RNA identificá-las, a fase é dada por encerrada.

#### 4.4. Testes e Validação

Os testes da aplicação ocorreram lado a lado com o desenvolvimento. Vários protótipos do jogo em fase de desenvolvimento foram disponibilizados. A última versão estável do protótipo ficou disponível de forma *online* para que todos pudessem acessar e usufruir do mesmo, para que fosse possível obter um *feedback* e para que os alunos também pudessem continuar o seu aprendizado em suas residências, não ficando limitados a um material disponível somente em sala de aula.

A validação do projeto ocorreu no dia 13 de novembro de 2015, nas dependências da Escola Estadual de Educação Básica Padre Abílio de Marcos Sponchiado, localizada em Vista Alegre - RS. Primeiramente os responsáveis pela Direção da Escola foram contatados, para que se buscasse a permissão para realizar a validação. O termo de autorização (Anexo A) comprova esta permissão.

Com o termo de autorização assinado (Anexo A), conversou-se com as professoras responsáveis pela turma do primeiro ano do Ensino Fundamental, para que fosse possível explicar a proposta do jogo e como seria realizada a validação. As professoras receberam o questionário (disponível no Anexo B) a ser preenchido para participarem da validação. Então, a turma de alunos, composta por oito crianças entre cinco a sete anos de idade, foi conduzida para o laboratório de informática da escola, que dispunha de vinte computadores para a execução do jogo. Como o jogo é multiplataforma, optou-se por utilizar a versão *online* (disponível na *web*) do jogo para a validação.

Antes do início das atividades perguntou-se aos alunos quantos deles tinham conhecimento da utilização de computadores e outros dispositivos. Sem que houvesse surpresa, todos afirmaram que tinham conhecimento. O que mais impressionou foi o fato de

que os próprios alunos ligavam suas máquinas e iniciavam os navegadores de Internet, questionando os autores para que fosse disponibilizado o endereço para o acesso ao jogo. Segundo os dados coletados no roteiro de observação (Anexo C), os alunos nem esperaram para que todos abrissem o jogo e, sem nenhuma dificuldade, iniciaram as partidas e foram percorrendo as fases do jogo. Os alunos que terminavam primeiro foram convidados a auxiliar os colegas com dicas.

O que mais chamou a atenção é que o jogo prendeu a atenção das crianças, fazendo com que as mesmas trabalhassem em grupo, principalmente nas fases dois e três, que eram as fases da escrita na tela, que tinham uma dificuldade maior, por estarem sendo jogadas com auxílio de um *mouse*.

Já que a escola não dispunha de dispositivos móveis (tais como celulares ou *tablets*) para a realização dos testes, utilizou-se um celular, de propriedade de um dos autores deste trabalho, e os alunos foram convidados para jogar novamente. O resultado foi, novamente, positivo, já que todos os alunos terminaram as fases sem demais dificuldades. Ao perguntá-las sobre qual dispositivo as mesmas acharam mais difícil a jogabilidade, todas responderam no computador.

O questionário aplicado às professoras (Anexo B), composto por seis perguntas, foi respondido pelas professoras após as crianças terem terminado de jogar e testar o jogo. A primeira pergunta que constava no questionário era se os alunos tinham contato com atividades de informática na escola. As professoras responderam que os mesmos tinham contato semanalmente em um programa que acontece no laboratório da escola e é chamado de “Curso mais Educação”, onde elas desempenhavam atividades como aprender a digitar e usar a internet.

Na segunda questão, as mesmas foram questionadas se já haviam utilizado outros jogos educacionais digitais como atividade de apoio. As professoras responderam que já haviam utilizado outros jogos no processo de alfabetização, porém não tinham usado um jogo com atividades mais específicas como o jogo desenvolvido neste artigo. A terceira questão fez referência a se os alunos tiveram interesse em utilizar o jogo educacional implementado. Como resposta a essa questão, as professoras afirmaram que sim, ressaltando que esta é uma atividade atrativa e que as leva a realizar as atividades do jogo com muito afinho e interesse.

