

CENTRO UNIVERSITÁRIO DE ANÁPOLIS – UNIEVANGÉLICA
BACHARELADO EM ENGENHARIA DE COMPUTAÇÃO

GABRIEL NERES DA SILVA
TIAGO NEVES DE SOUZA

DESENVOLVIMENTO DE UMA APLICAÇÃO PARA
DISPONIBILIZAÇÃO DE APLICATIVOS CLASSIFICADOS
CONFORME TAXONOMIA DE BLOOM

ANÁPOLIS - GO
2017

GABRIEL NERES DA SILVA
TIAGO NEVES DE SOUZA

**DESENVOLVIMENTO DE UMA APLICAÇÃO PARA
DISPONIBILIZAÇÃO DE APLICATIVOS CLASSIFICADOS
CONFORME TAXONOMIA DE BLOOM**

Projeto de Pesquisa apresentado ao Curso de Bacharelado em Engenharia de Computação do Centro Universitário de Anápolis – Uni EVANGÉLICA, sob orientação do Prof. Ms. Millys Fabrielle Araujo Carvalhaes.

ANÁPOLIS - GO
2017

LISTA DE ILUSTRAÇÕES

Figura 1 - Níveis de conhecimento da Taxonomia de Bloom publicada em 1956.....	16
Figura 2 - Novo modelo da Taxonomia de Bloom proposta por Anderson em 2001.	16
Figura 3 - Estrutura de comunicação entre <i>view</i> e <i>model</i> no Angular Js.....	19
Figura 4 - Estrutura de comunicação reativa do Vue JS.	20
Figura 5 - Modelo arquitetural MVC.	21
Figura 6 - Opções de exportação para jogos criados no Construct 2.....	25
Figura 7 - Processo de adição de efeitos em um objeto do cenário.....	26
Figura 8 - Novo layout de desenvolvimento do Construct 3.	26
Figura 9 - Interface de desenvolvimento do RPG Maker VX ACE.	27
Figura 10 - Suporte multiplataforma da versão RPG Maker MV.	28
Figura 11 - Interface inicial da Unity 3D em sua versão 5.....	29
Figura 12 - Configuração das propriedades de um objeto do dentro da Unity 3D.	30
Figura 13 - Interface de programação do Blue Print.....	31
Figura 14 - Processo de programação orientada a eventos utilizando o Blue Print.	31
Figura 15 - Tela de descrição de um dos níveis da Taxonomia.	35
Figura 16 - Tela de distribuição do jogo por turma.	35
Figura 17 - Jogo exemplo distribuído para os alunos.	36
Figura 18 - Tela de estatísticas referentes ao jogo distribuído.	36
Figura 19 - Jogo de quis desenvolvido para exemplo.	37
Figura 20 - Apresentação de uma pergunta dentro do quiz.	38
Figura 21 - Fim de jogo e pontuações finais no jogador.	38
Figura 22 - Modal de exclusão de um jogo distribuído.....	39

LISTA DE ABREVIATURAS E SIGLAS

API	<i>Application Programming Interface</i>
2D	2 Dimensões
3D	3 Dimensões
CSS	<i>Cascading Style Sheets</i>
C#	C Sharp
DOM	<i>Document Object Mode</i> (Modelo de Objeto de Documento)
FPS	<i>First Person Shooter</i>
HTML	<i>Hyper Text Markup Language</i>
LIB	Biblioteca
MEC	Ministério da Educação
MVC	<i>Model View Controller</i>
ORM	<i>Object Relational Mapper</i>
PHP	<i>Hypertext Preprocessor</i>
RPG	<i>Role Playing Game</i>
TPS	<i>Third Person Shooter</i>
TV	<i>Television</i>
XML	<i>Extensible Markup Language</i>

SUMÁRIO

RESUMO	9
ABSTRACT	10
INTRODUÇÃO.....	11
1 OBJETIVOS.....	12
1.1 OBJETIVO GERAL	12
1.2 OBJETIVOS ESPECÍFICOS	12
2 JUSTIFICATIVA.....	13
3 REFERENCIAL TEÓRICO	14
3.1 SISTEMA DE ENSINO-APRENDIZAGEM	14
3.1.1 DESAFIOS DA EDUCAÇÃO	14
3.1.2 TAXONOMIA DE BLOOM.....	15
3.2 TECNOLOGIAS DE DESENVOLVIMENTO.....	17
3.2.1 HTML.....	17
3.2.2 CSS.....	17
3.2.3 JAVA SCRIPT	18
3.2.4 ANGULAR 4.....	18
3.2.5 VUE JS	19
3.2.6 PHP.....	20
3.2.7 ZEND FRAMEWORK	21
3.2.8 LARAVEL	22
3.2.9 LUMEN.....	22
3.3 GAMEFICAÇÃO	23
3.4 GAME ENGINE.....	23
3.4.1 CONSTRUCT 2	24
3.4.2 RPG MAKER.....	26
3.4.3 UNITY 3D.....	28
3.4.4 UNREAL ENGINE 4.....	30
4 METODOLOGIA	33
5 RESULTADOS.....	34
CONSIDERAÇÕES FINAIS	41
REFERÊNCIAS BIBLIOGRÁFICAS	42

RESUMO

Os dispositivos com acesso à internet estão presentes entre alunos e professores, portanto torna-se cada dia mais importante utilizar esses recursos tecnológicos a favor do ensino aprendizagem. Neste intuito o trabalho buscou desenvolver uma ferramenta case que permitirá a classificação de aplicativos segundo os níveis de conhecimento da Taxonomia de Bloom. Toda pesquisa foi baseada em estudo bibliográfico onde há relatos de professores que tem utilizado tecnologias em sala de aula, porém os mesmos possuem dificuldades de mensurar o que foi aplicado, pois a maioria dos softwares existentes não utilizam de metodologias classificadoras ou mesmo de uma metodologia como a Taxonomia de Bloom para verificar o nível de aprendizado de suas turmas. A partir dessas premissas desenvolvemos a ferramenta proposta além de um jogo de quiz que servirá de exemplo e que estará classificado na guia de memorização da ferramenta.

Palavras-chaves: Tecnologia na educação. Metodologias de ensino, Taxonomia de Bloom.

ABSTRACT

Devices with Internet access are available to students and teachers, more easily for each day more important using these technological resources in favor of teaching learning. In this purpose the work sought to develop a database of the class of applications according to the knowledge levels of the Bloom Taxonomy. Total research in a bibliographic study where there are reports of teachers who have used technologies in the classroom, but they have difficulties to measure what was applied, since most of the existing software are not classifying methodologies or even a methodology as a Taxonomy of Bloom to check the learning of their classes. Based on assumptions developed, the proposed test kit is an example and is available in the tool memorization guide.

Keywords: Technology in education. Teaching methodologies, Bloom Taxonomy.

INTRODUÇÃO

A tecnologia tem atuado como parte integrante do processo de ensino-aprendizagem e está sendo cada vez mais utilizada nas instituições de ensino. Em pesquisa realizada por Bianca (2017) mostra que 85% dos alunos do ensino fundamental e médio utilizam a internet em sala de aula por meio de dispositivos móveis, os quais 52% a usam para realizar suas atividades escolares. Essa pesquisa ainda aponta que 49% dos professores declararam que fazem o uso destes dispositivos para atividades em sala de aula. Segundo a autora da pesquisa, a parcela de professores que usam a internet como ferramenta de ensino teve um aumento significativo desde 2015, ano que era de apenas de 39%.

De modo geral os recursos tecnológicos já têm sido utilizados em salas de aula a bastante tempo, como exemplo os *datashows*, lousas digitais, dentre outros. Atualmente a tecnologia tem sido aplicada em sala de aula diretamente no processo de ensino-aprendizado em diversas formas, tais como, jogos educacionais e realidade virtual segundo o site Blackboard (2017) “O uso de recursos tecnológicos vem ganhando cada vez mais espaço nas escolas.”. Até a Google, uma grande empresa do ramo tecnológico já vislumbra que a tecnologia é um grande aliado da educação, a empresa desenvolveu aplicativos para fins educacional, como por exemplo o Google for *Education*.

Por outro lado, um fator prejudicial da utilização dessas tecnologias é a dispersão da atenção dos alunos causada pelo excesso de informação instantânea, como é relatado por Ramos (2012, p.3) “Os aparelhos eletrônicos em sala de aula são um convite à distração, durante as aulas, utilizados em excesso por muitos alunos e muitas vezes prejudicam o aprendizado”.

Por fim, outro aspecto a ser considerado na utilização dessas tecnologias é o objetivo de sua aplicação, Ferreira, Ventura (2007, p.8) afirmam que “corre-se o risco de utilizar os recursos computacionais como simples substituições do giz e do quadro negro, tornando-os uma extensão de um modelo tradicional”. Pois muitas das vezes essas tecnologias são aplicadas contrariando o seu real propósito pedagógico, por falta de utilização de uma metodologia bem definida ou mesmo um guia para auxiliar no processo de aquisição do conhecimento.

Neste sentido, este trabalho propõe o desenvolvimento de um sistema que permitirá a classificação dos jogos inseridos de acordo com os níveis de conhecimentos da Taxonomia de Bloom. Dando oportunidade ao professor de medir o nível de conhecimento de seus alunos seguindo as classificações especificadas por essa metodologia.

1 OBJETIVOS

1.1 OBJETIVO GERAL

Desenvolver uma ferramenta *case* que possibilite classificar soluções tecnológicas ou aplicativos conforme os níveis de conhecimentos da Taxonomia de Bloom, permitindo assim um apoio pedagógico para os professores.

1.2 OBJETIVOS ESPECÍFICOS

- Definir as tecnologias que serão utilizadas para desenvolver a plataforma e o jogo;
- Estabelecer as etapas de desenvolvimento para a plataforma;
- Utilização da Taxonomia de Bloom para a classificação dos jogos inseridos dentro da ferramenta *case*;
- Definir uma metodologia ágil que permita apoiar o desenvolvimento do projeto;

2 JUSTIFICATIVA

Dado o avanço tecnológico e a inclusão cada vez maior de educadores e alunos no meio digital, surge a necessidade de criar novos meios de interagir com as tecnologias dentro do ambiente educacional. Além disso, as instituições de ensino precisam acompanhar esses processos, utilizando-se de suportes tecnológicos que auxiliem tanto os discentes quanto os docentes na construção do conhecimento.

O uso da tecnologia pode ser aprimorado com a utilização de recursos interativos, tornando os conteúdos abordados em sala mais atrativos e fazendo com que os alunos adotem uma postura mais participativa. Lopes (2002, p.3) relata que “a informática deve habilitar e dar oportunidade ao aluno de adquirir novos conhecimentos, facilitar o processo ensino-aprendizagem, enfim ser um complemento de conteúdos curriculares visando o desenvolvimento integral do indivíduo”.

Sabemos atualmente que boa parte de alunos e professores já utilizam algum tipo de tecnologia em sala de aula, com acesso à internet, por estarmos na era da tecnologia, onde grande parte dessas informações estão disponíveis na palma da mão. Essa tecnologia unificada com a prática pedagógica possibilitará o desenvolvimento de uma ferramenta tecnológica para professores e alunos com intuito de melhorar processo de ensino-aprendizagem.

É necessário o estudo tecnológico e pedagógico para o desenvolvimento de uma ferramenta case que busque auxiliar os docentes em suas atividades de ensino possibilitando verificar o nível de conhecimento de seus alunos.

O desenvolvimento desta ferramenta terá como apoio pedagógico a Taxonomia de Bloom, pois com ela o sistema desenvolvido permitirá a classificação de jogos dando suporte ao professor a trabalhar os níveis de conhecimentos da taxonomia, levando o conteúdo de uma forma dinâmica através do uso de jogos que proporcionem aos discentes uma melhor absorção do conteúdo e para o docente uma possibilidade de avaliar cada um de seus alunos ou sua turma.

3 REFERENCIAL TEÓRICO

3.1 SISTEMA DE ENSINO-APRENDIZAGEM

3.1.1 DESAFIOS DA EDUCAÇÃO

A forma tradicional em que as instituições trabalham o ensino-aprendizagem está em conflito com o cenário atual, no qual alunos recebem informações de diversas formas e lugares isso se dá devido ao acesso à tecnologia "internet das coisas, redes sociais, blogs, etc." Essa realidade tem transformado a forma e as didáticas que os docentes trabalhem em suas salas de aula, e com isso a escola tem a possibilidade de auxiliar os alunos na interpretação desses conteúdos e na resolução de problemas para que o aprendizado passe a ser dinâmico.

