



MINISTÉRIO DA EDUCAÇÃO
SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO TOCANTINS
CAMPUS PORTO NACIONAL
CURSO SUPERIOR DE LICENCIATURA EM COMPUTAÇÃO

ELISAMA MARTINS GONÇALVES

**GODOT GAME ENGINE - UM MOTOR DE JOGO LIVRE COMO FERRAMENTA
PARA CRIAÇÃO DE APPS EDUCACIONAIS**

**Porto Nacional – TO
2018**



MINISTÉRIO DA EDUCAÇÃO
SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO TOCANTINS
CAMPUS PORTO NACIONAL
CURSO SUPERIOR DE LICENCIATURA EM COMPUTAÇÃO

ELISAMA MARTINS GONÇALVES

**GODOT GAME ENGINE - UM MOTOR DE JOGO LIVRE COMO FERRAMENTA
PARA CRIAÇÃO DE APPS EDUCACIONAIS**

Trabalho de Conclusão de Curso apresentado à Coordenação do Curso de Licenciatura em Computação do Instituto Federal do Tocantins – Campus Porto Nacional, como exigência à obtenção do grau em Licenciado em Computação.

Orientador: Prof. M. Sc. Janio Carlos Nascimento Silva.

**Porto Nacional - TO
2018**

G586g Gonçalves, Elisama Martins
Godot game engine: um motor de jogo livre como ferramenta para criação de APPS educacionais/ Elisama Martins Gonçalves. – Porto Nacional: Instituto Federal de Educação, Ciência e Tecnologia, 2018.
70f.

Trabalho de Conclusão de Curso (Graduação) – Instituto Federal de Educação, Ciência e Tecnologia, Curso de Licenciatura em Computação, Porto Nacional, TO, 2018.

Orientador: Prof. Me. Jânio Carlos Nascimento Silva

1. Desenvolvimento ágil de aplicações. 2. Jogos educativos. 3. Apps educativos. 4. Plataformas livres de desenvolvimento. 5. Game engine I. Gonçalves, Elisama Martins. II. Título.

CDD: 005.1

ELISAMA MARTINS GONÇALVES

**GODOT GAME ENGINE - UM MOTOR DE JOGO LIVRE COMO FERRAMENTA
PARA CRIAÇÃO DE APPS EDUCACIONAIS**

Trabalho de Conclusão de Curso apresentado à Coordenação do Curso de Licenciatura em Computação do Instituto Federal do Tocantins – *Campus Porto Nacional*, como exigência à obtenção do grau em Licenciado em Computação.

Aprovado em: ____ / ____ / ____

BANCA AVALIADORA

Prof. M. Sc. Jânio Carlos Nascimento Silva
IFTO - *Campus Porto Nacional*

Prof. Esp. Paulo César de Sousa Patrício
IFTO - *Campus Porto Nacional*

Prof. Me. Tharles Lopes de Oliveira Guedes
IFTO - *Campus Porto Nacional*

Dedico a meus pais Agnaldo e Ester, irmã Laís e minha sobrinha Marcela (Meu “pacotinho” de amor).

Agradecimentos

O ato de agradecer é sempre uma experiência de reflexão sobre como os resultados alcançados são frutos da colaboração de muitas pessoas que nos orientam, corrigem o rumo que tomamos ou contribuem de inúmeras maneiras. (PETRILLO, 2008)

Agradeço primeiramente a Deus, por ter me sustentado e me guiado durante minha jornada acadêmica.

A minha família:

Mãe, obrigada! Por cada noite mal dormida me esperando nos primeiros semestre do curso, pelas orações a mim dedicadas, pois, sei que seus joelhos no chão é o que me mantém em pé. Pai agradeço ao senhor pela dedicação, empenho, ajuda e ao esforço em me manter. Irmã (“maninha”) agradeço por ser meu exemplo de dedicação e por sempre me incentivar e acreditar em mim.

Minhas sobrinhas, minhas “Elas”, Daniela, Isabella e Manoela, que sempre me recebe com alegria e com aquele abraço amoroso, quando eu chego em casa, onde destes eu tiro muitas forças para continuar.

A minha tia Dimailda, minha segunda mãe. Tia obrigada pelo afeto, apoio e orações.

Meu orientador, Mestre Janio Carlos, por ter aceitado ingressar nessa longa jornada comigo, por ser peça fundamental da realização do meu sonho.

Meus amigos:

Acredito que amigos sejam anjos colocados por Deus ao nosso lado para nos apoiar, ajudar, nos fazer rir e às vezes até mesmo nos dar broncas, então quero ser grata a eles que foram de suma importância para a realização desse sonho, em especial quero aqui citar aqueles que fizeram e fazem a diferença na minha vida todos os dias, Hellen, Denys, Karen, Luana, Pedro, Iago, Luciana, Adriana, Lucas.

Meus professores:

Ao logo desses anos meus professores se tornaram grandes exemplos, exemplo esses que seguirei em minha carreira docente, então professores quero agradecer, pelo esforço de vocês e pela dedicação e dizer que esse TCC, essa vitória não é somente minha, mas também a de vocês.

Agradeço também a todos que direta ou indiretamente contribuição para a construção desse trabalho.

A educação escolar e o professor que a ministrar não tem, no geral um referencial de mundo que se compatibiliza com a realidade circundante e com seus possíveis avanços. O espaço educacional parece imune, preservado desses avanços, mantendo o velho, pela indiferença às mudanças do meio. (Valente, 1996, p.129)

Resumo

O desenvolvimento de *apps* educativos com a utilização de *games engines* ou simplesmente motores de jogos, permite facilitar o processo criativo destes softwares aumentando assim a produtividade, pois, sua interface dinâmica e seus elementos pré-programados, dão agilidade ao procedimento. E o uso de plataformas livres para o desenvolvimento dos jogos ajuda no fator custo-benefício durante sua produção até o produto final. Este trabalho teve como objetivo realizar um breve estudo da Godot Game Engine, aplicado ao desenvolvimento de jogos educativos. Nessa perspectiva, levanta-se o seguinte problema: “É possível criar *apps* educacionais de maneira ágil usando a *game engine* selecionada?” O método de pesquisa utilizado para o desenvolvimento desse trabalho foi o qualitativo, e o estudo foi desenvolvido a partir de pesquisa bibliográfica e de campo. Conclui-se que a utilização da ferramenta tornou o desenvolvimento de jogos mais ágeis, pois possui uma quantidade notável de ferramentas pré-programas.

PALAVRAS-CHAVE: Desenvolvimento Ágil de Aplicações; Jogos Educativos; *Apps* Educativos; Plataformas Livres de Desenvolvimento; *Game Engine*.

Abstract

The development of educational apps with the use of games engines or simply game engines allows to facilitate the creative process of these softwares increasing productivity, because its dynamic interface and its preprogrammed elements give the procedure agility. And the use of free platforms for the development of games helps in the cost-benefit factor during its production until the final product. This work aimed to carry out a brief study of Godot Game Engine, applied to the development of educational games. In this perspective, the following problem arises: "Is it possible to create educational apps in an agile way using the selected game engine?" The research method used to develop this work was the qualitative one, and the study was developed from bibliographic research and field. It is concluded that the use of the tool made the development of games more agile, since it has a remarkable amount of pre-program tools

KEYWORDS: Agile Application Development; Educational Games; Educational Apps; Free Development Platforms; Game Engine.

Sumário

1.	INTRODUÇÃO.....	11
1.1.	Justificativa.....	13
1.2.	Problema.....	13
1.2.	Objetivo Geral.....	13
1.4.	Objetivos Específicos.....	14
1.5.	Hipóteses.....	14
1.6.	Estrutura do trabalho.....	14
2.	REVISÃO DA LITERATURA.....	16
2.1.	Desenvolvimento ágil de aplicações: Engenharia de Software.....	16
2.2.	Plataformas Livres de desenvolvimento.....	18
2.2.1.	Software Livre na Educação.....	19
2.2.2.	Conceitos de linguagem de programação.....	20
2.2.3.	Biblioteca.....	22
2.2.4.	API (Interface de Programação de Aplicativos).....	23
2.2.5.	<i>Game Engine</i> (Motor de Jogo).....	22
2.2.6.	Godot Game Engine.....	24
2.3.	Objetos de Aprendizagens: Conceitos Básicos.....	25
2.3.1.	Jogos Educativos.....	27
2.3.2.	Características de um Jogo.....	29
2.4.3.	Jogos Digitais na Educação.....	30
2.3.4.	<i>Apps</i> Educativos.....	33
3.	METODOLOGIA.....	35
3.1.	A Escolha do Software.....	35
3.2.1.	Ensaio de prototipação.....	43
3.2.2.	Oficina experimental.....	44
3.2.	Coleta de dados.....	45
3.3.	Análise.....	46
4.	RESULTADOS.....	47
4.1.	Evidências baseadas no <i>survey</i> comparativo das <i>engines</i>	47
4.2.	Evidências baseadas no ensaio de prototipação e oficina I.....	47
4.3.	Evidências baseadas no instrumento de coleta de dados aplicado na oficina II.....	48
5.	CONSIDERAÇÕES FINAIS.....	55

5.1. Limitações da Pesquisa.....	56
5.2. Trabalhos Futuros.....	56
REFERÊNCIAS.....	58
APENDÊNCE A – PLANO DE ENSINO OFICINA DA I.....	63
APENDÊNCE B – PLANO DE ENSINO OFICINA DA II.....	65
APENDÊNCE C – INSTRUMENTO DE COLETA DE DADOS.....	67

1. INTRODUÇÃO

Os jogos educativos vêm sendo utilizados como ferramentas em sala de aula há muito tempo, sendo esses um instrumento didático valioso aonde venha facilitar o aprendizado, já que, os jogos influenciam no desenvolvimento do educando, como por exemplo, a motricidade, a integração social, afetividade e a criatividade (MALUF, 2004). Falando especificamente sobre a educação infantil, temos um campo onde os jogos têm uma grande relevância nas fases iniciais, pois os jogos estimulam o processo cognitivo, ou seja, auxiliam no desenvolvimento da percepção, atenção e raciocínio da criança (MOYLES, 2002). Nesse contexto, os jogos não podem ser vistos somente com um entretenimento, mas de modo que venha possibilitar uma aprendizagem de várias habilidades e consequentemente um desenvolvimento prazeroso do aluno.

Por outro lado, com os avanços tecnológicos, os jogos eletrônicos começam a fazer parte do meio educacional, à vista disso, muitos professores foram conquistados pelos jogos digitais educacionais com a finalidade de criar uma dinamicidade para suas aulas e com um grande acervo já disponível de apps educacionais, encontrados facilmente e de forma gratuita contribuiu para que esses professores se interessassem por tecnologia e começassem a utilizá-la em sala de aula. A fim de reproduzir situações reais ou irreais, aproximando assim os alunos ao conteúdo trabalhado despertando um maior interesse pelas aulas e facilitando o processo de ensino/aprendizagem e permitindo assim que o aluno possa fazer descobertas e desenvolver seu raciocínio lógico, crítico e aumentando sua autoconfiança (FORTUNA, 2000).

Nota-se que o uso de jogos eletrônicos no meio educacional, pode ser uma forma de unir a educação e a tecnologia, da mesma forma que pode auxiliar no processo de investigação de problemas, tais como déficit de atenção, aprendizado, entre outros, que permite assim uma melhor administração das ações a serem tomadas a respeito do mesmo. E com isso deve-se refletir sobre as práticas educacionais, em contradição seria um meio para fechar algumas brechas deixadas pelo então conhecido como ensino tradicional, onde a tecnologia não venha a substituir o livro didático, mas que venha somar ao que ele oferece e ao mesmo tempo deve ser pensando “a necessidade dos alunos em participarem ativamente da aula e a possibilidade de materialização do objeto de estudo, visando o aumento de

sua interação com o sujeito, facilitando dessa forma a construção do conhecimento” (RAVELLI, 2010).

Devido a isso é proposta desse trabalho a utilização de mecanismos que venha oferecer facilidade a esse processo de desenvolvimento de jogos eletrônicos, propiciando ao professor que se interesse pela criação de *apps* um meio ágil pelo qual o mesmo possa criar uma ferramenta que irá intervir diretamente no problema ou apenas corroborar ao processo de ensino/aprendizagem. As game engines ou simplesmente motores de jogos são bibliotecas responsável por gerenciar o processo de entrada de dados dentre outras funcionalidades. Uma engine é uma coleção integrada de vários algoritmos, como vários módulos como, por exemplo: gráficos, 2D e/ou 3D, físico, entrada e saída, áudio, animação entre outras (STANG, 2003).

Para este trabalho então o motor de jogo escolhido foi a Godot Game Engine, uma ferramenta 2D e 3D, livre e *open source* com interface dinâmica e de fácil entendimento, oferecendo praticidade no desenvolvimento de *apps*. Possui uma linguagem de programação própria, e é baseado em uma linguagem já existente, o Python. O GDScript a linguagem de programação da Godot é simples e sucinta possibilitando fácil aprendizagem. Enfatiza-se que o desenvolvimento de tais jogos em plataformas livres e o uso de softwares livres na educação, visa o custo-benefício tanto ao desenvolvedor quanto ao usuário final. Almeida (2000) salienta que o uso do software dessa natureza colabora para a dissipação do uso de tecnologia na educação, decorrente das vantagens que os mesmos apresentam principalmente no que diz respeito à flexibilidade e custo para a implantação e uso.