Outra questão deste instrumento, perguntava se a proposta do jogo desenvolvido era relevante para apoiar o processo de alfabetização das crianças. A resposta a este item foi sim, e as professoras ressaltaram, ainda, que o jogo pode ser usado para reforçar o conteúdo visto em classe e que pode ajudar a desenvolver as atividades diárias desempenhadas pelos alunos.

A seguir, as professoras foram questionadas sobre os pontos positivos da aplicação desenvolvida. As mesmas afirmaram que o jogo ajuda na coordenação motora, devido à escrita na tela com o mouse, no raciocínio lógico, na concentração, e em um processo de alfabetização mais prazeroso e divertido.

A Figura 10 demonstra o momento em que as crianças utilizavam o mouse para desenhar as letras na terceira fase do jogo.



**Figura 10. Validação da terceira fase do jogo. Fonte: Dos autores, 2015.**

Acredita-se que o processo de validação apresentou resultados satisfatórios para o projeto, já que os alunos demonstraram entusiasmo pelo jogo. Dentre os comentários realizados na última questão que pedia se haviam pontos a se melhorar no jogo, uma sugestão chamou a atenção: “As universidades deveriam pensar na possibilidade de trazer oficinas e material de qualidade como esse jogo para os professores poderem repassar para os alunos das séries iniciais”.

## **5. Considerações Finais**

Com base nos estudos realizados para desenvolver este trabalho, acredita-se que os jogos educacionais digitais exercem uma forte influência nos processos de ensino e de aprendizagem das crianças. Os jogos podem prender a atenção e ensinar de forma simples e divertida todo o conteúdo proposto.

Acredita-se que foi possível alcançar os objetivos propostos, envolvendo a implementação do jogo educacional digital proposto, aplicando RNAs, além da disponibilização do jogo, de forma gratuita, na *Google Play* e na *Firefox Marketplace*, bem como via *web*. Com base na validação realizada, acredita-se que foi possível criar uma ferramenta de qualidade para apoiar os processos de ensino e de aprendizagem dos alunos das séries iniciais do Ensino Fundamental, além de mostrar uma aplicação possível de técnicas de Inteligência Artificial na construção de jogos.

Este estudo sobre jogos educacionais digitais também rendeu frutos adicionais aos previstos na proposta deste trabalho, que foi o desenvolvimento da *MB engine* para auxílio à construção de jogos digitais em contexto *2D*(duas dimensões) (BASSO et. al., 2015).

A maior dificuldade encontrada no desenvolvimento deste projeto envolveram a interdisciplinaridade e o grande número de tecnologias/ferramentas utilizadas pois para a construção de um jogo, faz-se necessário integrar diferentes áreas do conhecimento. Neste

jogo foram utilizados conhecimentos sobre projeto de jogos, produção sonora, produção visual, sem contar o estudo sobre as tecnologias de desenvolvimento de jogos e de aplicação dos algoritmos de RNA.

Por outro lado, a publicação do projeto também teve suas dificuldades, já que as plataformas escolhidas para a publicação (*Google Play* e *Firefox Marketplace*) estão em constante atualização, e possuem um número enorme de exigências e conformidades que o aplicativo a ser publicado deve seguir, não existindo um material explicativo suficiente e resumido de forma clara para a utilização das mesmas.

Como trabalhos futuros, pretende-se desenvolver uma segunda versão do jogo educacional digital, incluindo todas as letras do alfabeto, além de disponibilizar outras fases, aplicando diferentes técnicas de IA. Outra opção seria melhorar a classe PIC para que a mesma possa criar padrões de identificação levando em consideração as cores e não somente a opção monocromática e, também levar em consideração a disposição da imagem, realizando ajustes de ângulo e distorção da mesma.