A escola precisa exercer um novo papel em uma realidade radicalmente transformada. Em vez de doar conteúdos, passa a instigar a curiosidade do aluno e sua capacidade de aprender, orienta a busca de soluções para problemas reais, propõe a construção de valores éticos, formula questões ético cognitivas para reflexão. As disciplinas fragmentadas dão lugar aos problemas transdisciplinares, em que todos os professores orientam na busca de informação relevante para apoiar soluções. Os temas são questões relevantes da realidade, como água, alimentação, transporte, energia, sustentabilidade do planeta, manipulação genética, nanotecnologia e tantos outros. Na escola, deve-se aprender como determinar e definir um problema, para resolvê-lo, sabendo trabalhar em times para a busca de soluções (ABREU, 2013, p.142).

O Ministério das Comunicações entendeu a necessidade da implantação de ferramentas tecnológicas para o desenvolvimento da educação, permitindo que os próprios alunos auxiliem no processo, o MEC também disponibiliza de um guia sobre as tecnologias que podem ser utilizadas nas escolas. Segundo o Guia de Tecnologias Educacionais/MEC (2008), a metodologia permite que alunos com maior domínio na utilização do software passem a atuar no processo, colaborando em atividades na própria escola.

Para que isso aconteça, é necessário que as escolas e os docentes estejam preparadas para lidarem com essas tecnologias, grande parte das instituições de ensino do país não estão preparadas e não possuem estruturas suficientes para acompanharem os avanços digitais. Precisa-se motivar os profissionais da educação para o futuro, mas o problema do despreparo é comum, como mostra VALLE onde:

A inclusão digital dos profissionais da educação no município de Poços de Caldas, cujo principal objetivo era analisar como o professor percebe o desenvolvimento das tecnologias existentes e os impactos e as possibilidades de sua utilização no contexto educativo. Para isso, analisaram-se, de forma qualitativa, 108 questionários, em que

educadores responderam questões relacionadas a facilidades, dificuldades, perspectivas e receios quanto à aplicação de tecnologias computacionais na educação. Os resultados dessa análise apontam que, embora os educadores confirmem a presença das tecnologias como algo benéfico ao processo ensino-aprendizagem, associando essa ideia à facilidade e à perspectiva, paradoxalmente, eles não as utilizam em suas práticas pedagógicas (VALLE, 2013, p.57).

No cenário citado acima é notório que em muitas instituições de ensino exista dispositivos tecnológicos para tutores e alunos utilizarem agregado no processo de aprendizagem essas ferramentas tem a possibilidade de se adequar algum processo pedagógico seja o construtivismo ou a Taxonomia de Bloom.

3.1.2 TAXONOMIA DE BLOOM

A Taxonomia de Bloom teve sua publicação em 1956, com a liderança do americano Dr. Benjamin Samuel Bloom, com base na educação americana e tendo como objetivo o desenvolvimento de um sistema dividido em categorias de aprendizagem do comportamento conforme a Figura 1 como forma de auxiliar na concepção e avaliação da aprendizagem educacional.

Essa taxonomia vem sendo expandida desde então por outros colaboradores, onde em 2001 houve uma remodelagem da estrutura da taxonomia visando uma melhor adaptação da mesma com o novo modelo de ensino.

Essa ferramenta pedagógica também serve como uma espécie de *checklist* visando assegurar que todo o processo está sendo seguido corretamente.

Também oferece uma estrutura em forma de camadas divididas em categorias conforme a Figura 2, essas categorias são divididas em:

- **Lembrar:** É a habilidade de lembrar informações e conteúdos previamente abordados.
- **Entender:** Habilidade de compreender e dar significado ao conteúdo, entender a informação e de utilizá-la em contextos diferentes.
- **Aplicar:** Habilidade de usar informações, métodos e conteúdos aprendidos em novas situações concretas.
- **Analisar:** Habilidade de subdividir o conteúdo em partes menores com a finalidade de entender a estrutura final. Identificar partes e suas inter-relações.
- **Sintetizar:** Habilidade de agregar e juntar partes com a finalidade de criar um novo todo.

- **Criar:** Habilidade de julgar o valor do material (proposta, pesquisa, projeto) para um propósito específico.

Cada uma dessas etapas possui suas ponderações e objetivos a serem seguidos, do nível mais simples ao mais complexo, onde para que o aluno venha a avançar para o próximo nível ele possua domínio sobre os níveis anteriores.

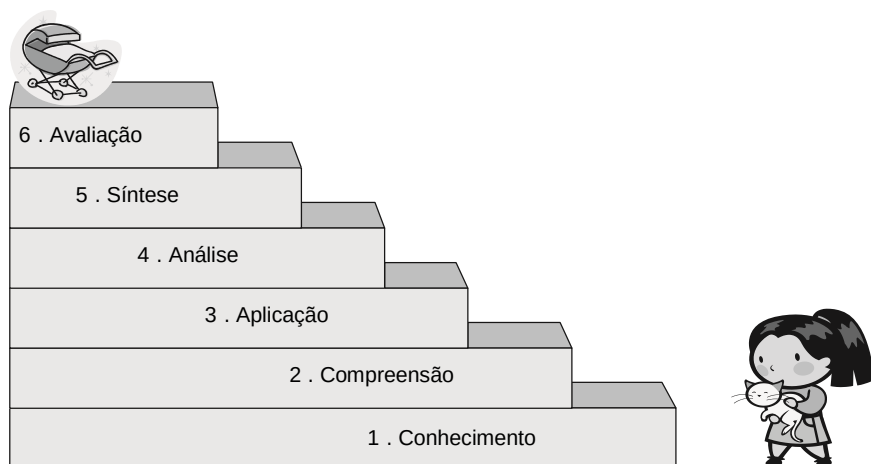


Figura 1 - Níveis de conhecimento da Taxonomia de Bloom publicada em 1956.
Fonte: Scielo Brasil.

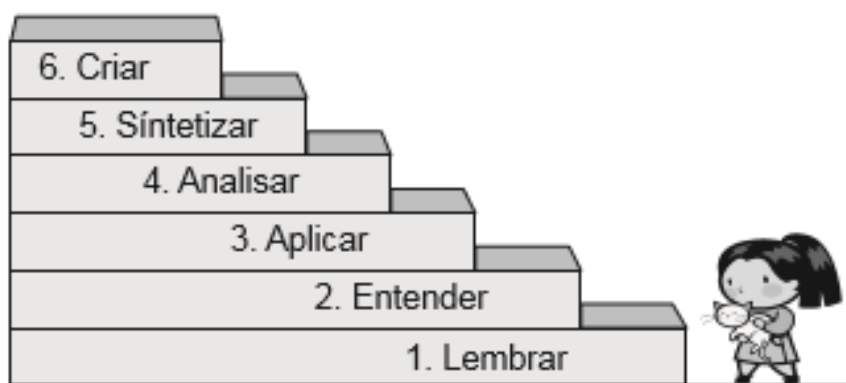


Figura 2 - Novo modelo da Taxonomia de Bloom proposta por Anderson em 2001.
Fonte: Scielo Brasil.

Muitas ferramentas educacionais estão sendo desenvolvidas nos últimos anos, e grande parte dessas ferramentas estão disponíveis nas lojas de aplicativos e muitas delas não possui um auxílio pedagógico e devido a essa estrutura utilizada pela Taxonomia de Bloom ela terá um papel importante para apoiar o desenvolvimento de uma tecnologia para auxiliar o

professor e os acadêmicos no processo de aprendizagem onde é relevante observar que o uso de tecnologias em práticas pedagógicas vem aumentando.

3.2 TECNOLOGIAS DE DESENVOLVIMENTO

Para o desenvolvimento da ferramenta case proposta e do jogo exemplo da aplicação, se faz necessário a utilização de tecnologias como: linguagens de programação, *framework*, *engines*, dentre outros, onde cada um deles terão um papel fundamental para que a ferramenta seja criada.

3.2.1 HTML

Com o surgimento da internet foi necessária a criação de uma forma de linguagem que fosse interpretado por diferentes meios e diversos tipos de acessos, no início da década de 80 Tim Berners-Lee propôs a criação de uma linguagem de marcação que contemplaria esses benefício, o HTML, após vários anos de pesquisas e melhorias a linguagem finalmente foi finalizada em 1990 e sofreu inúmeras revisões e alterações buscando sua melhoria, seu verdadeiro sucesso aconteceu quando suas próximas versões foram apresentadas em uma conferência mundial sobre a rede de comunicação de computadores.

Essas séries de mudanças foram de extrema importância para o crescimento e aceitação do HTML, a linguagem está atualmente em sua versão 5 que houve uma série de novas atualizações para sua melhoria:

O HTML 5 é a nova versão do HTML 4 e um dos seus principais objetivos é facilitar a manipulação dos elementos, possibilitando o desenvolvedor modificar as características dos objetos de forma não intrusiva, fazendo com que isso fique transparente para o usuário final (FEITOSA, 2012, n.p).

Sua nova versão é compatível com diversas plataformas como por exemplo: *browsers* (navegadores), *smartphones*, *tablets*, *smart TVs* e através de qualquer dispositivo que esteja conectado à rede mundial de computadores, tornando se uma linguagem prática e de fácil implementação encurtando o tempo de desenvolvimento de novas aplicações.

3.2.2 CSS

As primeiras versões do HTML não contavam com interfaces amigáveis para seus usuários, esse foi um dos motivos para o desenvolvimento do CSS, uma folha de estilo

composta por camadas e utilizada para criar a camada de apresentação (aparência) em páginas WEB desenvolvidas com linguagens de marcação como por exemplo o XML e HTML, sua principal função é definir como os componentes serão exibidos aos usuários através de elementos como cores, formatações de textos, fontes, imagens, entre outros proporcionando uma maior flexibilidade e evitando a repetição do conteúdo nas estruturas das páginas, atualmente sua versão 3 trouxe novas capacidades de formatações e recursos que auxiliam a criar interfaces ainda mais criativas e amigáveis.

3.2.3 JAVA SCRIPT

O Java Script foi criado em 1995 e foi originalmente desenvolvido sob o nome de Mocha e posteriormente teve seu nome modificado para Live Script e por fim, Java Script, é uma linguagem de programação orientada a objetos que é processada do lado do cliente trazendo mais praticidade e agilidade além de também deixar as páginas WEB mais dinâmicas, segundo Leonardo Balter engenheiro de software especialista em Java Script:

O Java Script tem um papel fundamental na evolução da WEB, pois a ideia central da Internet era ser uma plataforma aberta para todos. Foi superimportante que a sua principal linguagem de programação também fosse aberta, sem controle centralizado de corporações e de fácil aprendizado (MUSTAFA, 2016, n.p).

É uma linguagem de *script* o que significa que ela possibilita ao desenvolvedor a controlar funções de aplicações que foram criadas por terceiros, como por exemplo controlar comportamentos de navegadores por meio de um código enviados para uma página em HTML, esse tipo de linguagem é interpretada sendo assim, não dependem de um processo para que os códigos gerados sejam compilados. Os códigos são interpretados linha a linha enquanto são lidos pelo navegador. Sua principal característica é prover páginas dinâmicas do lado do cliente.

Com o grande sucesso da linguagem houve a necessidade de se implementar novas melhorias, hoje, o Java Script não é utilizado apenas para o desenvolvimento WEB, mas também está presente em aplicativos para *smartphones*, programas para desktop, dentre outros.

3.2.4 ANGULAR 4

Criado em 2009 o Angular Js é um o *framework* baseado em Java Script e que possui como objetivo facilitar a criação de aplicações web. Esse *framework* adapta o HTML tradicional, tornando a experiência do usuário mais dinâmica. Segundo Waltenberg (2016),

professor *front-end* da Alga Works, a filosofia do Angular Js “parte de que uma programação declarativa é muito mais importante que uma programação imperativa quando se trata de desenvolvimento web. Ele atinge isso estendendo o HTML e fazendo uma linguagem para o desenvolvimento de interfaces web dinâmicas”.

Atualmente o Angular é mantido pela Google e se encontra em sua versão 4 além de possuir uma comunidade sólida que auxilia na manutenção e na melhoria da linguagem. Uma das características que diferem o Angular de outros *frameworks* Java Script segundo Valente (2014) “é que ele funciona como uma extensão do HTML, diferente de outros *frameworks* Java Script, não é necessário manipular o DOM. Isso possibilita a criação de um *front-end* mais organizado”.