Salienta-se também a conceituação e a definição de Engenharia de Software, abordando padrões de projeto e desenvolvimento ágil, a fim de produzir softwares com padrões e recomendações da Engenharia de Software e forma ágil de produção. Temos ainda, conceituação de linguagem de programação que se trata da linguagem de comunicação e interação entre usuário e máquina (SEBESTA, 2011), de API (Interface de Programação de Aplicativos) que tem como definição, um conjunto de padrões de programação que permite a criação de aplicativos (SCHMITT, 2013) e de Bibliotecas, que se define como um conjunto de funções, compartilhando assim soluções por meio de métodos (FERNANDES, 2009), objetivando o desenvolvimento de *apps* educacionais, através da ferramenta de

desenvolvimento Godot Game Engine, objeto este estudado para construção de conhecimento acerca de uma maneira de trabalhar com objetos de aprendizagem.

1.1. Justificativa

Baseando-se na alta constante da informatização, podemos observar o uso intenso dos jogos eletrônicos de computadores, este trabalho justifica-se devido ao imenso crescimento na busca da utilização dos mesmos na educação, nos incentivando então a estudar o desenvolvimento de jogos eletrônicos como uma ferramenta de auxílio ao processo de ensino/aprendizagem, cujo desenvolvimento auxilia na obtenção do conhecimento e também na obtenção de uma ferramenta única para ser trabalhada em sala de aula, suprindo assim a necessidade específica na qual aquele profissional docente necessita.

Deve-se destacar também que há escassez de pesquisas na área de desenvolvimento de *apps* educacionais foi um ponto crucial para determinarmos o nosso objeto a ser estudado, contanto com a necessidade de encontrar uma ligação uma junção entre a área pedagógica e a computação.

Tal como o trabalho buscou fazer um breve estudo da Godot Game Engine, a princípio foram pesquisadas alguns dos principais motores de jogos que podem ser equiparados a ela, em seguida observamos sua utilização, para a criação de *apps* educacionais de forma ágil, visando à simplificação do processo criativo desde, que, pode ser visto como ferramenta valiosa, como meio de interação social, aumento do imaginativo e intelectual.

1.2. Problema

Diante das possibilidades do motor de jogo, surgiu a necessidade de responder a seguinte pergunta:

É possível criar *apps* educacionais de maneira ágil usando a *game engine* selecionada?

1.2. Objetivo Geral

Tendo como objetivo:

Construir conhecimento acerca da ferramenta Godot Game Engine que possibilite uma metodologia ágil de criação de *apps* educacionais.

1.4. Objetivos Específicos

Os objetivos são:

- Estudar e compreender o ambiente e as características do motor de jogo para utilização como ferramenta base para criação de *apps* educacionais.
- Avaliar a abordagem proposta (Processo e Ambiente) em um contexto de desenvolvimento de *apps* educacionais.
- Analisar a usabilidade da ferramenta selecionada para criação de *apps* educacionais e interação dos mesmos em sala de aula.

1.5. Hipóteses

Como hipóteses para o problema apresentado, temos:

- **Hipótese I:** A Godot Game Engine apresenta-se como uma notável e ágil ferramenta para o desenvolvimento de *apps* educacionais.
- **Hipótese II:** A engine selecionada apresenta diversas vantagens na produção de *apps* educacionais, para qualquer profissional da área docente.
- **Hipótese III:** A interface intuitiva e de fácil aprendizagem da Godot Game Engine contribui para o desenvolvimento de *apps* educacionais em pouco tempo, mesmo com a necessidade de programação.

1.6. Estrutura do trabalho

Este trabalho se encontra organizado em etapas estruturada como citado abaixo.

No capítulo primeiro, desenvolve-se a introdução, contendo também justificativa, problema, objetivos e hipóteses do trabalho.

No segundo capítulo, apresenta-se a revisão da literatura, que versará sobre:

Desenvolvimento Ágil de Aplicativos: Engenharia de Software, Plataformas Livres de Desenvolvimento, Softwares Livres, Conceito de Linguagem de Programação, Biblioteca, API, *Game Engine*, Godot Game Engine, Objetos de Aprendizagem: Conceitos Básicos, Jogos Educativos, Características de um Jogo, Jogos Digitais na Educação, Apps Educativos.

No capítulo terceiro, a metodologia, contendo as fases de desenvolvimento do trabalho e como foi realizada a escolha do software a ser estudado e realização das oficinas.

No quarto capítulo, apresentam-se os resultados da pesquisa. E no quinto capítulo considerações finais e trabalhos futuros.

2. REVISÃO DA LITERATURA

A literatura relacionada às metodologias de desenvolvimento de aplicativos educacionais está em crescimento, dado aos benefícios do uso das ferramentas tecnológicas na construção do conhecimento. A dinamicidade das novas tecnologias torna-se um incentivo a entender a educação de forma diferente, nos levando a reflexão das práticas e que impulsiona a novos parâmetros que “refliram essa necessidade humana de se completar, de desvendar, descobrir e se refazer” (SILVA, 2014).

O Desenvolvimento de *apps* depende de conceitos predefinidos pela Engenharia de *Software* bem como suas diretrizes nas quais serão apresentadas neste tópico. E no contexto da nossa proposta veremos também os jogos como ferramentas catalisadoras do aprendizado e o desenvolvimento do mesmo de uma forma ágil visando uma melhoria do processo de ensino/aprendizagem (ALVES, 2013).

2.1. Desenvolvimento ágil de aplicações: Engenharia de Software

Para Pressman (2001), o processo de *software* é um conjunto de tarefas necessárias para a produção de um *software* de qualidade. O resultado do processo é um produto, ou seja, um *software*, que reflete a forma como o processo foi realizado. Embora existam vários métodos de desenvolvimento de *software* existem atividades genéricas comuns a todos eles (SOARES, 2004):

Especificação do software: definição do que o sistema deve fazer, definição dos requisitos e das restrições do *software*.

Desenvolvimento do software: projeto e implementação do *software*, o sistema é desenvolvido conforme sua especificação.

Validação do software: o *software* é validado para garantir que as funcionalidades implementadas estejam de acordo com o especificado.

Evolução do software: a evolução do *software* é devido à necessidade dos clientes.

Objetivando a qualidade final do produto gerado são propostos métodos que buscam padronizar as atividades e o ciclo de vida de um processo de *software*. E

esta padronização provoca a criação de documentos, artefatos e marcos capazes de caracterizar o âmbito do *software* (NETO, 2004).

À vista disso, uma metodologia de desenvolvimento de *software* é um conjunto de representação simplificado de um processo, que combine com a descrição do trabalho e/ou procedimentos e convenções utilizadas por seus membros. Sendo ainda, uma metodologia de desenvolvimento de *software*, que seja concomitante de práticas recomendadas para o desenvolvimento do *software*, sendo que essas práticas geralmente passam por fases ou passos, que possa ser subdividido em processo, para ordená-la e melhor gerenciá-la (COSTA, 2006.)

De acordo com Fagundes (2005), o maior desafio e o mais comum dentro do desenvolvimento de *software* é entregar um produto que atenda as reais necessidades do cliente, e isso ainda deve ser dentro do prazo e do orçamento prévio. Para dar suporte a essa ação a Engenharia de *Software* oferta metodologias que contribuem para o processo de desenvolvimento, dentre elas as metodologias tradicionais e as metodologias ágeis.

Dentre esses processos, existem as metodologias tradicionais, que têm como uma de suas particularidades a grande quantidade de documentos gerados, essas metodologias surgiram em um contexto de desenvolvimento de *software* muito diferente do atual, baseado apenas em um mainframe e terminais burros (PRESSMAN, 2001). E ainda as metodologias tradicionais de desenvolvimento de *software* necessitam de requerimentos completos e fixos, tornando-os por vezes “pesados” e não flexíveis (SEGUNDO; MOURA, 2004), e as metodologias ágeis, focada mais nas pessoas do que na documentação.

As metodologias ágeis têm sido apresentadas como alternativa às metodologias tradicionais de desenvolvimento de *software*. Estas metodologias interpelam o desenvolvimento de *software* delineando de diferentes modos, preconizados anteriormente pela Engenharia de *Software*, que tinham forte ênfase na documentação e no processo (COSTA et alii, 2005).

Esse método surgiu como alternativa ao processo de desenvolvimento de *software* voltado para pequenas organizações, deixando muitas por vezes a padronização dos processos, documentação excessiva e gerenciamento dos requisitos, como destaca Zanatta (2004). A proposta das metodologias ágeis é o desenvolvimento de sistemas de uma maneira mais rápida, eficiente e que atenda as reais necessidades dos clientes.

Nas metodologias ágeis os clientes fazem parte ativamente do desenvolvimento do processo, a colaboração dos mesmos faz com que certos documentos se tornam desnecessários. A documentação prevê limites e condições, acordadas previamente pelas partes que pode até funcionar, mas somente com o trabalho corporativo desenvolvedores/clientes, poderá entender e chegar ao que o cliente realmente almeja (HILMAN, 2004).

Os métodos ágeis podem ser considerados como um meio termo entre a ausência de processo e o processo exagerado (ZANATTA, 2004). Pressman (2011) ressalta que as metodologias ágeis se desenvolvimento é um esforço para evitar fraquezas reais e nítidas na Engenharia de Software convencional, o desenvolvimento ágil oferece prerrogativas muito importantes, porém, não são adequados para todos os projetos, produtos, pessoas e situação.

As metodologias ágeis possuem a característica de serem aplicado em projetos não muito complexos, utilizando ciclos interativos curtos, planejamento guiado por funcionalidades, retroalimentação constante, tolerância a mudanças, proximidade da equipe, intimidade com o cliente e foco no ambiente geral de trabalho da equipe (HILMAN, 2004).

2.2. Plataformas Livres de desenvolvimento

De diversas plataformas de desenvolvimento de jogos, há necessidade de escolher o mais adequado para o projeto de acordo com o jogo que vai ser desenvolvido, é ideal que tenham planejamentos quanto ao custo, tempo de desenvolvimento, ferramentas disponíveis, linguagem a ser utilizada, entre outros (BITTENCOURT, 2005). Das plataformas existentes, se faz necessário o uso de ferramentas livres e acessíveis para desenvolvimento dos jogos digitais educacionais, *software* livre pode ter código aberto independente de ser multiplataforma.

De acordo Paticci (2006), a metodologia de cooperação defendida pelos adeptos do *Software* livre, teve início no fim dos anos 50, na forma de um modelo de *software* colaborativo ou cooperativo, da qual grandes beneficiados eram as fabricantes de *hardwares*, que dos poucos usuários de uma época em que economicamente se dava mais valor ao *hardware* do que ao *software* em si. Mas foi só a partir dos anos 70 com o surgimento dos “*Softwares Houses Independentes*”,

que se iniciou então a preocupação e os direitos de propriedade intelectual e/ou empresarial do *software*.

Para a Fundação de Softwares Livre (2017), o conceito de *software* livre, é o usuário ter a liberdade de “executá-lo” para qualquer propósito, “copiá-lo” e distribuir os mesmos de modo que você possa ajudar o próximo, “estudá-lo” como o programa funciona e adaptá-lo para suas necessidades e acesso ao código-fonte é pré-requisito para esta liberdade, “aperfeiçoá-lo” liberar aperfeiçoamento de modo que toda a comunidade se beneficie.

Segundo Alves (2011), o conceito de código aberto teve surgimento para tentar minimizar a resistência dos usuários ao *software* livre, então em 1988 um grupo ligado ao movimento do *software* livre decidiu criar a “*Open Source Initiative*” (OSI), que amoedou com o termo “código aberto” (*open source*). O *software open source* é conhecido também como “*free software*”, neste caso em questão a palavra *free* não indica gratuito, mas sim liberdade. A liberdade é um dos princípios apoiado pelo movimento *open source*, além do ideal de colaboração e de compartilhamento de conhecimento (IWASAKI, 2008).

2.2.1. Software Livre na Educação

Para Xavier (2001), debater *software* livre na educação não se trata de apenas discussões e perspectivas que deverão ser pensadas para o futuro. Hoje já existem resultados de projetos bem-sucedidos de implementação de sistemas livres na educação, sejam em escolas, universidades, privadas ou públicas. Os especialistas da área dizem que o uso do computador e do celular como fonte de material didático é indispensável, sejam em curso a distância ou em sala aula como fonte de acesso para os professores e alunos. A necessidade estimula cada vez mais o uso de softwares livres e o torna uma importante ferramenta para o avanço tecnológico na educação, do processo de aprendizagem e do maior acesso à informação.

De acordo com Hexsel (2002), o uso do *software* livre não é a solução de todos os problemas da educação, pode contribuir muito para a melhoria do setor, com custos mais baixos. O autor também ressalta que o fato do *software* ser livre, não resulta em baixa qualidade, mas pelo contrário pode-se conseguir o melhor *software*, sem custos e ainda com algumas vantagens, como ter a liberdade para

distribuir cópias para os alunos, para que eles se dediquem mais ao conteúdo a ser estudado e/ou até mesmo ser modificado e adequado, fazendo com que o processo de ensino/aprendizagem seja cada vez mais atrativo.

Pinheiro (2003), indaga que educação é uma “ponte” do aluno com a cidadania e autorrealização, e o *software* livre no contexto educacional é um importante atenuante para o processo de ensino, proporcionando a inclusão digital.

Segundo Michelazzo (2003, p. 270).