## Referências

- BASSO, M; LOPES, C. P.; PARREIRA, F. J.; SILVEIRA, S. R. (2015). **MB Engine: Game Engine para a Construção de Jogos em HTML 5**. Anais do VI EATI: VI Encontro Anual de Tecnologia da Informação. Instituto Federal Farroupilha: Frederico Westphalen.
- BISPO, D. M.; UMBUZEIRO, G. A.; ZABEU, M. S.; BORGES, M. A. F. (2012). Desenvolvimento de Jogo Educacional sobre Ecotoxicologia Utilizando HTML5. Limeira: **RBIE: Revista Brasileira de Informática na Educação**, v. 20, n. 1. Disponível em: <<http://www.br-ie.org/pub/index.php/rbie/article/view/1366>>. Acesso em 17 maio de 2015.
- FALKEMBACH, G. A. M. (2005). Concepção e Desenvolvimento de Material Educativo Digital. **Renote: Novas Tecnologias na Educação**, v. 3, n. 1. CINTED-UFRGS. Disponível em: <<http://seer.ufrgs.br/index.php/renote/article/view/13742/7970>>. Acesso em 23 de maio de 2015.
- FLANAGAN, D. (2004). **JavaScript: o guia definitivo**. 4. ed. Porto Alegre: Bookman.
- HAYKIN, S. (2001) **Redes Neurais: princípios e prática**. 2. ed. Porto Alegre: Bookman.
- LUGER, G. F. (2004). **Inteligência artificial: estruturas e estratégias para a solução de problemas complexos**. 4. ed. Porto Alegre: Bookmann.
- MEYER, J. (2011). **O Guia Essencial do HTML5: Usando jogos para aprender HTML5 e Javascript**. Rio de Janeiro: Editora Ciência Moderna.
- MILLINGTON, I.; FUNGE, J. (2009). **Artificial Intelligence for Games**. 2. ed. Burlington, USA: Morgan Kaufman.

- MORATORI, P. B. (2003). **Por Que Utilizar Jogos Educativos No Processo De Ensino Aprendizagem?** Trabalho da disciplina de Introdução à informática na educação. Mestrado em Informática aplicada à Educação. Rio de Janeiro: Universidade Federal Do Rio De Janeiro (UFRJ). Disponível em: <[http://ucbweb2.castelobranco.br/webcaf/arquivos/23678/15577/t\\_2003\\_patrick\\_barbosa\\_moratori.pdf](http://ucbweb2.castelobranco.br/webcaf/arquivos/23678/15577/t_2003_patrick_barbosa_moratori.pdf)>. Acesso em 15 março de 2015.
- OLIVEIRA, G. A. A.; BETTIO, R. W.; RODARTE, A. P. M.; BRAZ, J. E.; FERRARI, F. B. (2014). GrubiBots Educacional: jogo para o ensino algoritmos na educação básica. Lavras: **SBIE**: Simpósio Brasileiro de Informática na Educação. Disponível em: <<http://www.br-ie.org/pub/index.php/sbie/article/view/2988>>. Acesso em 17 de maio de 2015.
- PASCHOAL, L.; BENTO, T.; VELASCO, T.; SCHOCAIR, C. O.; CASTANEDA, R.; OLIVEIRA, T.; OGASAWARA, E. (2014). JOE: Jogo Ortográfico Educacional. Rio de Janeiro: **SBIE**: Simpósio Brasileiro de Informática na Educação. Disponível em: <<http://www.br-ie.org/pub/index.php/sbie/article/view/2995>>. Acesso em 17 de maio de 2015.
- PEREIRA, C. C.; ROCHA, C. M. G. S. M. L.; FERREIRA, P. A.; MARTINS, S. R. S. (2013). **Processo de Alfabetização**: uma realidade a ser repensada. Caldas Novas: FACULDADE DE CALDAS NOVAS (UNICALDAS). Disponível em: <<http://www.unicaldas.edu.br/uploads/files/alfabetizacao.pdf>>. Acesso em 15 de março de 2015.
- RIEDER, R.; ZANELATTO, E. M.; BRANCHER, J. D. (2004). **Observação e Análise da Aplicação de Jogos Educacionais Bidimensionais em um Ambiente Aberto**. IX Taller Internacional de Software Educativo (TISE). Erechim: Universidade Regional Integrada do Alto Uruguai e das Missões. Disponível em: <[http://www.nce.ufrj.br/ginape/iga502/Material\\_aulas/Observa%C3%A7%C3%A3o%20e%20An%C3%A1lise%20da%20Aplica%C3%A7%C3%A3o%20de%20Jogos.pdf](http://www.nce.ufrj.br/ginape/iga502/Material_aulas/Observa%C3%A7%C3%A3o%20e%20An%C3%A1lise%20da%20Aplica%C3%A7%C3%A3o%20de%20Jogos.pdf)>. Acesso em 22 de março de 2015.
- RUSSELL, S.; NORVIG, P. (2004) **Inteligência Artificial**. 2. ed. Rio de Janeiro: Elsevier.
- SAVI, R.; ULBRICHT, V. R. (2008). Jogos Digitais Educacionais: Benefícios E Desafios. **Renote**: Novas Tecnologias na Educação, v. 6, n. 1. CINTED-UFRGS. Disponível em: <<http://seer.ufrgs.br/renote/article/view/14405/8310>>. Acesso em 22 de março de 2015.
- SHANKAR, A. R. (2012) **Pro HTML5 Games (Expert's Voice in Web Development)**. E-book: Apress. Disponível em: <<http://www.apress.com/9781430247104>>. Acesso em 17 de maio de 2015.
- SILVA, I. N.; SPATTI, D. H.; FLAUZINO, R. A. (2010). **Redes Neurais Artificiais**: para engenharia e ciências aplicadas. São Paulo: Artliber.