Uma das características que fez do Angular um *framework* tão popular é o *Two-Way Data Binding*, que é responsável pela sincronização automática das alterações dos dados realizados entre a *view* e o *model* e do *model* para a *view* conforme a Figura 3.

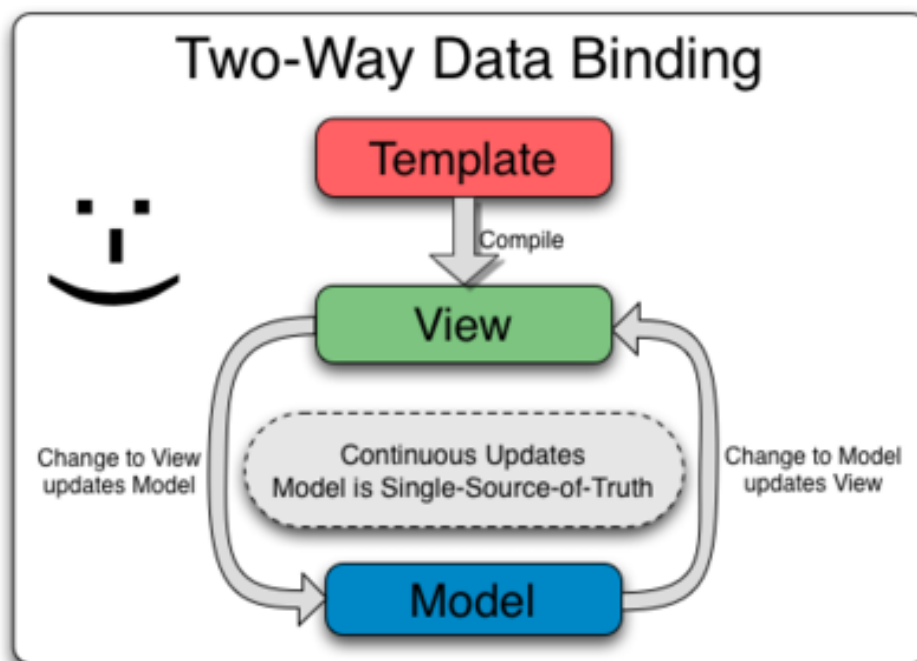


Figura 3 - Estrutura de comunicação entre *view* e *model* no Angular Js.

Fonte: Angular Js

3.2.5 VUE JS

Outro *framework* baseado em JS é o VUE JS que segundo Reis (2016) “ é uma biblioteca (lib) Java Script para o desenvolvimento de componentes reativos para interfaces web modernas”.

A utilização de componentes traz benefícios, como a reutilização de código por não ser necessário recriar trechos repetitivos. Em artigo publicado no site Vue JS Brasil, Reis (2016) afirma que “inicialmente se assume que ao criar um componente você está criando ele para reuso, mas nem todo componente é para reuso. Um componente pode existir apenas para compor uma parte de outro componente por exemplo”.

Outra característica do VUE é sua técnica de observação de objetos no DOM que faz com que essas alterações sejam refletidas nas páginas HTML conforme a Figura 4.

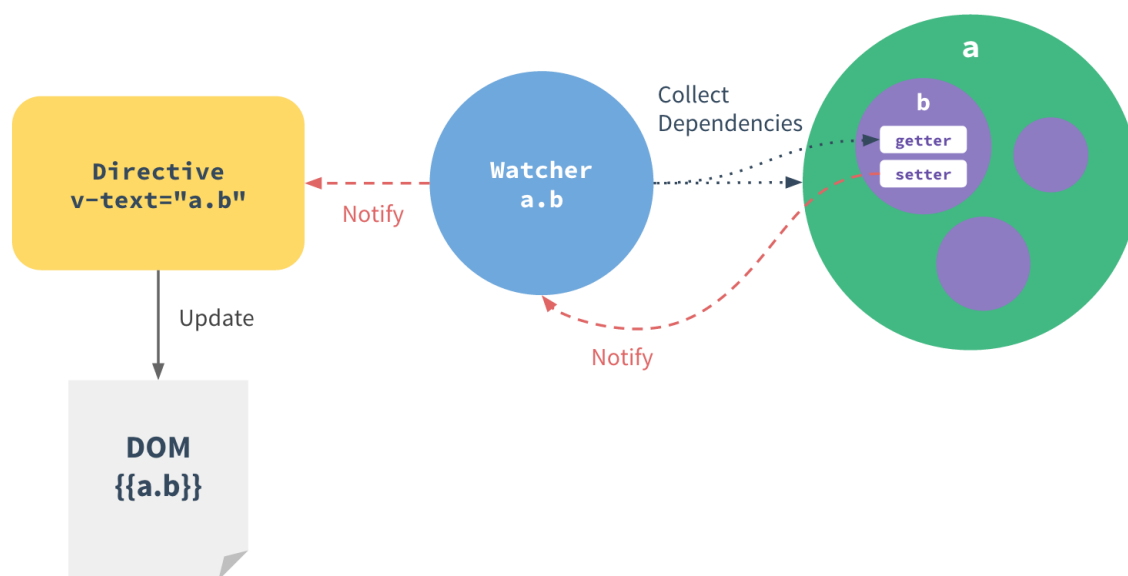


Figura 4 - Estrutura de comunicação reativa do Vue JS.

Fonte: Vue Js Brasil

3.2.6 PHP

Segundo o site da própria desenvolvedora The PHP Group (2017) o PHP (*Hypertext Preprocessor*) “é uma linguagem de script *open source*¹ de uso geral, muito utilizada, e especialmente adequada para o desenvolvimento web e que pode ser embutida dentro do HTML, criada em 1994 por Rasmus Lerdo”.

O PHP assim como qualquer outra linguagem possui suas vantagens e desvantagens é o que destaca Araújo (2009) “O PHP tem algumas vantagens muito interessantes: fácil aprendizado, velocidade, robustez, multiplataformas, e código-fonte aberto. Desvantagens do PHP: compatibilidade entre versões, documentação incompleta e suporte a datas. ”.

¹ Termo em inglês que significa código aberto, isso diz respeito ao código-fonte de um software, que pode ser adaptado para diferentes fins.

Com o crescimento da linguagem diversos *frameworks* foram desenvolvidos, segundo Leone (2016) “os 5 principais frameworks PHP que existem no mercado são: Zend Framework, Phalcon, Symfony, Cake PHP e Laravel”.

Sua versão 7.0 lançada em 2017 trouxe uma série de melhorias e alterações conforme a The PHP Group (2017) “essa versão foca principalmente na remoção de funcionalidades depreciadas em versões anteriores e na melhoria da consistência da linguagem”.

3.2.7 ZEND FRAMEWORK

O Zend é um robusto *framework* PHP, segundo Martini (2006) “o Zend Framework foi lançado no dia 04 de março de 2006 e inclui diferentes componentes desenvolvidos em PHP5 para prover alta qualidade para desenvolvimento de aplicações web e web *services*. ”.

Algumas das características desse *framework* é o modelo arquitetural MVC conforme a Figura 5, esse padrão visa separa a aplicação em camadas, onde cada uma delas possuem suas funções específicas mantendo a organização dos projetos, também possui suporte a uma variada gama de sistemas de bancos de dados, dentre outros.

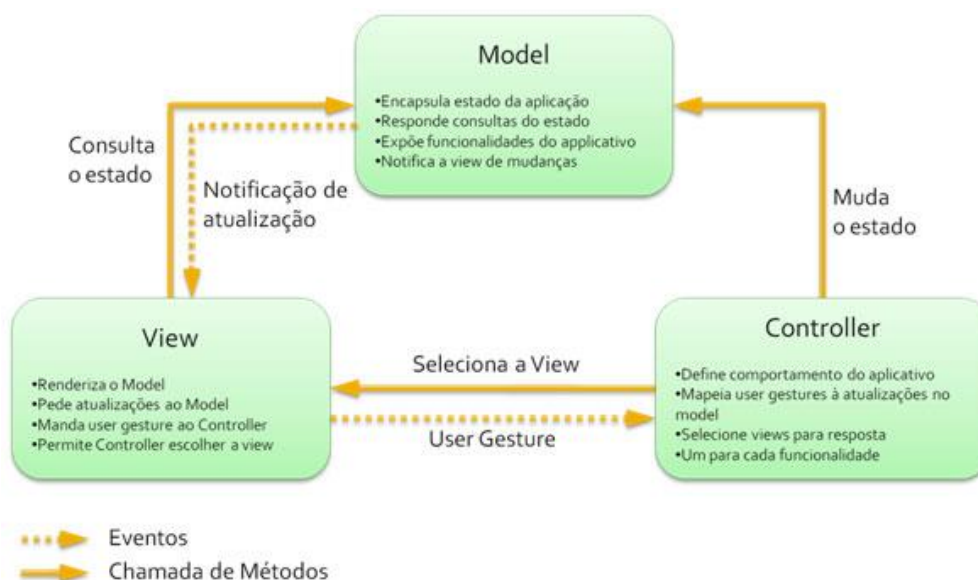


Figura 5 - Modelo arquitetural MVC.

Fonte: Master Info

3.2.8 LARAVEL

É um *framework* PHP para desenvolvimento de aplicações WEB, que utiliza a arquitetura MVC e segundo Adriel (2017) o Laravel “tem como principal característica ajudar a desenvolver aplicações seguras e performáticas de forma rápida, com código limpo e simples, já que ele incentiva o uso de boas práticas de programação e utiliza o padrão PSR-2 como guia para estilo de escrita do código.”.

O Laravel possui um *template* para criação de interfaces gráficas que lhe permite a criação de interfaces mais limpas e de forma rápida, esse *template* evita a repetição de códigos por permitir que o desenvolvedor possa utilizar de *templates* já desenvolvidos para compor as novas páginas criadas.

Já na comunicação com bancos de dados o Laravel utiliza um *Active Records*² chamada de Eloquent ORM³, segundo Reis (2017) o Eloquent “é uma implementação de *Active Records* muito simples e que deixa o seu código muito mais bonito e gostoso de trabalhar.”

3.2.9 LUMEN

O Lumen é um micro *framework* PHP que tem como base o Laravel, segundo Ferreira (2015) esse *framework* “foi feito para micro serviços, e não tanto para aplicações *front-end* que se valem de interações com o usuário (embora ele possa ser utilizado para qualquer coisa)”.

Por ter sido desenvolvido à partir da base do Laravel, sua documentação é resumida, o *framework* também sacrifica, recursos e automações, visando entregar um melhor desempenho, maior e mais eficiente, que a versão completa, porém, alguns recursos ainda estão presentes e podem ser utilizados facilmente habilitando seus recursos é o que diz Willians (2015) “a maior sacada que o Lumen teve, é que de fato ele criou uma ótima estrutura organizacional que permite você ir adicionando recursos do Laravel, apenas descomentando algumas linhas e sair utilizando.”

Com isso o Lumen se torna uma boa opção quando o assunto é micro serviços pela praticidade que a própria arquitetura do *framework* fornece aos desenvolvedores.

² Padrão de projeto presentes em softwares que armazenam seus dados em banco de dados relacionais.

³ Técnica de desenvolvimento de software que serve para o uso de tabelas de bancos de dados em formato de objetos relacionais, facilitando assim a manutenção de dados.

3.3 GAMEFICAÇÃO

O uso de conceitos para a aplicação de mecânicas de jogos, nas mais diversas áreas como educação, negócios, saúde, dentre outras se dá o nome de gameficação. Segundo VASCONCELLOS o principal objetivo é “aumentar o engajamento e despertar a curiosidade dos usuários e, além dos desafios propostos nos jogos, na gameficação as recompensas também são itens cruciais para o sucesso”.

Um dos objetivos de se utilizar desse conceito é empregar, comprometer além de recompensar os jogadores, o uso de recompensas aos usuários que realizam as atividades dentro do jogo. Outro fator que levou a utilização desse conceito dentro do desenvolvimento do projeto foi a possibilidade de se medir o desempenho de forma geral e individual dos jogadores, podendo ser convertidos em relatórios.

3.4 GAME ENGINE

Uma *game engine* é um conjunto de ferramentas e bibliotecas capazes de construir elementos de um jogo em tempo real, composta por motores gráficos que são utilizados para realizar as renderizações necessárias, tanto em gráficos 2D quanto para aqueles mais complexos como jogos 3D, as *engines* também são compostas por várias outras partes como por exemplo:

- Motores de Física;
- Suporte para áudios;
- Suporte para detecção de colisões;
- Dentre outros.