(...) O grande desafio é saber usá-lo. Iniciativas devem partir de todos, com a finalidade principal de modificar as atuais grades curriculares. Estas devem ser dotadas de mecanismos flexíveis para a adoção de novas tecnologias, como o software livre e suas disciplinas correlatas. Sem essa premissa, pouco se pode fazer para aproveitar todo o potencial do software livre e calemo-nos 18 status quo da educação de uma geração de “apertadores de teclas” (...)

O emprego de *software* dessa natureza colabora para a dissipação do uso de tecnologia na educação, decorrente das vantagens que os mesmos apresentam principalmente no que diz respeito à flexibilidade e custo para a implantação e uso. No contexto educacional o uso tecnológico do software livre aborda elementos importantes que devem ser considerados, são eles: *Hardware*, *Software*, Infraestrutura de redes e Recursos Humanos (Almeida, 2000).

Portanto, o sistema livre tem mostrado bons resultados em suas aplicações na área educacional.

2.2.2. Conceitos de linguagem de programação

Quando se trata de programação de computadores, a linguagem de programação serve como meio de comunicação entre o usuário e a máquina. Esta linguagem de programação é o meio de comunicação entre o pensamento humano e a precisão requerida para o processamento pelo computador (SEBESTA, 2011).

Além de ser uma comunicação, a linguagem de programação também é um conjunto de instruções, como afirma Willrich (2017, p. 83).

Informalmente, uma linguagem de programação pode ser definida como sendo um conjunto limitado de instruções (vocabulário), associado a um conjunto de regras (sintaxe) que define como as instruções podem ser associadas, ou seja, como se podem compor os programas para a resolução de um determinado problema.

Os programas de computadores podem ser escritos em várias linguagens, como por exemplo: C, Python e Java, dentre muitas outras. E essas linguagens são divididas em níveis, linguagem de baixo nível, também conhecido como linguagem de máquina (*machine language*) e linguagem de alto nível (SEBESTA, 2011).

A linguagem de máquina é a conhecida como a linguagem de mais baixo nível de entendimento pelo ser humano, ou seja, apenas uma Unidade Central de processamento (UCP), que fará a leitura dessa linguagem. A linguagem de máquina é composta completamente por números, conhecidos como números binários, formado apenas por zero e um, que a torna praticamente impossível de entender diretamente (SEBESTA, 2011).

Como mostra a **Figura 1**:

1	0	0	1	1	0	1	0	1	1	0	0	1	0	1	0
0	0	1	0	1	0	1	1	0	0	1	0	1	1	0	0
1	0	0	0	0	1	0	1	1	0	1	0	0	1	1	1
0	0	0	1	0	1	1	0	0	1	1	1	0	0	1	0

Figura 1: Linguagem de máquina.

A linguagem de alto nível é conhecida por se aproximar mais da linguagem natural, ela se torna mais fácil por possuir uma estrutura e palavras-chave. **Ver Figura 2.** Tornando os programas mais fáceis de serem lidos e escritos apresentando assim uma vantagem sobre a linguagem de baixo nível. Os programas escritos nessas linguagens são conversos para baixo nível através de um compilador (WILLRICH, 2017).

```

1 #include <stdio.h>
2
3 int main()
4 {
5     printf("Hello, World! \n");
6     return 0;
7 }
```

Figura 2: Linguagem de Alto Nível.

O objetivo do desenvolvimento da linguagem de alto nível seria a independência da linguagem de máquina e ela foi amplamente alcançada. Isso graças a estrutura e aos comandos das linguagens de alto nível que são atributos

que não se referenciam na linguagem de baixo nível. Os programas podem ser facilmente escritos e compilados em uma máquina ou outra (WILLRICH, 2017).

2.2.3. Biblioteca

De acordo com Fernandes (2009,) biblioteca é um conjunto de funções e é muito utilizada no mundo da programação, a ideia da biblioteca é compartilhar soluções por meio de funções ou métodos. Então desenvolvedores disponibiliza biblioteca com funções prontas, assim os programadores podem utilizá-las, permitindo assim que os programadores desenvolvam mais rapidamente.

A biblioteca abarca códigos e dados auxiliares, que deriva serviços a programas independentes o que permite o compartilhamento de dados e a alteração de códigos e dados de forma modular (FERNANDES, 2009).

Segundo o autor anteriormente citado, existem basicamente dois tipos de bibliotecas: as estáticas e as compartilhadas. As bibliotecas estáticas são identificadas pela extensão *.a*. O termo “estática” deve-se ao fato da linguagem se tornar uma parte fixa do binário final. Já as bibliotecas compartilhadas são identificadas pela extensão *.so*. O termo “compartilhado” “expõe precisamente a característica de que um ou mais programas poderá utilizar a mesma biblioteca, ocupando assim menos espaço na memória” (FERNANDES, 2009).

2.2.4. API (Interface de Programação de Aplicativos)

API é uma interface com conjuntos de padrões de programação que permite a construção de aplicativos e a sua utilização de maneira não tão evidenciada pelo usuário. A API é a “matrix”, ou seja, uma interface que roda por trás de tudo, enquanto o usuário usufrui de um aplicativo ou site a API estará conecta a diversos outros aplicativos. Uma API funciona, através da comunicação de vários códigos, ou seja, a API integra diversas funções em um *site*, por exemplo, como, busca de imagens, notícias e outros (SCHMITT, 2013).

2.2.5. Game Engine (Motor de Jogo)

Os motores de jogos são bibliotecas de desenvolvimento responsável pelo gerenciamento do jogo, das imagens, do processamento de entrada de dados e outras funções. Uma engine é uma coleção integrada de vários algoritmos, englobado em vários módulos como: gráficos, criação de jogos em 2D e/ou 3D, físico/detecção de colisão, entrada e saída, áudio, inteligência artificial, animação, banco de dados, e interface gráfica de usuário (STANG, 2003).

Estes sistemas possuem inúmeras vantagens, razão pela qual são tão populares nos dias de hoje, exemplo disso é que um motor de jogo pode simplificar o desenvolvimento dos jogos. A ferramenta chama às bibliotecas que executam funções que facilitam ao desenvolvedor como, por exemplo, uma função de desenhar imagens, ou de deixar seu jogo mais portável. Dessa forma, você pode conduzir seu jogo mais facilmente para outra biblioteca gráfica ou até outra plataforma. Para isso, você apenas modifica as chamadas bibliotecas gráficas que estão sendo utilizadas na própria *engine*. Disponibilizam todas as funcionalidades para desenvolver um jogo, reduzindo os custos que às vezes podem chegar a zero, quando se trata em adquirir um *software* e reduzindo o tempo de produção por ter várias etapas pré-prontas ou programadas. Outra grande vantagem é a concepção de plataformas, permitindo que o mesmo jogo possa ser implantado em várias plataformas com pouca ou nenhuma mudança no código base (TORI, *et al*, 2004).

Durante muito tempo, as empresas de jogos fizeram os seus próprios motores de jogo, sem os tornar públicos, hoje isso está mudando, no passar dos últimos anos, o custo de fazer estes motores cresceu e cada vez mais empresas têm-se especializado em fazer ou motores de jogo ou componentes para motores de jogo para vender a outras empresas, em vez de desenvolver jogos. Estas empresas desenvolvem, por exemplo, mediadores, ou seja, softwares que ligam duas aplicações separadas, e conseguem disponibilizá-los a preços mais baixos o que faz com que a maioria dos estúdios de desenvolvimento de jogos escolha por utilizá-los (ALVES *et al*, 2013, p.13).

2.2.6. Godot Game Engine

Segundo o site da própria *engine* a Godot teve o seu início em 2001, como uma ferramenta fechada escrita em C++, que era somente usada pelos seus criadores, Juan Linietzky e Ariel Manzur do estúdio OKAM na Argentina. A ferramenta continuou sendo melhorada para uso interno durante alguns anos, então decidiram lançá-la como um *software* livre e de *open source* em 2014, sendo que a *engine* era uma das poucas que é livre, *open source* e multiplataforma na mesma composição. A *engine* trabalha com filosofia de simplicidade, aplicada em praticamente toda a ferramenta desde algumas coisas relacionadas ao editor e ao *script* até as outras relacionadas ao núcleo da *engine*.

Ainda em seus documentos disponível em seu *site*, quanto ao sistema de *script*, ou seja, a linguagem de programação, os desenvolvedores decidiram implementar uma linguagem que fosse muito mais simples que a linguagem de programação C++. Eles chegaram a cogitar usar linguagens como Python ou Lua, mas tiveram problemas *garbage-collector* (coletor de lixo), ou seja, o gerenciamento de memória, entre outros. Então, acabaram criando uma linguagem de *script* própria que suportasse a arquitetura da *engine*, chamada de GDScript (*Godot Script*), o GDScript é uma linguagem simples que faz referência ao Python como algumas instruções específicas.

A Godot preserva os paradigmas de POO (Programação Orientada a Objeto), onde existe uma hierarquia de árvores de *nodes* (nós), e cada node possui um tipo de comportamento diferente, sendo assim, alguns *nodes* compartilham dos mesmos comportamentos e dessa forma pode haver um node “pai” que vai servir como base, mantendo assim uma relação de herança com os *nodes* “filhos” comuns na orientação a objetos (TUMEO, 2017, p 02).

A Godot permite a criação de jogos de qualquer tamanho e complexidade usando os formatos 2D, 3D ou ambos ao mesmo tempo. As criações feitas na *engine* ainda podem ser exportadas para plataformas desktop, móveis e web. O desenvolvimento do motor é completamente aberto, feito em conjunto com a comunidade (Godot Game Engine, 2017).

2.3. Objetos de Aprendizagens: Conceitos Básicos

De acordo com o comitê de padronização das Tecnologias de Aprendizagem (LTSC) um Objeto de Aprendizagem é definido como “qualquer entidade digital ou não digital que pode ser utilizada, reutilizada ou referenciada durante o aprendizado apoiado pela tecnologia.” (IEEE, 2002).

A flexibilidade e a possibilidade de reutilização como característica de um objeto de aprendizagem, que facilitam a expansão do conhecimento assim como a sua atualização (IEEE, 2002).

Os OA's podem ser criados em qualquer mídia ou formato podendo ser simples como uma animação ou uma apresentação de *slides* ou mais complexos como uma simulação, um jogo, entre outros.

Assim como salienta Gallotta (2004, p 101),

“Os objetos de aprendizagem permitem a construção de contextos digitais para os conteúdos que serão explorados. Esses contextos fazem uso de uma série de ferramentas midiáticas, tais como música, desenhos, gráficos, simulações, jogos etc. A contextualização permite aos alunos traçar mais facilmente uma relação entre determinado conteúdo e suas aplicações práticas e enxergar a interdependência das várias disciplinas”.

A metodologia na qual o OA é utilizado será um dos fatores-chaves a determinar se a sua escolha foi acertada ou não e se ela pode levar o aluno ao desenvolvimento do pensamento crítico. Enfatiza-se que, como em qualquer planejamento de aula, deve-se escolher o objeto de aprendizagem adequado para uso em atividade didática que fica definido a partir do objetivo que se pretende alcançar na aprendizagem de um determinado conteúdo, visando os diferentes tipos de OA's e suas características.

Para Singh (2001), os objetos de aprendizagem devem ser divididos em três importantes definidas partes:

- a) “Objetivos” devem-se esclarecer aos aprendizes quais objetivos pedagógicos norteiam o uso do recurso; além disso, apresentá-los os pré-requisitos, ou uma lista dos conhecimentos prévios necessários para um bom aproveitamento do conteúdo.
- b) “Conteúdo instrucional” é a apresentação do material didático necessário para que o aluno possa atingir os objetivos propostos.

- c) “Prática e *feedback*”, permite ao aluno utilizar o material e receber retorno sobre o atendimento dos objetivos propostos no objeto de aprendizagem.

Mendes (2004) complementa que, os objetos de aprendizagem são caracterizados em sua estrutura de operacionalidade. “Reusabilidade”, ou seja, o objeto poderá/deverá ser usado e reutilizado por diversas vezes em cenários diferentes de aprendizagem. “Adaptabilidade”, o mesmo deve-se adaptar a qualquer ambiente de ensino. “Granularidade” trata-se do “tamanho” do objeto de aprendizagem. Um OA de maior granularidade é considerado pequeno e/ou em estado “bruto”, exemplo de disso: “uma imagem, um texto, ou um fragmento de um áudio”. É considerado um OA de menor granularidade, “uma página web indireta que combinam textos, imagens e vídeos”. “Acessibilidade” faz acesso via Internet para ser usado em ambientes diversos. “Durabilidade”, viabilidade de permanecer sendo usada, independente se houver ou não mudanças tecnológicas. “Interoperabilidade”, o OA deve ter a habilidade de atuar em uma “variedade de *hardwares* sistemas operacionais”. “Metadados” (“dados sobre dados) descrevem a propriedade de um objeto, como título, autor, data, assunto, entre outros”.

Essa distinção e caracterização dos objetos de aprendizagem se fazem necessária uma vez que, aqui já citado, um OA possui qualquer formato desde uma simples imagem, chegando até os objetos mais complexos como uma simulação de realidade virtual.

Por ser um campo de pesquisa em alto crescimento Koper (2003), traz que a ideia fundamental implícito ao conceito de objeto de aprendizagem como sendo a construção de componentes de instrução relativamente pequenos, que possam ser reutilizados em variados contextos de aprendizagem. O autor propõe, então, que um objeto de aprendizagem seja “qualquer recurso digital que possa ser reutilizado para o suporte ao ensino”.