- SILVA, M. G. (2014) **Processo de Alfabetização**. Trabalho de conclusão do curso de Licenciatura Plena em Pedagogia. Catolé Do Rocha - PB: Universidade Estadual da Paraíba. Disponível em: <<http://dspace.bc.uepb.edu.br:8080/jspui/bitstream/123456789/4876/1/PDF%20-%20Maria%20da%20Gra%C3%A7as%20Silva.pdf>>. Acesso em 15 de março de 2015.
- SILVEIRA, S. R.; RANGEL, A. C. S.; CIRÍACO, E. L. (2012) Utilização De Jogos Digitais Para O Desenvolvimento Do Raciocínio Lógico-Matemático. Canoas: **Tear**: Revista de Educação Ciência e Tecnologia. Disponível em: <<http://seer.canoas.ifrs.edu.br/seer/index.php/tear/article/view/3>>. Acesso em 15 de março de 2015.
- SILVEIRA, S. R.; BARONE, D. A. C. (1998) **Jogos Educativos Computadorizados Utilizando a Abordagem de Algoritmos Genéticos**. In: Congresso da Rede Iberoamericana de Informática na Educação, 1998, Brasília. Actas... Brasília: Universidade de Brasília.
- TISSOT, H. C.; CAMARGO, L. C.; POZO, A. T. R. (2012) **Treinamento de Redes Neurais Feedforward: comparativo dos algoritmos Backpropagation e Differential Evolution**. IX Encontro Nacional de Inteligência Artificial (ENIA-2012). Curitiba: Universidade Federal do Paraná (UFPR). Disponível em: <[http://www.ppgia.pucpr.br/~enia/anais/enia/artigos/105243\\_2.pdf](http://www.ppgia.pucpr.br/~enia/anais/enia/artigos/105243_2.pdf)>. Acesso em 22 de março de 2015.
- VELLASCO, M. M. B. R. (2007) **Redes Neurais Artificiais**. Laboratório de inteligência computacional aplicada (ICA). Rio de Janeiro: Pontifícia Universidade Católica do Rio de Janeiro (PUC-RJ). Disponível em: <<http://www2.ica.ele.puc-rio.br/Downloads/33/ICA-introdu%C3%A7%C3%A3o%20RNs.pdf>>. Acesso em 22 março de 2015.

## Anexo A

### Termo de Autorização

Prezado(a) Vice-Diretora.

Vimos, por meio deste, apresentar-lhes o trabalho "Desenvolvimento de um Jogo Educacional Digital para auxílio à Alfabetização utilizando Redes Neurais" de autoria do acadêmico "Maik Basso" do Curso de Sistemas de Informação da UFSM/Frederico Westphalen, e solicitar autorização para que o referido trabalho possa ser validado nas dependências da Escola Estadual de Educação Básica Padre Abílio de Marcos Sponchiado, com a turma de alunos do 1º ano do ensino fundamental.