Por esses motivos as *engines* facilitam a criação de jogos pois trazem a praticidade aos desenvolvedores, fazendo com que os projetos sejam finalizados em um período menor de tempo pois não há necessidade de criar um jogo totalmente do zero. Atualmente existem várias *engines* disponíveis que vão desde as mais simples como o Construct 2, RPG Maker e o GameMaker até os mais avançados como a Unity 3D e o Unreal Engine 4.

3.4.1 CONSTRUCT 2

O Construct 2 é uma ferramenta desenvolvida pela Scirra em meados de 2013 especificamente para criação de jogos 2D baseados em HTML 5 e Java Script.

Um dos diferenciais da ferramenta é a não necessidade de um conhecimento prévio em desenvolvimento de jogo ou em alguma linguagem de programação, já que toda a lógica de programação é realizada através de eventos dentro da própria ferramenta.

O Construct 2 possui suporte a diversas plataformas como o iOS, Android, Windows Phone, Desktop, dentre outras plataformas conforme a Figura 6 possibilitando um desenvolvimento rápido e prático de jogos, pelo fato de não haver a necessidade de se criar uma programação específica para cada uma das plataformas.

A ferramenta também possui algumas limitações na quantidade de componentes, eventos e efeitos sonoros e visuais que podem ser utilizados em sua versão gratuita, já em suas versões *business* e *personal* se diferenciam no valor da receita que o desenvolvedor pode adquirir com o jogo.

Sua programação orientada a eventos possibilita que os desenvolvedores possam criar mesmo quando não possuem um conhecimento em programação, cada ação realizada será baseada em eventos que serão disparados pelo jogador ao interagir com as funcionalidades do jogo.

Essa ferramenta também possui comportamentos, eventos e diversas extensões como estilos e efeitos que facilitam o processo criação dos mais variados tipos de jogos 2D conforme ilustrado na Figura 7.

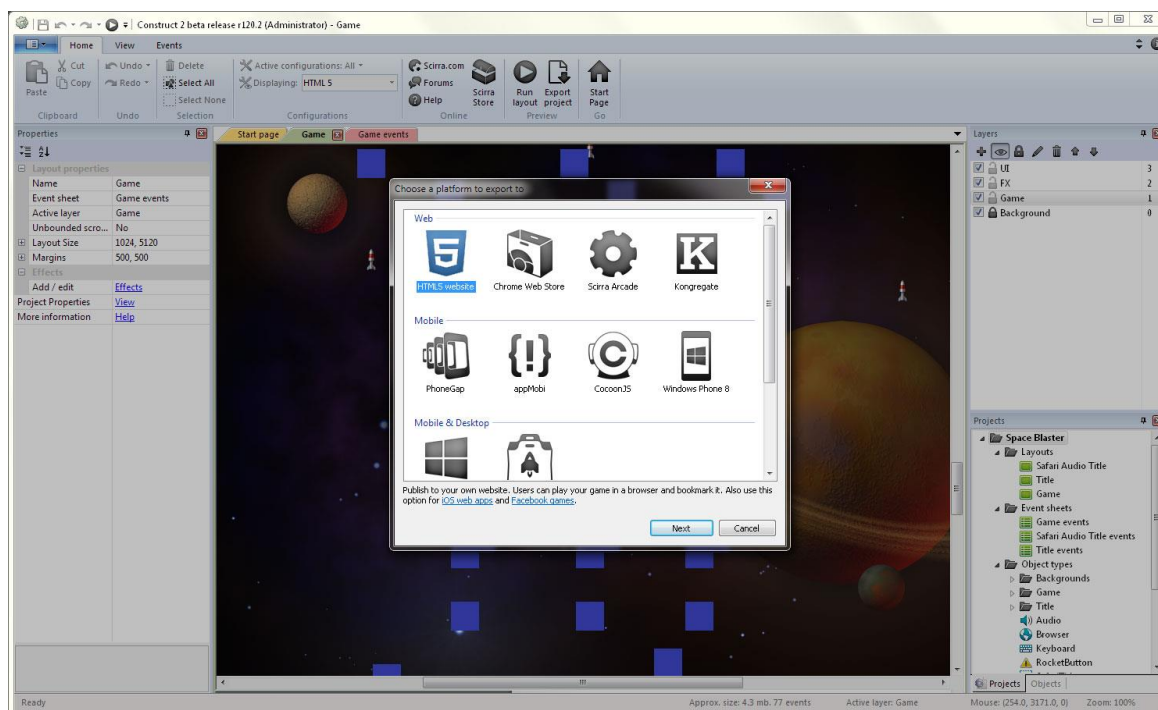


Figura 6 - Opções de exportação para jogos criados no Construct 2.
Fonte: Scirra – Construct 2

Para aqueles que possuem um conhecimento mais avançado, a ferramenta também fornece suporte para o desenvolvimento de jogos multijogadores, onde a própria Scirra fornece servidores básicos e gratuitos para aqueles que desejam iniciar o desenvolvimento desse estilo de jogo.

Sua comunidade em expansão colabora para o crescimento e divulgação da ferramenta, fazendo com que suas atualizações e melhorias sejam constantes.

A Scirra anunciou em fevereiro de 2017 a chegada do Construct 3 com uma série de inovações que irão permitir a criação de jogos direto no navegador, a nova ferramenta também possui um *layout* totalmente modificado conforme a Figura 8 além de contar com um suporte para armazenamento em nuvem fazendo com que o desenvolvedor possa criar seus jogos independentemente de onde esteja, a nova versão também possui a sincronização automática dos projetos em nuvem, no caso de não haver conexões de internet, dentre outras características.

Uma das limitações dessa ferramenta é o fato de não possuir suporte para a criação de jogos mais complexos como por exemplo jogos em 3D ou em jogos que necessitem de uma lógica muito complexa podem não ter um desempenho satisfatório nesta ferramenta.

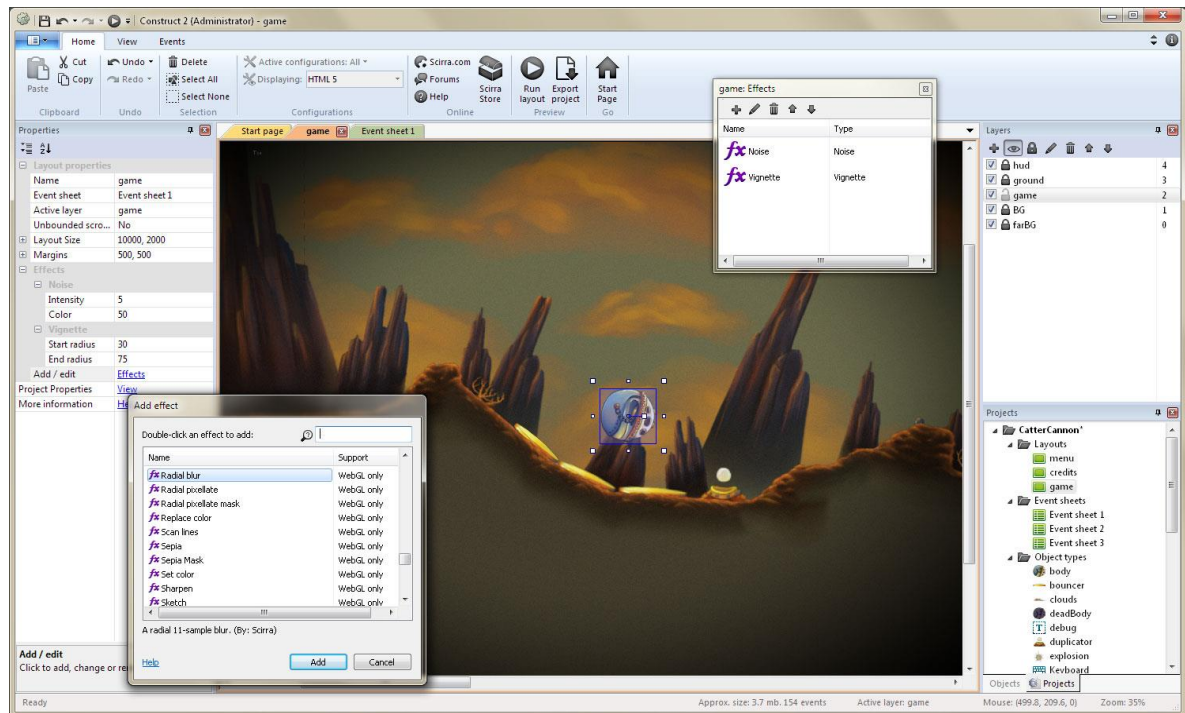


Figura 7 - Processo de adição de efeitos em um objeto do cenário.
Fonte: Scirra – Construct 2

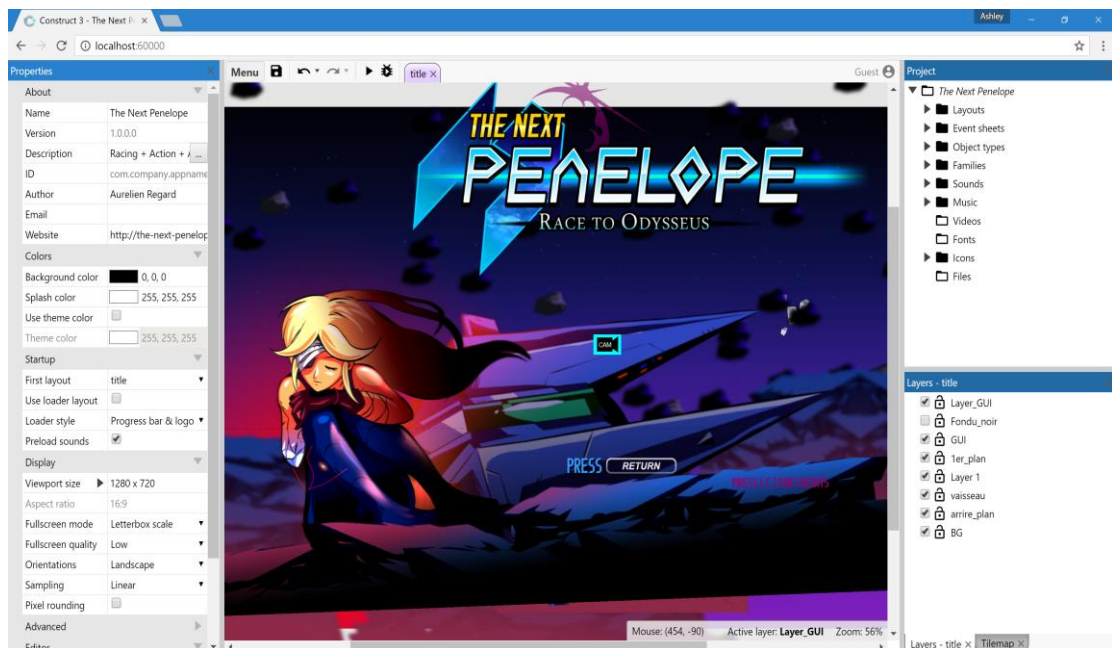


Figura 8 - Novo layout de desenvolvimento do Construct 3.
Fonte: Scirra – Construct 3

3.4.2 RPG MAKER

O RPG Maker consiste em um conjunto de diversas *engines* que possibilitam a criação de jogos, foi lançado em meados de 1997 pela empresa ASCII e atualmente pertence a uma

empresa associada a Enterbrain, uma editora de revista também conhecida por produzir outras *engines* para o desenvolvimento de jogos como o Game Maker e Shooter Maker.

Essa ferramenta dispõe de uma série de objetos conforme a Figura 9 onde o desenvolvedor pode usar da criatividade para montar seus próprios cenários, também podem haver a inserções de novos objetos para compor a cena, cada um dos objetos inseridos pode ser configurado de maneira individual para atender às necessidades do desenvolvedor.

Essa *engine* é destinada para o desenvolvimento de jogos no estilo RPG, em suas versões mais antigas somente era possível que os jogos fossem exportados para a plataforma Windows, já sua versão mais recente lançada em 2015 a RPG Maker MV utiliza-se Java Script juntamente com HTML 5, com isso permite a exportação dos jogos para uma gama maior de plataformas como iOS, Android, dentre outros conforme a Figura 10.