Koper (2003) define um objeto de aprendizagem como “qualquer recurso digital, reproduzível concretizado”. Existem diferentes definições para objetos de aprendizagem e muitos outros termos são utilizados para definir o mesmo, causando assim um conflito conceitual. E isso ocorre caso o autor queira definir tal objeto com características pedagógicas ou técnicas. E segundo Gutierrez (2004, p. 06).

Um objeto de aprendizagem pode ser conceituado como sendo todo o objeto que é utilizado como meio de ensino/aprendizagem. Um

cartaz, uma maquete, uma canção, um ato teatral, uma apostila, um filme, um livro um jornal, uma página web, podem ser um objetos de aprendizagem. A maioria destes de aprendizagem pode ser utilizada, modificada ou não e servir para outros objetivos que não os originais. Em muitas escolas existem aquele famoso depósito, nem sempre muito organizado onde guardam (ás vezes, sepultam) objetos que fizeram parte de aulas de projetos. Um depósito de onde se recuperam estes objetos para modificação, até que o desgaste inviabilize novas transformações e utilizações.

Com tudo, podemos entender que os objetos de aprendizagem são recursos capazes de proporcionar, mediante a combinação de diferentes mídias digitais ou não, situações de aprendizagem em que o professor assume o papel de mediador e o estudante o caráter de sujeito ativo dentro do processo de ensino/aprendizagem.

2.3.1. Jogos Educativos

A busca por inovações é incessante e a cada dia que passa, vemos o quanto isso é importante para o crescimento da humanidade. No universo de nossas salas de aulas, nos deparamos com diferenças relacionadas a níveis sociais, culturais, raciais, religiosos, entre outras. E diante de tanta tecnologia, acessível à maioria da população, por muitas vezes, não conseguem atrair a atenção dos alunos. Então fica claro a necessidade de diversificarmos nossas metodologias de ensino, na busca de resgatarmos o interesse e incentivar o gosto do estudante pelo aprender.

Os jogos usados no contexto educacional revelam-se de suma importância, pois promovem cenários de ensino/aprendizagem e amplia a construção do conhecimento. A introdução de jogos, como atividade lúdica no processo de ensino deve ser prazerosa, desenvolvendo de ação ativa, de concentração e motora. “A estimulação, a variedade, o interesse, a concentração e a motivação são igualmente proporcionados pela situação lúdica (...)” (MOYLES, 2002, p.21).

E ainda como indaga Maluf (2004), o ato de brincar/jogar é, e sempre foi, uma atividade desenvolvida espontaneamente que proporciona prazer, e está presente e disponível para o ser humano em qualquer fase de sua vida. E como ainda ressalta o autor “Brincar é: comunicação e expressão, associados a pensamento e ação; um ato instintivo voluntário; uma atividade exploratória; ajuda às crianças no seu desenvolvimento físico, mental, emocional e social; um meio de aprender a viver e não um mero passatempo” (MALUF, 2004, p. 17).

É interessante que os jogos educativos sejam utilizados como ferramenta de apoio, construindo assim elementos úteis no reforço de conteúdos já apreendidos anteriormente. Em contrapartida, essa ferramenta de ensino deve ser instrutiva, transformada numa disputa divertida, e, que consiga, de forma sutil, desenvolver um caminho correto ao aluno, ou seja, o brincar deve saudável, contribuindo assim com o crescimento do aluno, senso crítico, autoconfiança trabalhada em grupo, respeito de limites e ao próximo. O fator competição, durante os jogos sempre irá existir, mas não há motivos para preocupação, pois o professor precisa estar preparado para esclarecer que esse tipo de competição ocorre apenas no jogo e não, na vida. (MALUF, 2004).

Marinho (2007) ressalta que, as atividades lúdicas realizadas com os alunos possibilitam o desenvolvimento da sua sociabilidade, do seu reconhecimento corporal, espacial e até mesmo da linguagem. É através dessas competências que os mesmos aprendem a criar e vivenciar situações fora do seu cotidiano, a interagir umas com as outras e a trocar conhecimento. E para Marinho (2007, p.85).

Podemos dizer que a criança, quando brinca e joga, também treina para um melhor convívio social, pois aprende a cumprir regras, trabalhar em grupo, conhecer e desafiar limites, ao mesmo tempo em que melhora sua agilidade e perspicácia diante das situações que aparecem durante as brincadeiras e jogos.

É importante elencar também que deve existir uma relação com a aprendizagem, mesmo que seja de forma empírica e que seja marcado pelo desenvolvimento, tanto do estudante, quanto do professor. E nessa relação aluno/professor e vice-versa ambos estão sendo, a sua maneira enriquecida e inserida no processo de ensino/aprendizagem e instigando assim o prazer das apropriações e da construção do conhecimento (MARINHO 2007).

Desta forma estaremos colaborando na construção da sua individualidade, da sua marca pessoal, do desenvolvimento cognitivo e altruísta. Não devemos nos esquecer, também, que ao proporcionar estes momentos de animação e diferentes do rotineiro, não estamos deixando de lado o compromisso de repassar os conteúdos previstos para a série (MARINHO 2007).

2.3.2. Características de um Jogo

Para Huizinga (2003), o jogo pode ser definido como uma atividade lúdica muito mais ampla que um fenômeno físico ou reflexo psicológico, sendo ainda, um ato voluntário concretizado como evasão da vida real, limitado pelo tempo e espaço, criando a ordem através de uma perfeição temporária. E Huizinga (1971), caracteriza ainda os jogos da seguinte forma, que ele ser uma atividade livre, permitindo ao jogador não ser vida “corrente”, nem vida “real”, mas antes possibilitar uma evasão para uma esfera temporária de atividade com orientação própria, ou seja, mesmo que a atividade seja dirigida deve-se haver espaço para o jogador criar seu mundo particular e desenvolver-se. O jogador deve se manter jogando até o fim, dentro de certos limites de tempo e espaço, possuindo um caminho e um sentido próprios. “Criar ordem e ser ordem”, uma vez que quando há a menor desobediência, o jogo acaba e todo jogador deve respeitar regras, caso contrário ele é excluído do jogo, para que o jogador tenha apreensão das noções de limites. Deve-se permitir repetir o jogo tantas vezes quantas forem necessárias, dando assim oportunidade a qualquer instante de analisar os resultados obtidos com a atividade e por último o jogo deve ser permanentemente dinâmico.

Lucchese; Ribeiro (2009) caracteriza o jogo em quatro elementos, sendo eles: “representação, interação, conflito e segurança”.

A “representação”, o jogo deve oferecer uma representação simplificada e subjetiva da realidade, tendo um conjunto de regras.

A “interação”, o ponto crucial na representação da realidade situa-se na forma como ela se altera e a representação interativa.

O “conflito”, o conflito surge naturalmente a partir da interação do jogador e esse elemento está presente em todos os jogos.

A “segurança” uma vez que o conflito tende a criar um cenário de perigo, dele surge uma situação de risco físico.

Em uma interpelação mais cognitiva, Passerino (1998), demonstra que deve haver a capacidade de absorver o participante de maneira intensa, ou seja, em sua totalidade, com o clima de entusiasmo, sentimento de exaltação e tensão seguidas por um estado de alegria e distensão e também envolvimento emocional, criando assim uma atmosfera de espontaneidade e criatividade. Tendo que ser pensado também em limites, pois, o jogo deve ter um estado inicial, meio e fim,

caracterizando assim um caráter de dinamismo. O jogo também deve ter possibilidades de repetição, dando assim mais chances de dinamismo aos jogadores.

No jogo deve haver limitações de espaço, o espaço reservado qual for assumirá forma de um mundo temporário e fantástico, durante o tempo no qual os participantes estiverem jogando. Dentro do jogo deve haver a existência de regra, então o jogo acontecerá de acordo com certas regras que determinam o que “vale” ou não dentro do mundo imaginário do jogo, que auxiliará no processo de integração social das crianças, estimulando assim, a imaginação e auto afirmação e autonomia do participante do jogo (PASSERINO, 1998).

A participação em jogos contribui para a formação de atitudes sociais como respeito mútuo, cooperação, obediência às regras, senso de responsabilidade, senso de justiça, iniciativa pessoal e em grupo. O jogo é o vínculo que une a vontade e o prazer durante a realização de uma atividade. O ensino utilizando meios lúdicos permite criar ambientes gratificantes e atraentes servindo como estímulo para o desenvolvimento integral dos usuários. Desta forma as abordagens citadas acima se complementam, identificando os principais fatores pertinentes ao estudo de jogos em um contexto geral (KISHIMOTO, 1996).

Para Farias (1995), os jogos proporcionam o acesso da criança a vários tipos de conhecimentos e habilidades ao mesmo tempo em que propicia uma melhor interação entre os colegas em sala de aula, e, com o próprio professor. No entanto, para cumprir o papel educativo, o jogo deve favorecer o desenvolvimento da linguagem, criatividade e o raciocínio dedutivo.

2.4.3. Jogos Digitais na Educação

Há muito tempo, estão sendo pesquisadas madeiras diversas para tornar o processo de ensino/aprendizagem, satisfatório tanto para os alunos quanto para os professores e na busca por esses avanços está à inserção dos computadores na qualidade da educação. O computador foi inserido nas escolas com a intenção de melhorar a qualidade da educação (SILVA; MORAIS, 2011).

A era digital tem modificado a natureza da relação professor/aluno, entre os imigrantes e os nativos digitais, da qual surgem alguns dos principais desafios em educação nos tempos atuais. Os métodos tradicionais de ensino confrontam-se com

alunos digitalmente alfabetizados, os quais cresceram usando múltiplos recursos tecnológicos e que, ao invés de simplesmente receber e memorizar os conteúdos, exigem uma formação diferenciada. Esta geração tem a tecnologia como uma aliada ao processo de ensino/aprendizagem, usando-a com naturalidade (SILVA; MORAIS, 2011).

Baseado nisso vem crescendo o número de pesquisas para encontrarem formas de unir ensino e diversão com a criação de jogos digitais educativos. Por proporcionarem métodos educativos atraentes e inovadores, o educando tem possibilidade de aprender de maneira mais ativa, dinâmica e motivadora, os jogos se tornam auxiliares fundamentais no processo de ensino/aprendizagem (SAVI; ULBRICH, 2008). Jogo e educação com o objetivo de potencializar o aprendizado também se torna auxiliador do aluno que tem dificuldades em aprender com ensino tradicional e no mundo de livros, lousas e pincéis.

Jogos digitais muitas vezes são utilizados nas escolas para intensificar o processo de ensino/aprendizagem e estimula o aprendizado tornando-o mais significativo na construção do conhecimento. Fortuna (2000) aborda que os professores não são difíceis de compreender a influência que o jogo tem no desenvolvimento do indivíduo. A maioria entende que brincar e estudar são coisas distintas que professores de crianças pequenas oferecem o brincar; e demais séries os professores oferecem o estudar. Ainda de acordo com Fortuna (2000) os professores ao tentar quebrar essa oposição o reforçam, porém constantemente o vínculo jogo-aprendizagem favorece o ensino dirigido sobre o jogo, o modificando ao ocultá-lo. Jogos educativos são utilizados em todos os níveis de ensino desde a pré-escola a cursos de graduação, especializações entre outros. Os jogos não devem ser vistos apenas como meio de diversão, mas como forma de aproximar o aluno da tecnologia.

Bittencourt (2005) ressalta que diversas escolas se preocupam demais com métodos de ensino que não se dão oportunidade de fazer uso de meios tecnológicos como jogos, assim limitando o aluno de um aprender mais atraente e inovador diferente de seu costume. Escolas limitam os jogos para momentos de intervalo ou se der tempo para diversão após a aula. Inserir o jogo como exercício independente e agradável é aceitar que o indivíduo não é integrado somente de razão e que os alunos do século XXI também necessitam se aperfeiçoar em habilidades

humanísticas para estarem qualificados a “viverem em uma sociedade de conhecimento, da informação e da mobilidade” (BITTENCOURT, 2005).

Para serem utilizados como eficientes ferramentas didáticas pelas escolas são importantes que tenham uma atenção às dificuldades técnicas e pedagógicas apresentadas pelos professores que interferem na inserção dos jogos no processo de ensino/aprendizagem (SAVI; ULRICHT, 2008). A intenção de levar para sala de aulas cada vez mais as TIC's (Tecnologias da Informação e Comunicação) é o que vem sendo inserido nas propostas de ensino inovador, mas cabe ressaltar o importante caminho da especialização e capacitação tecnológica do professor em que o deixe habilitado para conciliar o jogo e o conteúdo a ser trabalhado em foco no que ele deseja que o aluno aprenda.

Para Fortuna (2000), os jogos desenvolvem imaginação e criação, atingindo as funções simbólicas e construção do sistema de representação que por sua vez manipula a aquisição da leitura e escrita. À medida que o jogo trabalha a ação mental há desenvolvimento do raciocínio, quanto à atividade lúdica desenvolvendo os aspectos operativos e figurativos do pensamento. O desafio não é informatizar a escola, mas oferecer maneiras que facilitem o aprendizado e desperte o interesse em aprender com as possibilidades que o jogo oferece.

Segundo Falkembach, (2006, p. 111):

Os jogos educacionais oferecem um mecanismo alternativo de aprendizagem e ganham popularidade nas escolas. Eles devem ser usados adequadamente pelos professores como um poderoso motivador para o início do processo de aprendizagem, estimulando as relações cognitivas como o desenvolvimento da inteligência, as relações afetivas, verbais, psicomotoras e sociais. Os jogos provocam uma reação ativa, crítica e criativa dos educandos, socializando o conhecimento. O aluno é diferenciado e valorizado como pessoa. Os aspectos afetivos são determinantes na construção da personalidade e eles se revelam de forma explícita no jogo.