O trabalho é parte integrante do TGSi – Trabalho de Graduação em Sistemas de Informação do Curso de Bacharelado em Sistemas de Informação, integrando as atividades do grupo de pesquisa IATE – Inteligência Artificial e Tecnologia Educacional, sobre orientação dos professores Fábio José Parreira e Sidnei Renato Silveira.

A validação consiste em utilizar, de forma prática, o jogo educacional digital "Aprendendo com o Zag" desenvolvido, visando validá-lo e aprimorá-lo, nas dependências do Laboratório de Informática da escola.

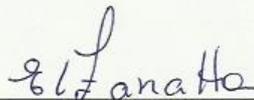
Desde já agradecemos pela sua valiosa contribuição na validação deste trabalho.

Qualquer informação adicional pode ser obtida pelo e-mail: maik@maikbasso.com.br

### AUTORIZAÇÃO

Eu, Elenir T. G. Zanatta autorizo que o acadêmico Maik Basso realize a validação do jogo educacional digital Aprendendo com o Zag nas dependências da Escola Estadual de Educação Básica Padre Abílio de Marcos Sponchiado, com a turma de alunos do 1º ano do ensino fundamental.

GOVERNO DO ESTADO DO RIO GRANDE DO SUL  
SECRETARIA DE EDUCAÇÃO E CULTURA  
20ª DELEGACIA DE EDUCAÇÃO  
ESCOLA ESTADUAL DE EDUCAÇÃO BÁSICA  
Pe. ABÍLIO DE MARCOS SPONCHIADO  
DEC. DE CRIAÇÃO Nº 1416 DE 29/12/44  
DEC. DE REORGANIZAÇÃO Nº 26391 DE 20/12/77  
DEC. DE TRANSFORMAÇÃO Nº 33.109 DE 12/01/89  
(0XX) 55-3730-1005  
VISTA ALEGRE-RS



Escola Estadual de Educação Básica  
Padre Abílio de Marcos Sponchiado.

13 de novembro de 2015.

## Anexo B

### Questionário para o Professor

Escola:

Turma:

Série:

Número de Alunos:

Professor que acompanhou a atividade:

1) Os alunos realizam, regularmente, atividades no Laboratório de Informática?

Sim. De que forma e com que frequência? \_\_\_\_\_

Não

2) Os alunos já utilizaram outros jogos educacionais digitais como forma de apoio às atividades desenvolvidas na sala de aula?

Sim. Foram utilizados que tipos de jogos e para quais áreas de estudo?

\_\_\_\_\_  
 Não

3) Os alunos demonstraram interesse em participar das atividades propostas no jogo?

Sim

Não

Por quê? (Justifique sua resposta)

4) Com relação ao jogo educacional digital desenvolvido, você considera que a proposta é relevante para apoiar os processos de ensino e de aprendizagem na área de Alfabetização?

Sim

Não

Por quê? (Justifique sua resposta)

5) Quais os pontos positivos que você pode destacar com relação ao jogo educacional desenvolvido?

6) Quais os pontos a melhorar que você pode destacar com relação ao jogo educacional desenvolvido?

(Fonte: dos autores, 2015)

## Anexo C

### Roteiro de Observação

Escola:

Turma:

Série:

Número de Alunos:

Professor que acompanhou a atividade:

1) Os alunos demonstraram interesse/entusiasmo ao serem convidados a utilizar o jogo educacional digital desenvolvido?

Sim

Não

Comentários/Observações:

2) Os alunos utilizaram os computadores de forma individual, em duplas ou em grupos?

3) Os alunos sentiram alguma dificuldade para entender as regras de funcionamento do jogo? (os alunos precisaram pedir ajuda para a professora ou para o acadêmico?)

Sim

Não

Comentários/Observações:

4) Durante o jogo, os alunos estavam entusiasmados a desenvolver as atividades propostas e atingir as próximas fases do jogo?

Sim

Não

Comentários/Observações:

5) Os alunos fizeram algum comentário (positivo ou negativo) que possa ser utilizado para validar o desenvolvimento do jogo ou para repensar a proposta desenvolvida?

Sim

Não

Comentários/Observações:

(Fonte: dos autores, 2015)