Para desenvolvedores mais experientes a nova versão da ferramenta possibilita a alteração e personalização dos elementos, possibilitando ao desenvolvedor possuir mais controle sobre o jogo a ser desenvolvido.



Figura 9 - Interface de desenvolvimento do RPG Maker VX ACE.
Fonte: RPG Maker Web.



Figura 10 - Suporte multiplataforma da versão RPG Maker MV.
Fonte: RPG Maker Web.

3.4.3 UNITY 3D

A Unity é uma das principais *engines* para criação de games, ela possibilita a criação de jogos mais complexos tanto em 2D quanto para 3D e utiliza de duas linguagens padrões para o desenvolvimento, o Java Script e o C# que é uma linguagem orientada a objetos desenvolvida pela Microsoft em meados de 1999 e que teve sua aprovação em 2001.

A ferramenta também possibilita a criação de jogos para uma grande quantidade de plataformas como por exemplo iOS, Android, Windows Phone, Windows, dentre outras sem que haja a necessidade de reescrever o código e também conta com um suporte oferecido tanto para a criação de jogos em 2D quanto para jogos mais complexos como aqueles desenvolvidos em 3D, a Unity também dispõe de vários tutoriais que permitem que os desenvolvedores possam como iniciar seus projetos e dar seus primeiros passos para a criação de jogos, atualmente a *engine* se encontra em sua versão 5 que conta com uma série de melhorias e para um melhor desempenho dos jogos, além de contar com uma interface simples e intuitiva conforme a Figura 11 que torna a ferramenta mais amigável aos usuários.

Para ter acesso a todas as funcionalidades que essa ferramenta pode fornecer é necessário por sua vez obter a licença de desenvolvimento, já que a versão gratuita possui uma série de limitações como quantidades de plataformas disponíveis para a exportação dos projetos, filtros de áudio e outras opções importantes como por exemplo informações sobre a performance do jogo. Assim como em outras *engines* a Unity também possui funções já

carregadas e também conta com uma série de texturas disponíveis pela própria desenvolvedora, alguns totalmente gratuitos.

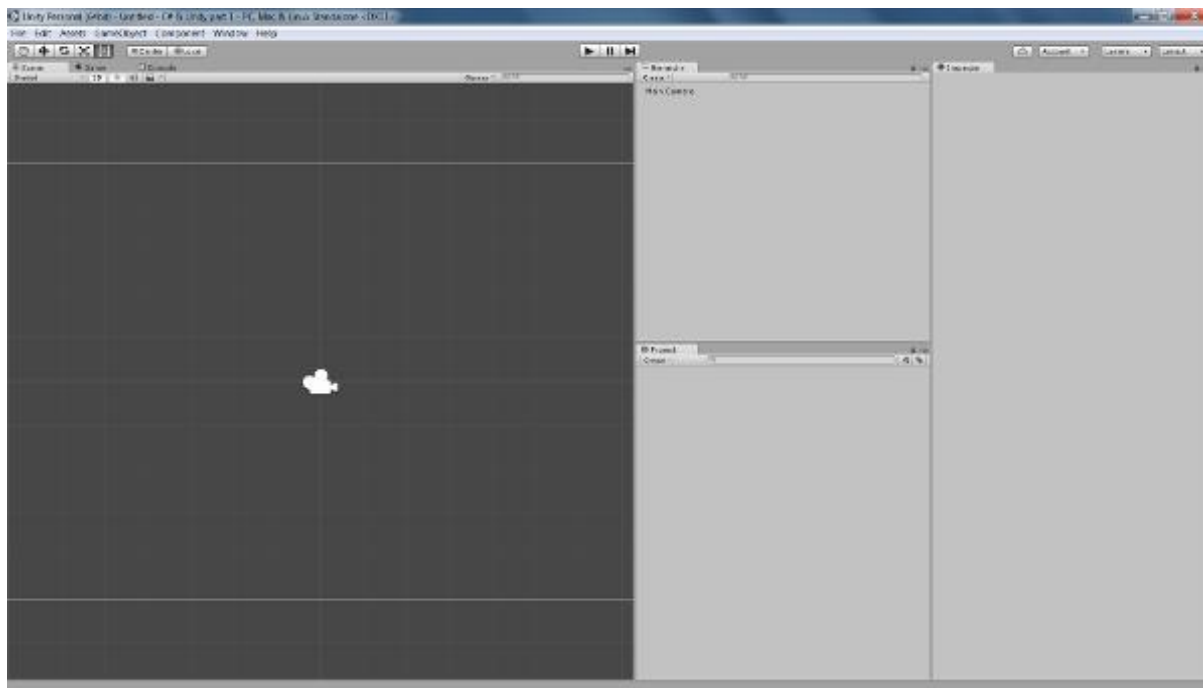


Figura 11 - Interface inicial da Unity 3D em sua versão 5.

Fonte: Game Code School.

Ao contrário de ferramentas como o Construct 2 em que a programação é feita orientada a eventos a Unity 3D requer um nível de conhecimento mais avançado em programação, a codificação é feita manualmente embora alguns *scripts* de outros desenvolvedores ou até mesmo de outros projetos podem ser reutilizados e incluídos em seus respectivos componentes, onde o mesmo ficará responsável por executar as ações definidas pelo desenvolvedor conforme a Figura 12.

Todas as cenas desenvolvidas através dessa *engine* são baseados em *Game Objects*, podem ser exemplos de *Game Object* quaisquer elementos dentro do cenário do jogo através de um sistema de coordenadas no que está sendo desenvolvido como câmeras, luzes, partículas, dentre outros.

A Unity tem foco muito claro no desenvolvimento de jogos, embora essa ferramenta também possa ser utilizada para outros fins, segundo Machado (2016) a Unity “propõe a ser um modelo para a criação de jogos de aventura, como RPGs, FPSs e TPSs. Tudo isso está permeado por uma capacidade gráfica muito grande”.

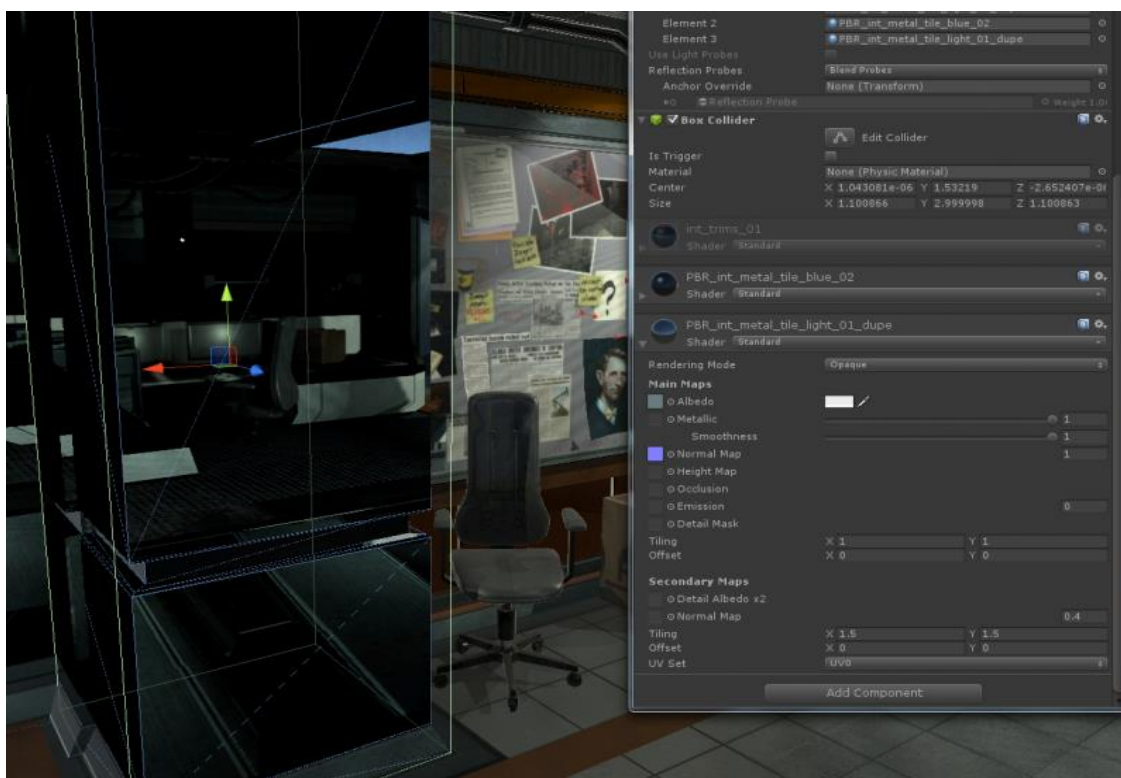


Figura 12 - Configuração das propriedades de um objeto do dentro da Unity 3D.
Fonte: Unity Blogs.

3.4.4 UNREAL ENGINE 4

Outro exemplo de ferramenta para o desenvolvimento de jogos é a Unreal Engine 4, desenvolvida em 2014 a nova versão da *engine* criada pela Epic Games.

A ferramenta conta com um desenvolvimento em através da linguagem C++ e um de seus principais diferenciais é o seu sistema de *Scripts* denominado *Blue Print* conforme a Figura 13. Esse sistema permite ao desenvolvedor montar e gerar protótipos para todas as mecânicas existentes no jogo, alterações de interface, tudo sem que haja a necessidade de programar qualquer linha de código para que as ações possam ser executadas. Também possui uma opção de *debug* onde o desenvolvedor pode efetuar os testes e visualizar o jogo em tempo real.

De uma maneira semelhante ao Construct 2 o *Blue Print* é programado baseado em eventos onde o desenvolvedor informa os passos que o objeto deverá realizar a partir de uma entrada do jogador. Na Figura 14 temos um simples exemplo de uma programação realizada no *Blue Print* e de seu funcionamento, onde o jogador ao realizar um a ação de *touch* na opção “Exit”, será disparado um evento que irá requisitar uma função que irá prover a saída do jogador e caso a ação seja executada na opção “Play” uma nova função é chamada para prover o carregamento de um novo nível para o jogador.

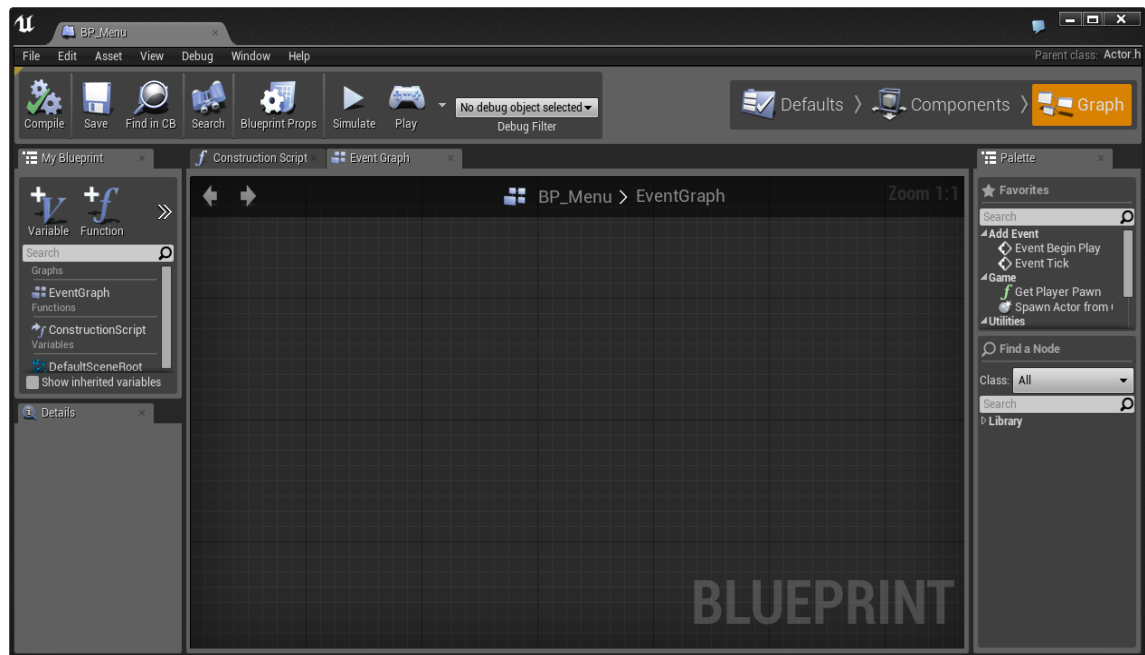


Figura 13 - Interface de programação do Blue Print.
Fonte: Unreal Brasil.