O jogo precisa ser visto como ferramenta que irá desenvolver o aprendizado do aluno, o deixará mais próximo da informação que deseja passar com auxílio de um jogo digital. Conforme o autor o educando se torna mais ativo, crítico e criativo e isso possibilita construção seu próprio conhecimento com a troca de informações entre aluno-aluno e aluno-professor. O jogo nem sempre deve levar o aluno para um mundo digital cheio de fantasias fora do mundo real, ele também necessita ser preciso e com informações que contribuam no processo ensino/aprendizagem.

O objetivo não é transformar as escolas em *lan houses*, nem que cada escola possua ambientes diferentes para aprendizagens diversas, mas formular locais nos quais o professores encontre apoio para trabalhar jogos que remetem as questões éticas, políticas, ideológicas, culturais, etc. Podendo ser investigadas e comentadas com os alunos, escutando e percebendo a intimidade que os jogadores, os alunos, formam com as mídias, indagando, intervindo, mediando ao desenvolvimento de novos aspectos para os fatos. Também descobrir com os mesmos novos meios de ver e entender tais instrumentos educacionais (ALVES, 2008). Segundo a visão do autor as escolas devem entender que utilizar jogos em sala de aula não estará fugindo do objetivo da aula, mas sim ser associado com o conteúdo para facilitar o aprendizado.

Entretanto não é apenas inserir jogos no processo ensino/aprendizagem e sim possuir professores preparados para desenvolver novas formas de ensino, inovando suas práticas por meio da aproximação que cada aluno tem com as tecnologias tornado aliado na construção do conhecimento.

2.3.4. Apps Educativos

O computador se tornou uma ferramenta pela qual todos, sem distinção, podem atingir um determinado grau de conhecimento. Mesmo através de uma dominação mínima sobre essa máquina podemos fazer uso dela e assim obtermos dela variadas informações e conhecimento. Surge então um novo olhar para o computador como uma ferramenta de auxílio no processo de aprendizagem.

E para que haja um detalhamento em relação aos aspectos referentes a avaliação de cada tipo de software educacional é preciso ressaltar a existência de uma questão geral, que é fundamental neste processo. Como salienta Galvis (1988, p. 117):

(...) o computador deve ser usado no processo ensino-aprendizagem, antes de qualquer outra coisa, como um meio para implementar o que com outros meios não seria possível ou seria difícil obter. Diferentemente do que alguns educadores temem, não se trata de implementar com o computador a ação de outros meios educativos cuja qualidade está bem demonstrada. Este raciocínio não é estranho, se considera que o computador é um bem escasso e também custoso, cujo uso deve oferecer o máximo de benefícios, neste caso educativo (...).

O uso de *apps* na educação possibilita motivar o aluno, o que por sua vez permite o mesmo ampliar sua criatividade e a posteriori a reflexão do que foi aprendido, gerando assim um aprendizado de suas próprias experiências vividas, conhecimento esse empírico através de suas próprias percepções sensoriais (KANT, 2010).

Dentre várias maneiras de desenvolver ambientes computacionais que podem favorecer o processo de ensino/aprendizagem, destacam-se os *apps* educacionais. Um *app* educacional é caracterizado por Lucena (1992), como sendo “todo aquele programa que possa ser usado para algum objetivo educacional, pedagogicamente defensável, por professores e alunos, qualquer que seja a natureza ou finalidade para o qual tenha sido criado”.

3. METODOLOGIA

O método de pesquisa utilizado para o desenvolvimento desse trabalho foi o qualitativo, apoiando-se em técnicas de coleta de dados, pois, a pesquisa procura interpretar e compreender as opiniões e expectativas de entes relacionados à educação, especialmente neste trabalho, acadêmicos de Licenciatura em Computação, Física, Matemática. Lüdke e André (1986) afirmam que a pesquisa qualitativa tem o ambiente natural como sua fonte direta de dados e o pesquisador como seu principal instrumento.

O estudo foi desenvolvido a partir de pesquisa bibliográfica, uma vez que o pesquisador procura interpretar a realidade, sem nela interferir, por meio de pesquisa em material já existente, como afirma Gil (2008) esse tipo de pesquisa, é desenvolvida com base em material já elaborado, constituído principalmente de livros e artigos científicos.

O trabalho caracteriza-se também como de campo, pois para Marconi e Lakatos (2010) “pesquisa de campo é aquela utilizada com o objetivo de conseguir informações e/ou conhecimentos acerca de um problema para o qual se procura uma resposta, ou de uma hipótese”.

3.1. A Escolha do Software

Existe um grande número de motores de jogo, cada um com as suas vantagens e desvantagens e selecionar o mais adequado nem sempre é uma decisão fácil. Ao realizar esta escolha, devem-se considerar alguns fatores importantes: o orçamento disponível, uma vez que existem motores com licenças comerciais e com licenças livres, o tipo de jogo a ser desenvolvido em 2D, 3D, RPG, MOBA e entre outros, visto que um motor de jogo pode adequar-se mais a um determinado tipo de jogo; o tempo disponível para a produção do jogo (alguns motores permitem criar um jogo em poucos dias e outras podem dispor de mais tempo).

Para escolha do melhor software a ser estudado, um survey foi realizada com base em alguns parâmetros para análise comparativa, que foram:

- Proporciona maior agilidade no desenvolvimento de jogos;
- Facilidade de aprendizado;

- Necessidade de utilização de programação;
- Tipo de licença;
- Portabilidade entre plataformas;
- Quantidade de ferramentas prontas.

Foram avaliadas também as seguintes questões, abstração da linguagem de programação, interface e também a licença para o desenvolvimento.

Após o estudo 13 engines foram selecionadas, como mostra a tabela 1:

Tabela 1: Lista de Game Engines

Engine	Linguagem	Licença
Blender Game Engine	C, C++ e Python	BL / GNU GPL
Box2D	C++	Zlib
CELstart	Python e/ou XML	GNU LGPL
Crystal Space	C++	GNU LGPL
Godot Game Engine	GD Script	MIT
Irrlicht	C++	GNU
jMonkey Engine	Java	BSD
Ren'Py Visual Novel Engine	Python	MIT
LÖVE	Lua	zlib/libpng
OGRE	C++	MIT
Panda 2D	C++	BSD
PLIB	C++	LGPL
Torque 2D	C++	MIT

Fonte: Gonçalves (2017), produção própria da autora.

Blender Game Engine

Disponível sob licença dupla BL License / GNU General Public License , Blender também é conhecido como Ketsui, ou Game Blender, se trata de uma plataforma de desenvolvimento de jogos de aplicação de código aberto. Projetado para desenvolvimento de aplicações interativas em 3D, como jogos digitais, projetos arquitetônicos, exposição entre diversas funcionalidades. Possui segmentos licenciados sob Python Software Foundation License. O Blender possui também

multiplataforma estando disponível para Windows, Linux, Mac Os X, Solaris, FreeBSD, IRIX, Darwin, BeOS e iPaq.

Blender é um dos programas mais prático e intuitivo para se utilizar no desenvolvimento de um jogo digital, sem exigência de uso de tipos de linguagens de programação, pois ele disponibiliza diferentes ferramentas sem precisar criar códigos. O Blender Game Engine usa OpenGL para os gráficos, OpenAL para som 3D, Bullet para física e detecção de colisão, e Python para scripts. Suporta nativamente apenas o formato WAV para sons. Ainda sim se for desenvolvido com os scripts de Python , é possível utilizar outros arquivos de som (NUNES, 2009).

Box2D

A Box2D é um motor para simulação de física de corpos rígidos, desenvolvida originalmente em C++ e possui código aberto. Ela possui várias funcionalidades: detecção de colisão contínua, colisão por categorias e grupos, polígonos convexos e círculos, contato fricção e restituição, física contínua com tempo de impacto, pontos de junção, entre outras funcionalidades (BOX2D, 2017).

CELstart

A CELstart é um motor de jogo, que reúne o Crystal Space e o CEL, para desenvolvimento de jogos escritos inteiramente em Python e/ou XML. Não há a necessidade de um compilador C++, para usar o CELstart, pois ele age como um interpretador. Como Crystal Space, CELstart é portável e funciona nas plataformas Microsoft Windows, Linux,, Mac OS X e FreeBSD. CELstart é código aberto, licenciado sob a GNU LGPL (CELstart, 2017).

CRYSTAL SPACE

Crystal Space é um projeto desenvolvido usando C++, como o irrlicht é multiplataformas sendo compatível com Windows, Linux e MacOs. No windows ele suporta o DirectX e em todos sistemas disponibiliza assistência a biblioteca OpenGL. É um programa Open Source desenvolvido por Jorit Tyberghein [CRY 05] [WEN 02]. É compartilhado gratuitamente sob a licença LGPL e não é um engine, mas refere-se a um kit de desenvolvimento para jogos computadorizados 3D (BITTENCOURT, 2005).

Godot Game Engine

Godot é um mecanismo de jogo avançado, funcional, multiplataforma, 2D e 3D. Ele fornece um enorme conjunto de ferramentas comuns, então você pode se concentrar em fazer seu jogo sem reinventar a roda.

A Godot é totalmente gratuita e de código aberto sob a licença MIT Sem restrições, sem royalties, nada. O jogo é do desenvolvedor, até a última linha do código do motor (Godot, 2017).

IRRЛИCHT

Trata-se de uma plataforma orientada a objetos escritos em C++. Um programa de código aberto que possui um funcionamento avançado e elevado. Podendo ser utilizado para desenvolver jogos 2D e 3D, para as diversas aplicações gráficas de aplicação geral. Direcionado para otimização de jogos, logo possui sua parte física de desenvolvimento formada, na qual consiste em diferentes métodos que enfrentam reconhecimento de contradição, porém existem plataformas livres que trabalham juntas, como o-ewton em C++ (AMAZONAS; JUNIOR, 2007).

Atualmente Irrlicht é multiplataforma que executando no Windows, Linux e Mac, se transformando em uma grande portabilidade. Possuindo rapidez de execução e diversas aplicabilidades, tendo potencial de ser utilizado livremente para desenvolvimento de projetos comerciais, seu código é aberto e distribuído gratuitamente sob os termos de autorização GNU - General Public License (NUNES, 2009).

jME (jMonkey Engine)

Seu código fonte é plenamente aberto sob a licença BSD. Podendo vir a ser utilizado como um hobby ou para projetos comerciais. Possui aparência gráfica de alto desempenho, foi desenvolvido para suprir a carência de plataformas de desenvolvimentos de jogos completos, contém gráficos escritos em Java. Seu relevante benefício é que a linguagem mais utilizada é o JAVA, se tornando agradável para programadores que possuem mais habilidades com esse tipo de linguagem, assim com mais facilidade de adaptação com o programa (NUNES, 2009).

Utiliza uma camada de abstração, que possibilita que qualquer sistema de processamento possa ser plugged in. Atualmente, tanto o LWJGL (Lightweight Java Game Library, tal como o nome indica é uma biblioteca de software para programadores de jogos digitais) como os JOGL (Java OpenGL, não é nada mais do que uma biblioteca wrapper que permite que o OpenGL possa ser utilizado na linguagem de programação java) são suportados (NUNES, 2009).

Ren'Py Visual Novel Engine

Ren'Py é um mecanismo de novela visual - usado por milhares de criadores de todo o mundo - que ajuda você a usar palavras, imagens e sons para contar histórias interativas que funcionam em computadores e dispositivos móveis. Estes podem ser tanto romances visuais quanto jogos de simulação de vida. A linguagem de script fácil de aprender permite que qualquer pessoa escreva eficientemente grandes novelas visuais, enquanto o seu script Python é suficiente para jogos de simulação complexos. Ren'Py é de código aberto e gratuito para uso comercial.

LÖVE

A plataforma LÖVE é desenvolvida em C++ e implementa algumas ótimas técnicas de ponta. A LÖVE é uma estrutura que é usada para desenvolvimento de jogos 2D em Lua. A LÖVE está licenciada sob a licença liberal zlib / libpng isso significa que é grátis, open source e multiplataforma, ou seja, para Windows, Mac OS X e Linux (LÖVE, 2017).

OGRE3D (Object Oriented Graphics Rendering Engine)

Plataforma de desenvolvimento mais completa possuindo maior número de *plugins*, para criação de jogos livres. Ogre3D é uma ferramenta formada em C++, orientada a objetos e distribuído livremente e gratuito sob a licença LGPL. Elaborado como dissertação de mestrado de Jeff Plummer na Universidade do Arizona/EUA. Ogre3D como *Crystal Space* consiste em grupo de segmentos que podem ser usados para criação de um *engine*, mas ele não é precisamente um *engine*. Ambos apresentam recursos ao desenvolvedor de extensão e personalização, intensificando o aproveitamento do código, desconsiderando fontes de baixo nível e

possibilitando desenvolvimento de novas técnicas que podem ser inseridas na arquitetura das plataformas (BITTENCOURT, 2005).

Panda 3D

Desenvolvido em C++, contém conjunto de bibliotecas em C++ que dispõe de linguagem *script* para criar estas bibliotecas e ampliando, assim, o grau de abstração no decorrer do desenvolvimento do jogo. Essa particularidade possibilita atingir uma aprendizagem positiva. O Panda3D trabalha a definição de objetos e nodos, sendo um nodo com vários “filhos”, estes “filhos” terão os mesmos atributos que o nodo “pai” como cor, iluminação, transparência e movimento, entre outras características.

Essa plataforma constitui-se de diversas particularidades fundamentais para o desenvolvimento de jogos digitais e diferentes efeitos de sombras (desenho, *deferred*, etc.), efeitos de *motion trails*, dupla visão, distorção do tipo *wings of blue* e efeito redemoinho. Possibilita visualizar o desempenho em tempo real e contém funções para potencializar a desempenho, o mesmo percebe pontos de estreitamento entre o CPU e GPU e os recupera. Dispõe de recursos de *debugging*, com destaque a tolerância de erro com impedimento de falhas. Também reúne diferentes comandos para encontrar erros precoces na elaboração do código (NUNES, 2009).

PLIB

PLIB é um conjunto de bibliotecas livres e de código aberto para facilitar o desenvolvimento de jogos de computador. Foi originalmente escrito por Steve Baker em 1997 e licenciado sob a LGPL. O PLIB inclui efeitos sonoros, música, um mecanismo 3D completo, renderização de fontes, uma biblioteca de janelas simples, uma linguagem de script de jogo, uma interface de usuário gráfica (GUI), rede, biblioteca de matemática 3D e uma coleção de funções de utilidade (Get Alternative PLIB, 2017).

TORQUE 2D

O Torque 2D é um motor extremamente poderoso, flexível e de código aberto, dedicado ao desenvolvimento de jogos 2D. A versão licenciada MIT do Torque 2D

agora está disponível no GitHub. O Torque é multiplataforma sendo elas: OS X, Windows e iOS. O Torque foi desenvolvido em C++ e possui em script próprio o TorqueScript (GarageGames Torque, 2017).

3.2. Construção de conhecimento acerca da *Engine*

O desenvolvimento de jogos tornou-se difundido com a melhoria da interface da game *engines* ou simplesmente os motores de jogo possibilitou a adesão de um público que não possui tanto conhecimento técnico para desenvolvimento de um jogo.

E outro fator de suma importância para impulsionar essas melhorias na interface foi à complicaçāo existente nos motores de jogos. Mesmo que o designer de jogos utilize uma ferramenta que exclua por completa a programação o desenvolvedor ainda teria dificuldades caso fosse utilizar um *engine* cuja interface não fosse dinâmica e intuitiva (OLIVEIRA, 2013).

Para melhor entendimento da ferramenta descreveremos o ambiente de desenvolvimento da Godot Game Engine.

Controlador de Objetos de Cena

Pode-se controlar diversos tipos de *sprites* (objetos), no jogo facilitando a interação com o software. Ao desenvolver na Godot Game Engine, o desenvolvedor terá o comando de todos os objetos, sejam esses, naves, avatares, carros, ou seja, todos os elementos que compõem o jogo. Podendo dá-los interação com a cena, movimentação e criação de espaço dentro do jogo.

Esta janela apresenta algumas funções que alteram as propriedades dos objetos no app (**Ver Figura 3**). Exemplo disso é a função *Animated Sprite 2D* que é responsável pela movimentação e animação dos objetos, seja através de *scripts* ou não, isso dependerá do desenvolvedor e essa função é atribuída facilmente para vários objetos em cena e apesar de estar sendo trabalhando na função 2D da engine, pode-se observar que há sobreposição de objetos, no exemplo citado a bola vermelha (objeto animado), sobrepõe o fundo branco (*background*) e isso pode ser repetida inúmeras vezes, criando várias camadas na cena.

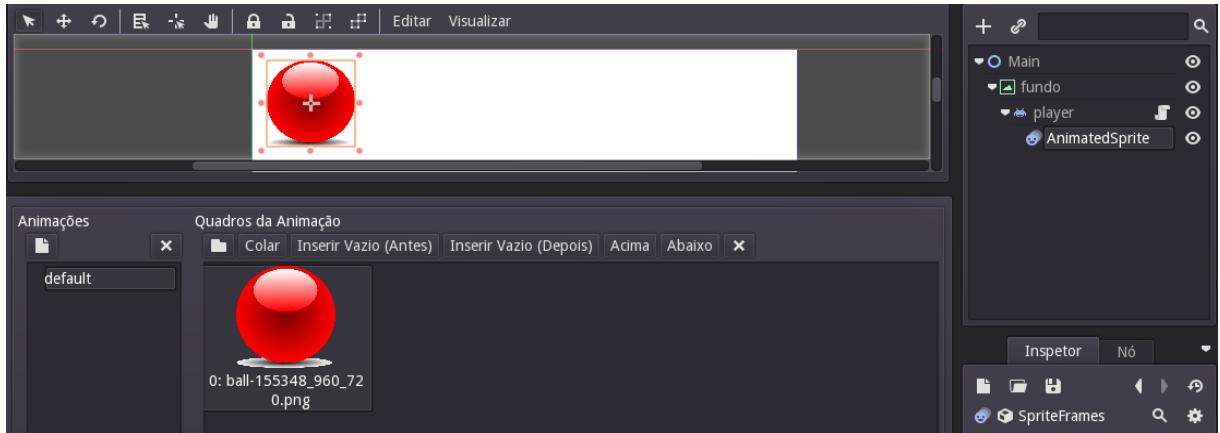


Figura 3: Controlador de Objetos de Cena.

Física e Colisões

Esta ferramenta permite manipular a física e a detecção de movimentos e propriedade dos atores, como velocidade, peso e gravidade do jogo (**Ver Figura 4**). Com a função *Static Body 2D* podemos criar e transformar as propriedades dos atores para que alcancem o maior nível de realidade possível. O objeto tem sua superfície de contato e a colisão, é detectada quando existe a intersecção entre as superfícies de contato de dois ou mais atores ou com a colocação de uma barreira.

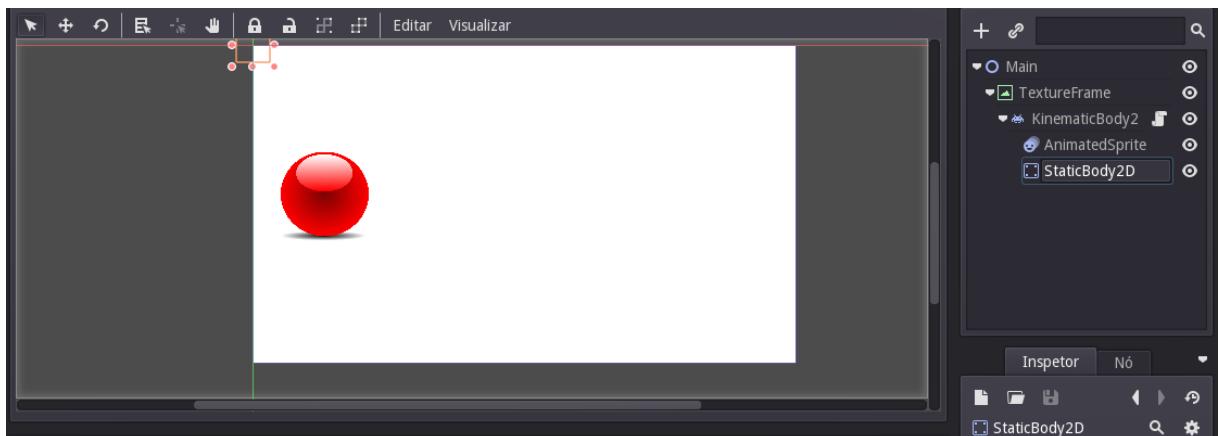


Figura 4: Física e Colisões

Criação de Script

Esse recurso permite que o usuário crie comportamentos e funções personalizadas para seu jogo (**Ver Figura 5**). Particularmente, considero essa a janela mais poderosa da Godot Game Engine. Nesta função o desenvolvedor irá precisar de conhecimento em programação, mais especificamente da linguagem própria de programação da engine o GDScript.

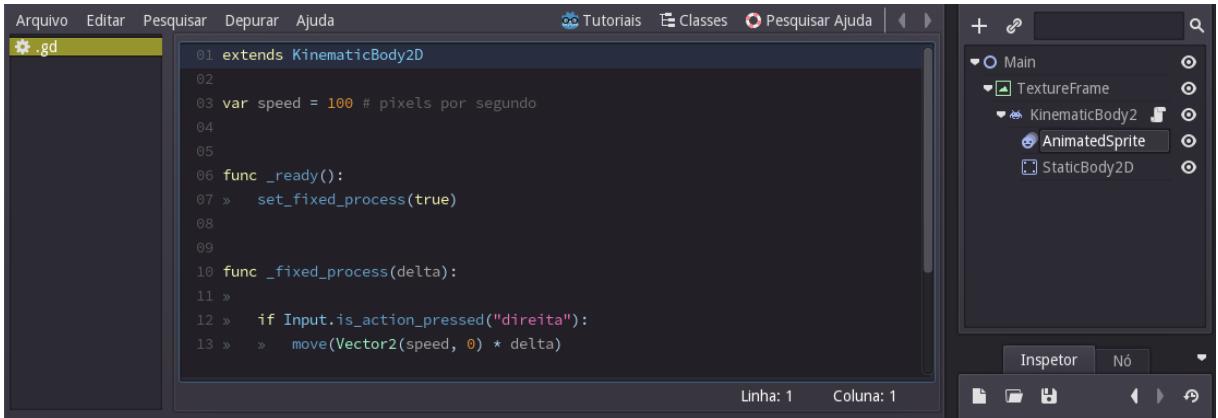


Figura 5: Criação de Script

Após os estudos e pesquisas realizadas a Godot Game Engine foi a escolhida, pois, além de atender todos os pré-requisitos já mencionados, contava ainda com algumas vantagens, como, por exemplo, interface dinâmica, sistema de nós de simples compreensão, linguagem de programação de fácil entendimento mesmo sendo própria da engine, tendo a facilidade também de inserir imagens em vários formatos como: “PNG”, “JPEG”, “JPG” e vetor, podendo ser inseridos também áudio e escrita. E ainda a engine conta com uma quantidade de conteúdo considerável sobre a ferramenta disponível em português, que vai desde documentos em “PDF” oferecido no próprio site da engine até vídeos aulas no YouTube e comunidades no GitHub. E a incomplexidade de nem sempre precisar de programação para a criação dos jogos.

3.2.1. Ensaio de prototipação

A partir dos estudos da engine surgiu a ideia da construção de um protótipo, um app educacional baseado em jogo de tabuleiro com foco no suporte para as séries iniciais do ensino fundamental e ajudaria no ensino da história do município, pois, teria como cenário a cidade de Porto Nacional - TO.

O protótipo, por sua vez, começou a ser desenvolvido (**Ver Figura 6**). O cenário escolhido foi criado e um personagem foi escolhido. Também parte do script de programação que daria movimento ao personagem foi inserida e também parte da física do jogo. Entretanto ao chegar à parte de finalização do app ocorreram alguns contratemplos entre os quais o tempo disponível para o término do jogo.

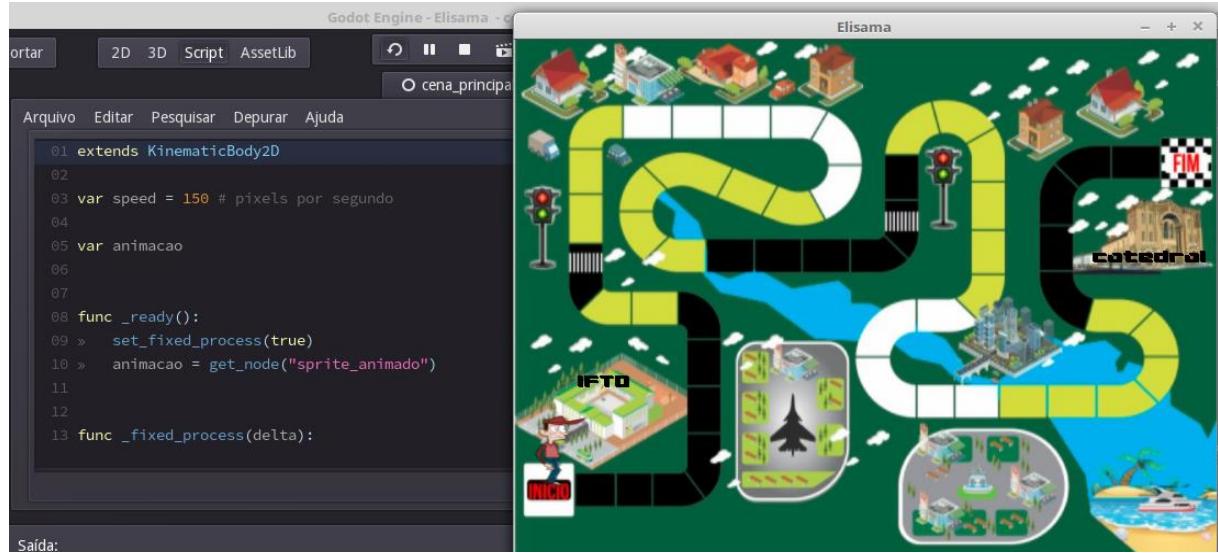


Figura 6: Tela do Ensaio do Protótipo.

Por fim, mesmo sem a finalização do protótipo ao longo do ensaio de sua construção, grande conhecimento foi adquirido a respeito da *engine*, pois, ao ter a possibilidade de entender como é o funcionamento na prática da ferramenta foi de suma importância para construção desse trabalho.