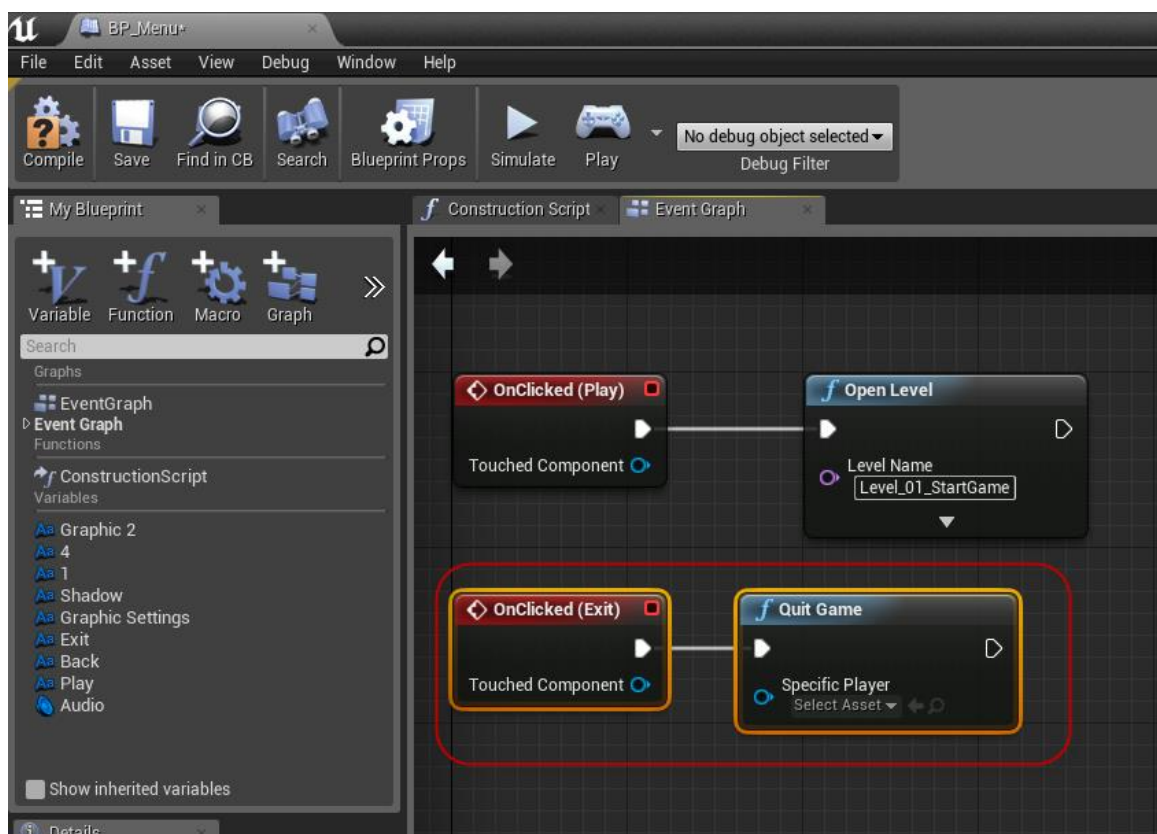


Figura 14 - Processo de programação orientada a eventos utilizando o Blue Print.
Fonte: Unreal Brasil.

Apesar das facilidades para o desenvolvimento a Unreal Engine 4 é uma ferramenta complexa, que suporta multiplataformas, em sua versão mais recente 4.16 a engine contempla

suporte a plataformas como Windows, Linux, Mac OS, além de possuir suporte a consoles como Xbox 360, Xbox One, PlayStation 4, Nitendo Switch dentre outros.

4 METODOLOGIA

O trabalho teve início a partir do levantamento bibliográfico a respeito da Taxonomia de Bloom, buscando entender seus objetivos, o motivo de sua criação, e como ela pode auxiliar na melhoria do ensino em sala de aula.

Em seguida serão realizados estudos sobre:

- O modelo atual de ensino;
- Como a tecnologia está sendo empregada nas escolas;
- Qual a estrutura que essas instituições possuem para que a tecnologia possa ser utilizada como forma de ensino;
- Os desafios enfrentados pelos professores ao fazerem o uso dessas tecnologias dentro de sala;

Após a etapa de levantamento de requisitos foram realizados estudos a respeito do tema de gameificação, como ela vem sendo utilizada na educação e através desse estudo foi possível compreender quais as melhores formas de se abordar os mais diversos conteúdos por meio da utilização dos jogos, tais como as mecânicas e as dinâmicas que serão utilizadas para melhor envolver os alunos nos temas propostos pelos docentes.

Também foram realizados estudos sobre ferramentas de desenvolvimento de jogos também conhecidas como *game engines*, através dessas ferramentas foram desenvolvidos juntamente com os conceitos de gameificação os jogos que compõem a ferramenta. Além do estudo de ferramentas para o desenvolvimento de jogos, também houveram estudos de diferentes *frameworks* para a criação da ferramenta proposta, buscando saber qual a melhor ferramenta a ser utilizada para o desenvolvimento da plataforma.

5 RESULTADOS

No desenvolvimento da ferramenta case proposta foram utilizados os *frameworks* Lumen para o *back-end* em sua versão 5.5, a mesma oferece suporte a micro serviços, além de contar com uma arquitetura MVC simplificada e organizada, durante o processo de desenvolvimento houve a necessidade de algumas adaptações na arquitetura do *framework* visando um melhor aproveitamento de suas funcionalidades e divisão de cada uma das camadas da aplicação.

Para o *front-end* foi utilizado o *framework* Angular em sua versão 4, que simplifica o processo de criação de páginas WEB e as tornam mais dinâmicas e sua capacidade de reutilização de código através da criação de componentes.

Para a parte visual da ferramenta foi utilizado um *framework* CSS, o Bootstrap, esse *framework* conta com uma diversidade de estilizações já existentes, facilitando a criação páginas da aplicação como afirma Barbieri (2017) “o Bootstrap oferece uma enorme variedade de *plugins* e temas. Além disto, possui integração com qualquer linguagem de programação. ”, na criação dos *layouts*⁴ foi utilizado como base a Google Play Store, uma plataforma virtual do Google para sistemas Android onde é possível encontrar jogos, filmes, músicas e livros, todos separados em suas respectivas categorias e classificações.

Na criação do jogo que serviu para exemplificar o funcionamento da ferramenta, foi utilizado como *game engine* o Construct 2 em sua *release*⁵ 247, essa *engine* nos permite utilizar de uma série de *behaviours* e efeitos pré-existentes, para manipulação dos objetos dentro do cenário em que o jogo será construído, além de possuir suporte a multiplataformas possibilitando uma integração com a ferramenta case proposta. Como exemplo foi desenvolvido um jogo em formato de quiz, onde as perguntas são carregadas aleatoriamente através de um arquivo XML, as demais funcionalidades do jogo foram criadas por meio de eventos, que são disparados pelo jogador à medida que o mesmo interage com os objetos no cenário.

Dentro da ferramenta, cada um dos jogos é classificado conforme sua categoria dentro dos níveis de conhecimento da Taxonomia de Bloom, que são: lembrar, entender, aplicar, analisar, sintetizar e criar, com base em sua reformulação de 2001 conforme a Figura 15. Dentro de cada uma das categorias há uma breve explicação a respeito do nível ao qual ela representa, os objetivos a serem alcançados e os jogos referentes ao nível em questão.

⁴ Esboço ou rascunho que mostra a estrutura física de uma página de um jornal, revista ou página na internet.

⁵ Nova versão de um software geralmente dividido em versões.

Na página inicial da aplicação os docentes podem ter acesso aos jogos já distribuídos e as datas de cada uma das distribuições. Já por parte dos alunos essa tela conta com a listagem de todos os jogos na qual ele tenha acesso, tanto para aqueles já finalizados, onde o mesmo pode ter acesso as suas pontuações e aqueles jogos que ainda se encontram dentro do prazo de distribuição delimitado pelo professor.

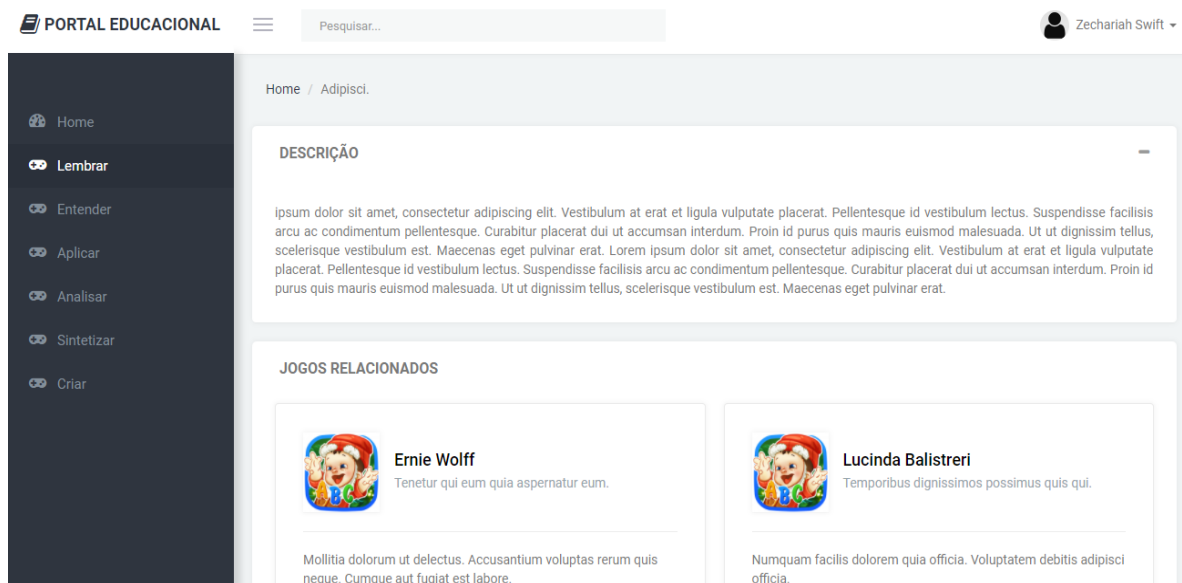


Figura 15 - Tela de descrição de um dos níveis da Taxonomia.

Fonte: Gabriel Neres, Tiago Neves

Os docentes podem distribuir os jogos existentes na plataforma aos alunos por um período estimado de tempo conforme a Figura 16.

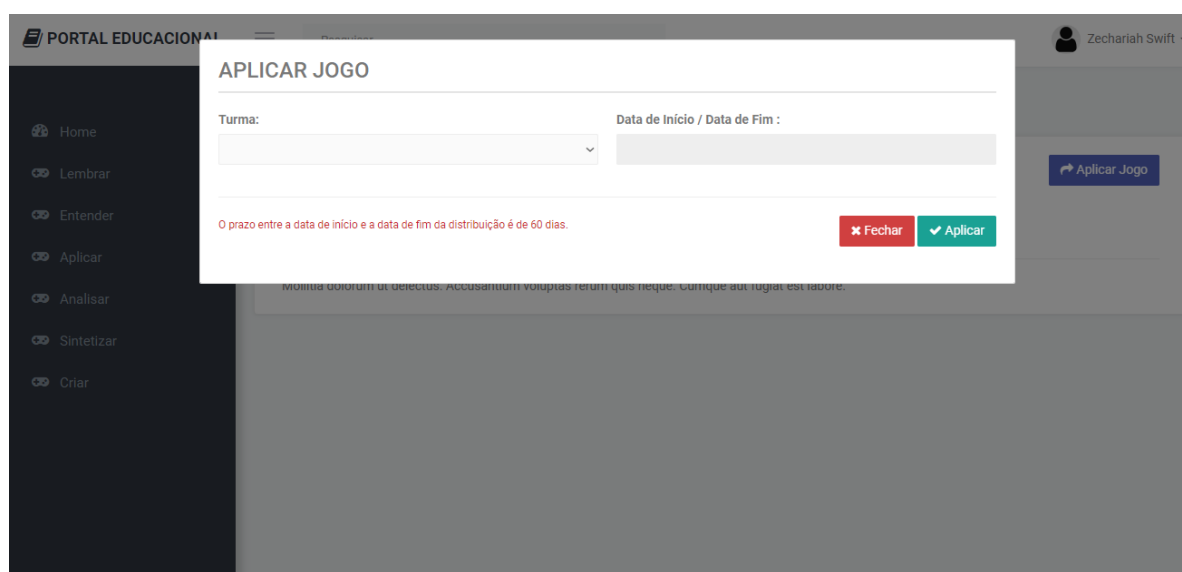


Figura 16 - Tela de distribuição do jogo por turma.