3.2.2. Oficina experimental

Com o objetivo de avaliar preliminarmente a *engine* realizou-se a primeira oficina que foi ministrada somente como um experimento, sem aplicação de questionários, realizada a princípio para que pudesse ser observado o tempo que levaria para a execução da oficina, o tempo disponibilizado foi de duas horas para a realização da oficina e ao final notou-se que o tempo seria insuficiente para o conteúdo que havia sido programado e que na próxima ou necessitariámos de mais tempo ou a oficina devia ser repensada e foi observado também qual seria a primeira impressão do público acerca da *engine*, que seria as principais dificuldades e dúvidas mesmo que o público fosse somente de discentes da área da computação. Para conhecimento do público acerca da ferramenta foi feita uma demonstração na prática da *engine*, sendo mostrados inserção de imagens, movimentação, física e *script* de programação. Como mostra a (Figura 7).

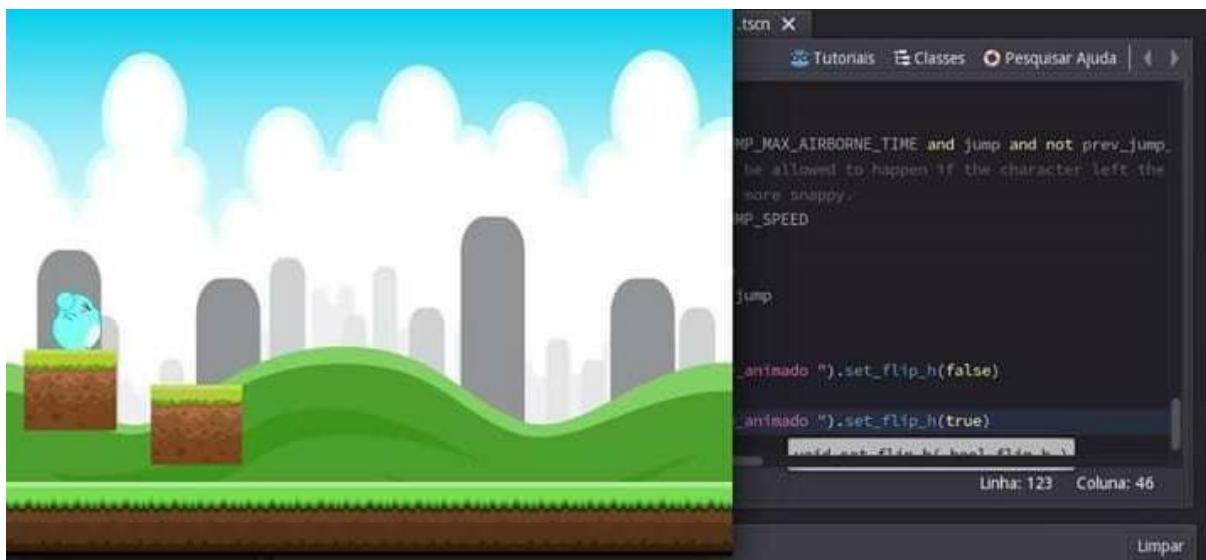


Figura 7: Tela do conteúdo dado na oficina experimental.

A oficina foi realizada no IFTO - *Campus Araguatins*, na V edição do Festival Latino Americano de Instalação de Software Livre (FLISOL), nessa oportunidade houve a demonstração da Engine Godot para 25 alunos, sendo eles do curso de Licenciatura em Computação e do Técnico em Informática. Mesmo sem conhecerem a engine não tiveram grandes dificuldades em trabalhar com ela, mesmo por serem acadêmicos da área de computação.

3.2. Coleta de dados

Como ferramenta de levantamento de dados um questionário foi aplicado após a realização de uma oficina. A ideia de a priori oferecer oficinas ocorreu no pressuposto de oportunizar um maior contato com a ferramenta estuda para quem ainda não a conhecia. A segunda oficina foi realizada no IFTO - *Campus Palmas*, no VI Encontro do Programa Institucional de Bolsas de Iniciação à Docência (PIBID), onde se encontrava acadêmicos dos cursos de Licenciatura em Computação, Biologia, Física e Matemática. O VI Encontro do PIBID foi escolhido por na ocasião estarão presentes diferentes graduandos de licenciaturas diferentes e na situação também estava presente duas professoras, uma do curso de Licenciatura em Computação e outra professora do estado tendo como formação o curso de Pedagogia.

Para Jacobsen (2011, p.28), “na pesquisa, os métodos empregados para coletar os dados precisam estar em plena consonância com os seus objetivos, isto é, devem possibilitar o atendimento da sua demanda”. Então para coleta de dados

dessa pesquisa foi escolhido questionários semiaberto, pois, acredita-se através deles pode-se haver uma melhor amostragem dos resultados esperados pelo pesquisador.

Após ser ministrada a última oficina foram aplicados os questionários para os participantes, tendo como objetivo a avaliação da Godot Game Engine como ferramenta de criação de *apps* educacionais para auxiliar no processo de ensino/aprendizagem em sala de aula e apesar de nem todos terem conhecimento sobre o que é uma *game engine*, nota-se que houve facilidade no manuseio e que todos conseguiram chegar ao resultado esperado.

3.3. Análise

Quanto aos procedimentos de tabulação dos dados, foi realizada o processamento em planilhas e os resultados expressos em gráficos, valendo-se de análise estatística descritiva e cruzamento entre as respostas de alunos e professores com base nas evidências baseadas no *survey* comparativo das engines **4.1**; Evidências baseadas no ensaio de prototipação e oficina I **4.2**; e Evidências baseadas no instrumento de coleta de dados aplicado na oficina II **4.3**.

4. RESULTADOS

Haja vista, que o objetivo desse trabalho é construir conhecimento acerca da ferramenta Godot Game Engine que possibilite uma metodologia ágil de criação de apps educacionais, para consecução dos resultados, realizou-se análise das evidências encontradas ao longo da pesquisa, nas aplicações das oficinas, no ensaio de prototipação e questionários aplicados aos docentes e discentes.

4.1. Evidências baseadas no survey comparativo das engines

Nas evidências baseadas no survey comparativos das engines ficou claro a motivação em usar a Godot Game Engine como ferramenta de estudo, haja vista que seus atributos obtêm na facilidade no uso, no sentido e que apesar de haver a necessidade de programação a ferramenta permite fazer isso de forma intuitiva e simples, além da linguagem ser de fácil aprendizagem e a *engine* possui alto nível de flexibilidade combinado com a quantidade de funcionalidades pré-dispostas. Percebe-se ainda que uma das maiores vantagens que a ferramenta é totalmente gratuita e multiplataforma.

Devido ao fato da ferramenta escolhida atender todos os pré-requisitos anteriormente citados no item **3.1**, estipulados ainda no início da pesquisa, conclui-se que a **Hipótese I**: A Godot Game Engine apresenta-se como uma notável e ágil ferramenta para o desenvolvimento de apps educacionais, é válida às evidências baseadas no survey comparativo das engines.

4.2. Evidências baseadas no ensaio de prototipação e oficina I

Ao longo da construção do conhecimento acerca da engine acreditava-se que desenvolvimento de apps seria de forma rápida e sem muitas dificuldades, já que a *engine* se mostrava intuitiva e de fácil manipulação. Mas baseadas nas evidências no ensaio de prototipação e na oficina I, percebeu-se que a demanda de tempo foi um dos maiores empecilhos para a finalização, tanto do protótipo, quanto da oficina I, pois a grande quantidade de recursos a ser estudados e apresentados demandou muito tempo, pois a *engine* é uma ferramenta completa, contendo vários recursos e alguns com certa complexibilidade de entendimento. Portanto, deve-se salientar que

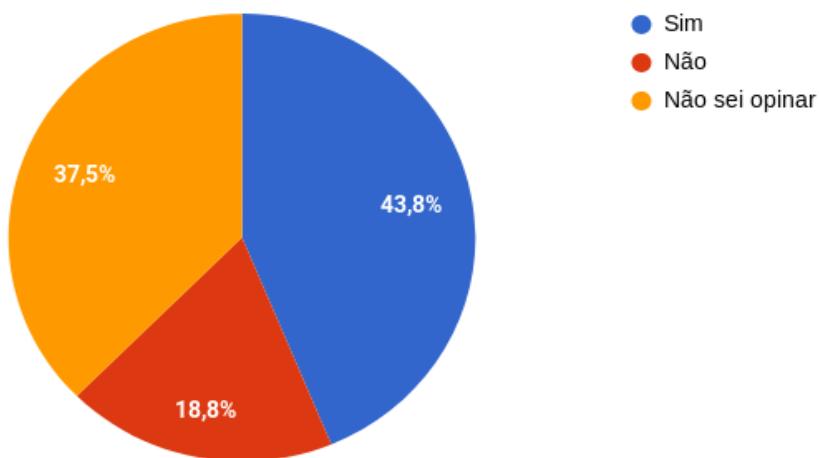
a **Hipótese III:** A interface intuitiva e de fácil aprendizagem da Godot Game Engine contribui para o desenvolvimento de *apps* educacionais em pouco tempo, mesmo com a necessidade de programação. Foi refutada parcialmente, pois, mesmo sendo de fácil aprendizagem e com interface intuitiva teve a duração maior que o previsto para o desenvolvimento.

4.3. Evidências baseadas no instrumento de coleta de dados aplicado na oficina II

Para a obtenção dos resultados primeiramente foi realizada uma oficina para que o público tivesse conhecimento acerca da ferramenta e conhecesse seus recursos podendo assim ter uma opinião sobre a *engine*, após a oficina, houve a aplicação de um questionário semiaberto. Baseando-se na oficina experimental e sabendo que a durabilidade da oficina seria a mesma da primeira, então algo mais simples foi apresentado aos participantes, sem deixar de ser apresentado nenhum recurso da *engine*, como por exemplo, Inserção de sprites (objetos), física, colisão e script de programação. Durante a realização da oficina mesmo com o surgimento de dúvidas, todos conseguiram realizar as atividades propostas mesmo sendo um público misto. Devido ao fato que o público entrevistado encontrou vantagens na *engine* para desenvolvimento de *apps* educacionais e mesmo alguns participantes não sendo da área da computação obtiveram sucesso no desenvolvimento, enfatiza-se a validação da **Hipótese II:** A *engine* selecionada apresenta diversas vantagens na produção de *apps* educacionais, para qualquer profissional da área docente.

Por fim, para obtenção da amostra de resultados um questionário semiaberto foi aplicado após a realização de uma oficina a docentes e discentes, contendo perguntas que abrangia o conhecimento obtido pelos mesmos durante a oficina, a mesma foi realizada no VI Encontro Anual do Programa Institucional de Bolsas de Iniciação à Docência (PIBID), realizado no Campus do IFTO - Palmas, no dia 24 de novembro de 2017. Dentre esses 18,8% eram docentes, sendo que 33,3% pedagogos e 66,7% da área da computação e 81,2% discentes, sendo que 50% eram alunos do curso de Licenciatura em Computação, 21,4% do curso de Licenciatura em Matemática, 14,3% do curso de Licenciatura em Biologia e 14,3% do curso de Licenciatura em Física.

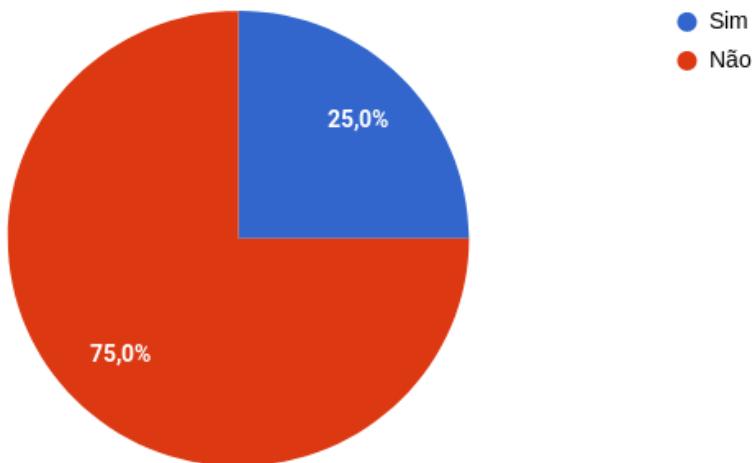
Gráfico 1. Você conhece alguma ferramenta que considere eficiente para o desenvolvimento de apps educacionais?



Fonte: Gonçalves (2017), produção própria da autora.

Para termos conhecimento do público com o qual estava sendo trabalhando e qual o nível de instrução sobre ferramentas de desenvolvimento, pergunta-se se eles já tinham conhecimento acerca de alguma ferramenta, no qual se teve a soma de 43,8% **Gráfico 1** de afirmativas, questiona-se também, quais seriam essas. Tendo o resultado de que 50,0% dos entrevistados afirmaram conhecer o software “MIT App Inventor”, que “trata-se de um software web que permite desenvolvimento de aplicativos Android, usando um navegador da Web e um telefone ou emulador conectados” (AndroidPro, 2018). E ainda 37,5% dos participantes citam o “Scratch”, essa ferramenta “trata-se de um software que se utiliza de blocos lógicos, e itens de som e imagem, para o desenvolvimento de histórias interativas, jogos e animações, além de compartilhar de maneira online toda as criações (Scratch Brasil, 2018). E por fim, 12,5% do público afirmam ter conhecimento sobre a ferramenta “Blender”, que “trata-se de um conjunto de criação 3D gratuito e de código aberto. “Ele suporta a totalidade do molde de pipeline 3D, manipulação, animação, simulação, renderização, composição e rastreamento de movimento, até edição de vídeo e criação de jogos” (Blender, 2018). Estes softwares citados, mesmo se aplicando a categoria de ferramenta de desenvolvimento não se encaixam como motor de jogos e isso os diferem da ferramenta estudada nesse trabalho.