Fonte: Gabriel Neres, Tiago Neves

Os alunos por sua vez recebem essa distribuição em suas contas e caso as mesmas estejam dentro do prazo estabelecido por seus professores podem ter acesso ao jogo em questão em sua página principal (Figura 17), ao finalizarem, as informações referentes às pontuações do aluno são registradas e se tornam dados estatísticos para os docentes conforme a Figura 18 a partir desse momento o jogo se torna inacessível até que haja novas distribuições.

PORTAL EDUCACIONAL

Lafayette Nader

Home

JOGOS DISPONÍVEIS

Sala/Turma	Jogo	Data de Início	Data de Fim	Ações
Dr. Kristy Skiles DVM	Lucinda Balistreri	21/10/2017	29/11/2017	🔗

JOGOS ENCERRADOS

Sala/Turma	Jogo	Data de Início	Data de Fim
Dr. Kristy Skiles DVM	Thad Heller	06/10/1978	06/02/1997
Dr. Kristy Skiles DVM	Lucinda Balistreri	20/05/1979	01/11/1980

Figura 17 - Jogo exemplo distribuído para os alunos.
Fonte: Gabriel Neres, Tiago Neves

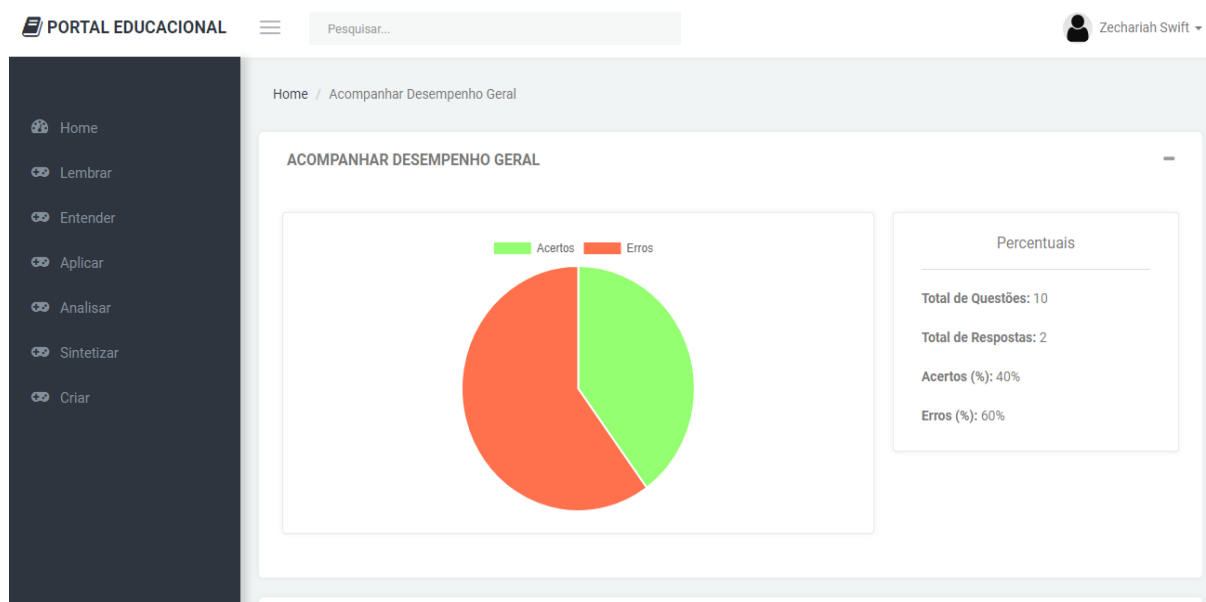


Figura 18 - Tela de estatísticas referentes ao jogo distribuído.
Fonte: Gabriel Neres, Tiago Neves

Quando o tempo limite da distribuição é encerrado ou quando o docente exclui a distribuição, nenhum dos alunos vinculados à aquela turma ao qual o jogo havia sido distribuído, podem ter acesso ao jogo em questão e todos os dados já obtidos são removidos.

Foi desenvolvido um jogo no formato de quis (Figura 19) para exemplificar o funcionamento da ferramenta ele conta com uma série de perguntas definidas pelo desenvolvedor em um arquivo XML, quando o jogo se inicia essas perguntas são carregadas de forma aleatória e exibidas para os usuários conforme mostrado na Figura 20, cada uma das perguntas contam com 4 alternativas dispostas na tela do jogo, onde somente uma das alternativas é a correta, caso o usuário informe uma resposta incorreta, o mesmo deixa de acumular pontos, já quando as perguntas são respondidas de forma correta o usuário recebe uma pontuação atribuída automaticamente pelo sistema e então uma nova pergunta é sorteada.

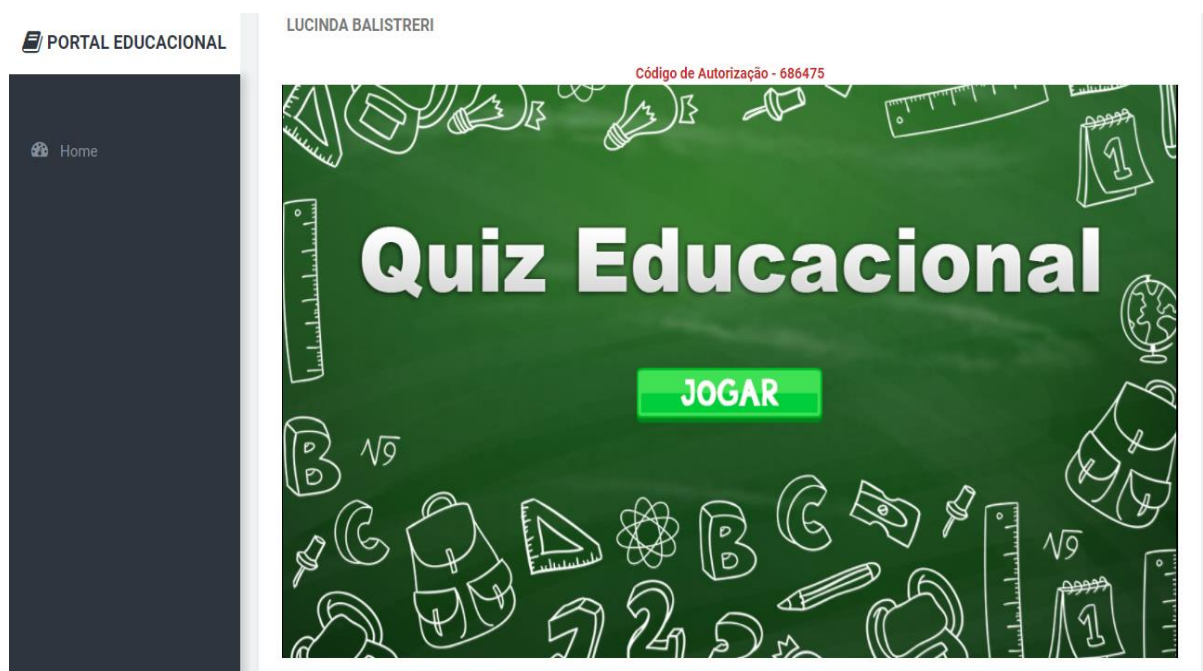


Figura 19 - Jogo de quis desenvolvido para exemplo.
Fonte: Gabriel Neres, Tiago Neves

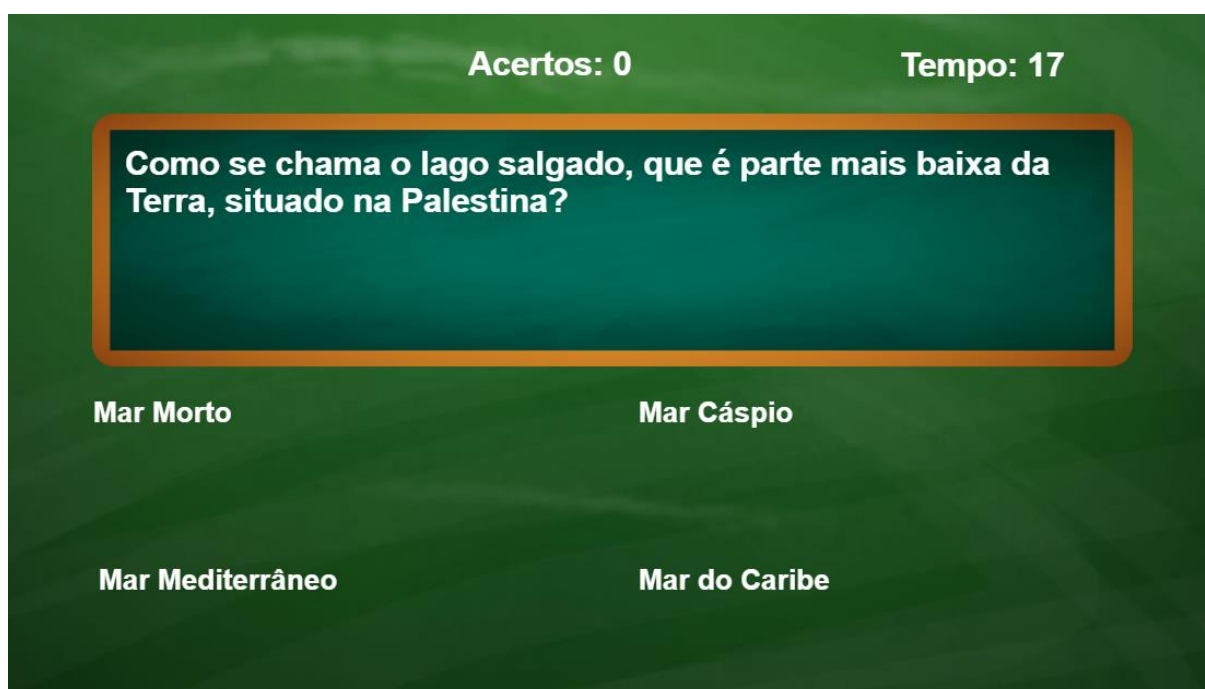


Figura 20 - Apresentação de uma pergunta dentro do quiz.
Fonte: Gabriel Neres, Tiago Neves

Ao fim de todas as perguntas, o jogo é encerrado e então é exibida a tela de finalização, onde são apresentados a pontuação conquistada pelo jogador e uma medalha simbólica (Figura 21) calculada a partir da porcentagem de acertos que ele obteve dentro do jogo após a confirmação de fim de jogo informada pelo usuário, esses dados são enviados à aplicação que será responsável por tratar e armazenar esses dados obtidos para que posteriormente eles venham a ser utilizados pelos docentes em forma de gráficos e dados estatísticos.



Figura 21 - Fim de jogo e pontuações finais no jogador.
Fonte: Gabriel Neres, Tiago Neves

Cada um dos jogos desenvolvidos para a ferramenta deve atender a um dos seis níveis da Taxonomia de Bloom, buscando atender os objetivos e características de cada um deles, um mesmo jogo pode ser distribuído várias vezes para a mesma turma, desde que, o período de distribuição esteja encerrado ou quando o docente solicita o cancelamento de uma determinada distribuição (Figura 22).



Figura 22 - Modal de exclusão de um jogo distribuído.

Fonte: Gabriel Neres, Tiago Neves

O processo de inserção dos jogos na plataforma se dará em duas etapas, na primeira delas o Desenvolvedor devidamente autenticado no sistema terá acesso ao formulário com os dados necessários para a avaliação do jogo que será submetido, nesse formulário o Desenvolvedor pode selecionar uma ou mais categorias nas quais ele acha que seu jogo pode se enquadrar e em seguida deve informar uma descrição a respeito do porque esse jogo deve ser enquadrado dentro das categorias informadas e também deverá informar um e-mail para contato. Após submetido o formulário com os dados necessários uma equipe será responsável por validar e avaliar o jogo em questão, através dos dados informados anteriormente pelo Desenvolvedor. A equipe pode aprovar, reprovar ou solicitar uma adequação ou melhoria para o jogo, caso a equipe verifique que os dados descritos não estão de acordo com o jogo e a categoria mencionada os mesmos podem reprovar o jogo ou solicitar alguma melhoria para que o mesmo entre em acordo com a categoria e suas regras, nesses casos o Desenvolvedor receberá um e-mail com as devidas considerações realizadas pela equipe e pode estar readequando o jogo e reenviando o mesmo para uma nova avaliação. Caso a equipe aprove o jogo, o Desenvolvedor será informado via e-mail de que o jogo será publicado nas categorias

informadas e estará disponível em até 24 horas. Os cadastros dos jogos podem ser cancelados a qualquer momento pelo Desenvolvedor necessitando apenas informar os motivos e o mesmo será removido da plataforma em até 24 horas. Nesse momento todas as distribuições realizadas desde jogo serão canceladas e os professores serão avisados via e-mail.

A ferramenta foi desenvolvida de forma a dar suporte a todos os níveis da Taxonomia de Bloom, para exemplificar o funcionamento da mesma foi criado um jogo exemplo utilizando o Construct 2 o que propiciou um desenvolvimento rápido, através do seu conceito de programação utilizando a orientação a eventos. A utilização do Lumen nos proporcionou a construção da ferramenta de forma que a mesma possa ser utilizada como uma API, baseada em serviços, o que nos permite que futuramente a aplicação possa ser utilizada ou expandida como forma de aplicativo, sem que haja uma grande mudança na programação já existente. Já na construção do *front-end* com a utilização do Angular JS e do Bootstrap, possibilitou a criação de interfaces mais limpas e intuitivas, dispondo inclusive de suporte para dispositivos *mobile* nas mais diversas resoluções disponíveis.

CONSIDERAÇÕES FINAIS

O presente estudo possibilitou o desenvolvimento de uma ferramenta case, que permite a classificação de soluções tecnológicas segundo os níveis de conhecimento da Taxonomia de Bloom, além disso, proporcionou o estudo de novas tecnologias para o desenvolvimento da aplicação, processo esse que nos permitiu agilizar a criação da mesma.

De um modo geral os professores e alunos, possuem acesso as tecnologias, e vários desses professores já fazem o uso das mesmas dentro da sala de aula, no entanto, não utilizam uma estratégia ou metodologia para analisar, se o uso desses equipamentos está realmente colaborado com o processo de aprendizagem por parte de seus alunos.

Ao desenvolver a ferramenta proposta, verificou-se que existem diversas tecnologias para o desenvolvimento de sistemas e aplicativos, algumas delas citadas neste trabalho. A solução proposta possui duas partes, uma é a plataforma web que foi desenvolvida com Angular JS, Bootstrap, HTML 5 e Lumen já para a segunda parte foi utilizado o Construct 2 na criação do game de perguntas e respostas, o quis.

Outra parte fundamental para o desenvolvimento do software foi o estudo da Taxonomia de Bloom, este estudo possibilitou que a ferramenta tivesse um apoio pedagógico, já que a maioria das soluções encontradas para educação, não seguem nenhum processo ou mesmo uma metodologia, permitindo assim que os objetivos propostos sejam alcançados. O uso da Taxonomia de Bloom possibilitou uma melhor categorização dos jogos conforme o objetivo de cada uma. Dessa forma, o docente pode aplicar o jogo que melhor se adeque a prática que ele busque.

Dada a importância do desenvolvimento do software, torna-se necessário a classificação de novos games na ferramenta para que contribua de forma direta com os professores.

Nessa perspectiva a utilização dessa ferramenta tecnológica em sala de aula, pode possibilitar ao professor tornar o conteúdo mais desafiador e instigante para seus alunos, além disso, o professor terá uma ferramenta como suporte nesse processo, onde o mesmo poderá verificar, acompanhar e aprender ainda mais a respeito dessa taxonomia, seus níveis de conhecimentos e seus objetivos propiciando a interação entre alunos e professor no processo de aquisição de conhecimento.

REFERÊNCIAS BIBLIOGRÁFICAS

ABREU, Cristiano de, EISENSTEIN, Evelyn, ESTEFENON, Susana Bruno. **Vivendo esse Mundo Digital: Impactos na Saúde, na Educação e nos Comportamentos Sociais**. ArtMed, 08/2013. VitalSource Bookshelf Online.

ADRIEL, Wendell, **Introdução ao Laravel Framework PHP**. Disponível em: <<https://www.devmedia.com.br/introducao-ao-laravel-framework-php/33173/>> Acesso em 13 de novembro de 2017.

ARAÚJO, Fabrício, **Vantagens e Desvantagens do PHP**. Disponível em: <<https://www.inforlogia.com/vantagens-e-desvantagens-do-php/>> Acesso em 11 de novembro de 2017.

BARBIERE, Lu, **O Que é Bootstrap e Para Que Serve?** Disponível em: <<https://www.ciawebsites.com.br/dicas-e-tutoriais/o-que-e-bootstrap/>> Acesso em 07 de novembro de 2017.

BIANCA, Sonnewend. **Infográfico, Uso de tecnologias na sala de aula tende a aumentar**. Disponível em: <<http://info.geekie.com.br/uso-de-tecnologias-na-sala-de-aula/>>. Acessado em 10 outubro de 2017.

BLACKBOARD. **Uso da tecnologia em sala de aula desperta interesse nos alunos**. Disponível em: <<http://blackboard.grupoa.com.br/blog/uso-da-tecnologia-em-sala-de-aula-desperta-interesse-nos-alunos/>>. Acessado em 15 dezembro de 2017.

BRASIL, Ministério da educação. **Guia de tecnologias educacionais**. Brasília, 2008. 98 p. Disponível em: <http://portal.mec.gov.br/seb/arquivos/pdf/Avalmat/guia_de_tecnologias_educacionais.pdf> Acesso em 24 de outubro de 2016.

DIAS, Raphael, **Game Engine: o que é, para que serve e como escolher a sua**. Disponível em: <<http://producaodejogos.com/game-engine/>>. Acesso em: 19 de maio de 2017.

EIS, Diego, **Uma breve história do CSS**. Disponível em: <<http://tableless.com.br/uma-breve-historia-do-css/>>. Acesso em 25 de outubro de 2016.

ESPÍNDOLA, Rafaela, **O que é a gamificação e como ela funciona?** Disponível em: <<http://www.edools.com/o-que-e-gamificacao/>>. Acesso em: 20 de agosto de 2017.

FEITOSA, Eduardo , **O que é o HTML5**. Disponível em: <<http://www.devmedia.com.br/o-que-e-o-html5/25820>>. Acesso em 24 de outubro de 2016.

FERREIRA, Andréia de Assis; VENTURA Paulo Cezar Santos. **Artigo, Concepções de professores de História da rede municipal de ensino de Belo Horizonte acerca da informática educacional**. Disponível em: <<http://coralx.ufsm.br/revce/revce/2007/02/a11.htm/>>. Acessado em 10 outubro de 2017.

FERREIRA, Fabio, **Lumen – Um simples tutorial do micro framework PHP**. Disponível em: <<http://cafenaveia.blog.br/lumen-um-simples-tutorial-sobre-o-microframework-php-em-laravel/>>. Acesso em 13 de novembro de 2017.

LEONE, Leonello, **6 frameworks PHP que você precisa conhecer!** Disponível em: <<https://becode.com.br/frameworks-php-que-voce-precisa-conhecer/>>. Acesso em: 11 de novembro de 2017.

LOPES, José Junio, **A Introdução da Informática no Ambiente Escolar**. Disponível em: <<http://smec.salvador.ba.gov.br/site/documentos/espaco-virtual/espaco-edu-com-tec/artigos/a%20introducao%20da%20informatica%20no%20ambiente%20escolar.pdf>> Acesso em 20 de maio de 2017.

LORENZONI, Marcela, **Gamificação: O que é e como pode transformar a aprendizagem**. Disponível em: <<http://info.geekie.com.br/gamificacao/>>. Acesso em 17 de maio de 2017.

MACHADO, Henrique, **Unity 3D: Introdução ao desenvolvimento de games**. Disponível em: <<http://www.devmedia.com.br/unity-3d-introducao-ao-desenvolvimento-de-games/30653>>. Acesso em 21 de maio de 2017.

MARTINI, Júlio César, **Zend Framework – Uma visão geral**. Disponível em: <<https://imasters.com.br/artigo/4609/zend/zend-framework-uma-visao-geral?trace=1519021197&source=single>>. Acesso em 10 de novembro de 2017.

MOREIRA, Marco Antonio. **O Que é Afinal Aprendizagem Significativa**. Disponível em: <<http://moreira.if.ufrgs.br/oqueefinal.pdf>> Acesso em 17 de maio de 2017.

MUSTAFA, Eduardo, **JavaScript – 20 anos de história e construção da web**. Disponível em: <<http://imasters.com.br/front-end/javascript/javascript-20-anos-de-historia-e-construcao-da-web/?trace=1519021197>>. Acesso em 24 de outubro de 2016.

PACIEVITCH, Yuri, **C#**. Disponível em: <<http://www.infoescola.com/informatica/c-sharp/>>. Acesso em 17 de maio de 2017.

PILGRIM, Mark. **HTML5: Up and Running**: Dive into the Future of Web Development. ArtMed, 08/2010. O'Reilly media.

RAMOS, Mario Roberto Vieira, **O uso de tecnologias em sala de aula**. Disponível em: <<http://www.uel.br/revistas/lenpes-pibid/pages/arquivos/2%20Edicao/MARCIO%20RAMOS%20-%20ORIENT%20PROF%20ANGELA.pdf>> Acesso em 17 de setembro de 2017.

REIS, Paulo, **Simplificando a sua vida com o Eloquent ORM**. Disponível em: <<https://www.bugginhoacademy.com.br/blog/simplificando-a-sua-vida-com-o-eloquent-orm/>>. Acesso em 11 de novembro de 2017.

REIS, Vinicius, **Por que Vue JS é uma boa opção?** Disponível em: <<http://www.devmedia.com.br/a-evolucao-da-linguagem-de-programacao-c/28639>>. Acesso em 12 de novembro de 2017.

REIS, Vinicius, **Aplicações com Vue JS: Pensando em componentes**. Disponível em: <<http://www.vuejs-brasil.com.br/aplicacoes-com-vuejs-pensando-em-componentes/>>. Acesso em 12 de novembro de 2017.

SOUZA, Jefferson, **Introdução ao padrão MVC**. Disponível em: <<https://www.redemasterinfo.com.br/blog/post/introducao-ao-padrao-mvc/?titulo=introducao-ao-padrao-mvc/>>. Acesso em 13 de novembro de 2017.

THE PHP GROUP, **História do PHP**. Disponível em: <https://secure.php.net/manual/pt_BR/history.php.php>. Acesso em 10 de novembro de 2017.

TOLEDO, Thiago Ferreira, **A evolução da linguagem de programação C#**. Disponível em: <<http://www.devmedia.com.br/a-evolucao-da-linguagem-de-programacao-c/28639>>. Acesso em 17 de maio de 2017.

TORI, Romero, **Tendências tecnológicas no ensino superior**. Disponível em: <<http://www.aredde.inf.br/tendencias-tecnologicas-no-ensino-superior>>. Acesso em 05 de março de 2017.

VALENTE, Kaio. **Porque utilizar AngularJS no seu próximo projeto**. Disponível em: <<https://tasaf0.org/2014/11/26/porque-utilizar-angularjs-no-seu-proximo-projeto/>>. Acesso em 18 de maio de 2017.

VALLE, DO, Luiza Ribeiro, MATTOS, Maria J.V. de, COSTA, José Wilson. **Educação Digital: A Tecnologia a favor da Inclusão**. Penso, 08/2013. VitalSource Bookshelf Online.

VASCONCELLOS, Paulo. **O que é Gamificação? Conheça a ciência que traz os jogos para o cotidiano**. Disponível em: <<http://www.techtudo.com.br/noticias/noticia/2016/07/o-que-e-gamificacao-conheca-ciencia-que-traz-os-jogos-para-o-cotidiano.html>>. Acesso em 12 de dezembro de 2017.

WALTENBERG, Rodrigo. **Angular JS: O que é e porquê utilizar**. Disponível em: <<http://blog.algaworks.com/o-que-e-angularjs/>>. Acesso em 15 de maio de 2017.

WILLIAMS, Wesley. **Lumen, o Micro Framework do Laravel**. Disponível em: <<https://blog.schoolofnet.com/2015/04/lumen-o-microframework-do-laravel/>>. Acesso em 09 de novembro de 2017.