Gráfico 2: Você já tinha algum conhecimento sobre a ferramenta Godot ?



Fonte: Gonçalves (2017), produção própria da autora.

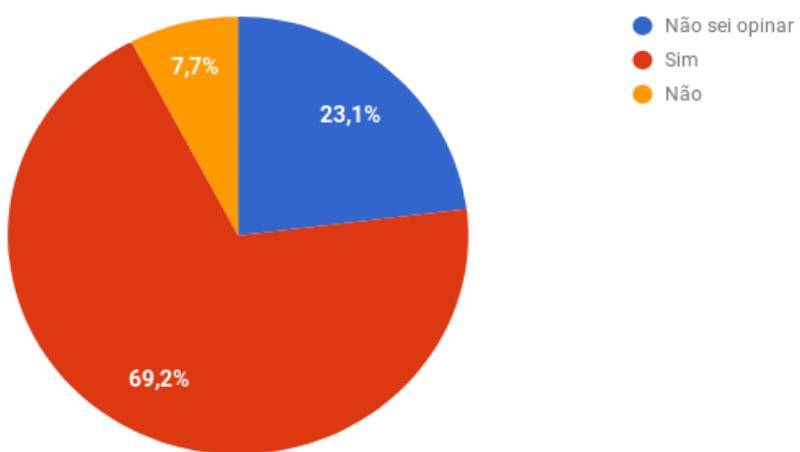
A Godot Game Engine mesmo que já esteja se popularizando entre os desenvolvedores de jogos, ainda é pouco conhecida pelo público em geral, como demonstrado no **Gráfico 2**, que apenas 25,0% do público entrevistado tinha conhecimento sobre a ferramenta e 75,0% afirmam não conhecer. E dentre esses que já tinha um conhecimento prévio da ferramenta, 50,0% o adquiriu com colegas e os outros 50,0% adquiriu através da pesquisadora na oficina experimental realizada no Festival Latino-americano de Instalação de Software Livre (FLISOL), no campus do IFTO - Araguatins, em abril de 2017. E o resultado foi surpreendente, que mesmo em sua maioria, o público ainda não conhecendo a ferramenta estuda ou outra parecida, o desempenho foi sem tamanhas dificuldades. Concluindo-se assim que a **Hipótese I** é válida a essa questão já que a Godot Game Engine se apresentou como uma notável e ágil ferramenta de desenvolvimento de *apps* educacionais.

Após a realização da oficina II foi indagado a seguinte questão aos participantes: Você acredita que é possível fomentar a produção de *apps* educacionais com a utilização da Godot Game Engine?

Há uma unanimidade na resposta, onde 100,0% afirmam que sim. E isso também se mostra verdade, pois, ao decorrer da oficina todos conseguiram com êxito a finalização da atividade proposta, mesmo havendo a necessidade programação e parte do público não sendo da área da computação. Nota-se então que a **Hipótese II** é válida a essa questão, já que a *engine* fomentou a produção de *apps* educacionais em público onde se encontrava pessoas de licenciaturas diversas.

Gráfico 3: Você acredita que esse tipo de ferramenta fornece alguma vantagem dentro da sua área de formação?

Um dos intuios deste trabalho era avaliar se a ferramenta estudada seria útil para docentes de qualquer área de formação e quando perguntado se a ferramenta oferece alguma vantagem dentro da área de formação dos entrevistados, 69,2% afirmam que sim, 23,1% não sabe opinar e 7,7% acha que não, como mostra o gráfico 3, atentando-se então que a **Hipótese I**, se reafirma nessa questão já que em sua maioria os entrevistados acreditam que a ferramenta oferece vantagem a sua área de formação.



Fonte: Gonçalves (2017), produção própria da autora.

E ainda no caso de respostas afirmativas perguntamos aos entrevistados quais seriam essas vantagens obtidas para sua área de formação, obteve-se então as seguintes respostas. Ver Tabela 2:

Tabela 2: Resposta dos entrevistados

Entrevistado 1:	Sim, pois através de aplicativos como o Godot podemos criar jogos educacionais que contribuam para o melhor ensino por trazer uma forma lúdica para o processo de aprendizagem.
Entrevistado 2:	Essa ferramenta pode ser útil na criação de jogos matemáticos, muito utilizados hoje para melhoria do processo de aprendizagem.
Entrevistado 3:	Sim, pois como licenciando poderei aplicá-lo na educação como

	ferramenta de ensino.
Entrevistado 4:	A ferramenta pode ser útil para criação de jogos de modo que posso aplicá-lo melhorando assim o desempenho educacional dos meus alunos.

Fonte: Gonçalves (2017), produção própria da autora.

Diante a resposta aberta dos entrevistados nota-se que os mesmos acreditam que a ferramenta apresenta vantagens dentro da sua área de formação, de forma que ela pode ser útil para criação de *apps* educacionais para a melhoria do ambiente educacional.

E para alcançarmos um melhor resultado acerca da *engine* trabalhada foram aplicadas questões abertas aos entrevistados, sendo estas.

Quais foram às maiores dificuldades durante o desenvolvimento de *apps* com utilização da Godot?

Tabela 3: Resposta dos entrevistados

Entrevistado 1:	A plataforma é em inglês dificulta um pouco o manuseio
Entrevistado 2:	Dificuldades por não ter conhecimento do assunto
Entrevistado 3:	É uma ferramenta de fácil acesso, mas tive dificuldades por nunca ter tido nenhum contato com a ferramenta.
Entrevistado 4:	Não ter conhecimento na área de programação.
Entrevistado 5:	Como não tinha conhecimento dela (Godot), tive um pouco de dificuldades, porém a professora bem atenciosa sanou todas as dúvidas.
Entrevistado 6:	Não tive dificuldades, a Godot tem fácil utilização. Apesar de ser em inglês é bem intuitivo e parecido com outros programas que uso.

Fonte: Gonçalves (2017), produção própria da autora.

Já se esperava que dificuldades surgissem pelo fato do público entrevistado ser misto e por se tratar de uma ferramenta de pouco conhecimento pelos mesmos, mas alguns dos entrevistados não encontraram nenhuma dificuldade para outros a

maior dificuldade foi a ferramenta não está disponível em português, recurso já disponível para o sistema operacional Linux, entretanto no dia da realização da oficina não tinha laboratório de informática disponível com o sistema Linux.

Quais os maiores benefícios percebidos após uma primeira impressão da Godot?

Tabela 4: Resposta dos entrevistados

Entrevistado 1:	Por ser utilizado como auxílio pedagógico.
Entrevistado 2:	Percebi que é uma ótima ferramenta que pode ser utilizada na educação pode ser trabalhada em várias disciplinas.
Entrevistado 3:	Mais uma ferramenta para ensinar os conteúdos trabalhados nas escolas
Entrevistado 4:	Acredito que com essa ferramenta é possível criar apps que possam melhorar o processo de ensino/aprendizagem. É uma ferramenta de fácil acesso e manuseio.
Entrevistado 5:	É uma forma fácil, rápida e eficiente de programar jogos que ajudariam na sala de aula.
Entrevistado 6:	A linguagem de programação da engine ser parecida com linguagem C.

Fonte: Gonçalves (2017), produção própria da autora.

Ao analisarmos as afirmações dos entrevistados nota-se que a **Hipótese III** é válida, pois mesmo com a necessidade de programação e a demanda de tempo para o desenvolvimento a *engine* no ponto de vista dos entrevistados se mostrou intuitiva e de fácil acesso e pela linguagem de programação da ferramenta ser parecida com uma linguagem que eles já conhecem também foi um ponto positivo para quem já tem conhecimento na área.

Quais as maiores desvantagens percebidas após a primeira impressão da Godot?

Tabela 5: Resposta dos entrevistados

Entrevistado 1:	Não percebi nenhuma desvantagem.
------------------------	----------------------------------

Entrevistado 2:	Não ser em português.
Entrevistado 3:	O fato de não ser oferecido em português.
Entrevistado 4:	Não encontrei desvantagens. É um programa bem eficiente e eficaz.
Entrevistado: 5	Por estar em inglês, mas as dificuldades diminuíram com o uso.

Fonte: Gonçalves (2017), produção própria da autora.

Para a maioria dos entrevistados afirmaram que a questão da ferramenta ser na língua inglesa é uma das maiores desvantagens da *engine*, mas como já esclarecido a *engine* já está disponível em português, por outro lado os entrevistados não encontraram maiores dificuldades, pelo contrário considerou a mesma eficiente para o que foi proposto.

5. CONSIDERAÇÕES FINAIS

Esse trabalho teve como objetivo a construção de conhecimento acerca da ferramenta Godot Game Engine que possibilite uma metodologia ágil de criação de apps educacionais. A criação de apps educacionais utilizando a *engine* se mostrou na prática que com conhecimento básico a respeito de lógica de programação, ou até mesmo sem ele, já que a *engine* apresenta o recurso de abstração da programação e também pouco de conhecimento de jogos e com dedicação de tempo, qualquer interessado poderá desenvolver um app.

Com base nos estudos realizados, observou-se que a utilização da Godot Game Engine torna o desenvolvimento de jogos mais ágeis, pois possui uma quantidade notável da ferramenta preestabelecida, mesmo se fazendo necessária a utilização de programação.

Pode-se observar que a *engine* mostrou-se uma ferramenta intuitiva, com uma interface amigável, com recursos diversos que atende o propósito da *engine*, que é a criação de apps sem muita complexidade.

A ferramenta mostrou-se eficaz quanto a disponibilidade de recursos e de interação entre o usuário. Quanto às vantagens observadas na utilização da ferramenta para o ambiente educacional, a *engine* apresentou-se como um recurso que pode ser empregado de forma ágil na educação, como se pode notar nos resultados deste trabalho.

Observa-se também que a *engine* mostrou-se eficiente para ser trabalhada em qualquer área docente, seja para profissionais que já tenha conhecimento de programação ou não, já que há facilidade na aprendizagem da linguagem de programação e também pelo fato de haver bastante material já disponível nas comunidades da ferramenta e no GitHub.

Conclui-se então a importância da usabilidade de novas ferramentas em sala de aula como um novo recurso contribuindo assim ao processo de ensino/aprendizagem. Apresentada aos profissionais docentes a uma nova forma de criação de objetos de aprendizagem e nova maneira de se pensar a educação reenviando o modo de trabalhar a tecnologia em sala de aula e fugindo do modo tradicional de educação e mostrando ao mesmo que é possível repensar a educação de forma mais criativa e intuitiva tornando o processo de ensino mais agradável não somente para os alunos mas também para o professor.

5.1. Limitações da Pesquisa

De modo geral a pesquisa apresentou-se interessante, inovadora e relevante ao curso de Licenciatura em Computação, pois se tratava de um recurso novo para inserção na educação.

A escassez de pesquisas na área para fundamentação do trabalho tornou-se uma das limitações desta pesquisa, pois é de grande relevância trabalhos realizados anteriormente para embasamento do atual.

Falta de documentos sobre a *engine*: A Godot Game Engine trata-se de uma ferramenta de desenvolvimento de jogos ainda pouco conhecida no mercado, devido a isso a disponibilidade de documentos sobre a mesma ainda se é escassa.

A limitação de documento em português acerca da *engine*: A ferramenta por se tratar de uma ferramenta de uma empresa argentina, a documentação ainda é quase encontrada em sua totalidade em espanhol ou em inglês, já se encontram vários documentos em português devido às variadas comunidades da ferramenta no GitHub e em outras plataformas, mas alguns documentos ainda estão em português de Portugal.

O desconhecimento acerca da *engine* estudada ou de outros recursos parecidos: Uma das maiores limitações foi explicar ao público entrevistado do que se tratava a Godot Game Engine, pois, a maiorias das pessoas não conheciam a mesmas ou nenhum recurso parecido e/ou não sabia nem do que se tratava uma *game engine*.

A dificuldade na coleta de dados: O tempo foi um dos limitadores na coleta de dados. Para conhecimento da ferramenta oficinas foram realizadas e a mesmo demandou muito tempo, limitando assim o tempo dos entrevistados para responderem os questionários, que por sua vez na pressa de entregar logo ou por falta de tempo muitas perguntas ficaram sem respostas.

5.2. Trabalhos futuros

O estudo possibilita o avanço desse campo através das seguintes abordagens:

Proposta 1: Disseminação da ferramenta através de desenvolvimento de tutoriais e protótipos de *apps*, voltados para a educação, no intuito de demonstrar a

capacidade da Godot Game Engine e da sua aplicabilidade no processo de ensino/aprendizagem.

Proposta 2: O desenvolvimento de *App Educativo* com base na Godot *Game Engine* que traga um dinamismo maior na aprendizagem para alunos especiais, pode ser voltado para o ensino de Libras ou para alunos com deficiências auditivas, por exemplo. Podendo conter fases de assimilação do novo vocabulário, com imagens e sons.

Proposta 3: Surgimento de uma comunidade de desenvolvimento de *apps* para a educação livres e *open source*, disponibilizado no GitHub e em outras plataformas.

REFERÊNCIAS

AndroidPro [Online] [Citado em: 01 de jan de 2018]. Disponível em: <http://www.androidpro.com.br/app-inventor/#introducao_ao_app_inventor>.

ALVES, Lynn. Relações entre os jogos digitais e aprendizagem: delineando percurso. Educação, Formação & Tecnologias, v. 1, n. 2, p. [3-10], 2008.

ALVES, João, COUTINHO, João, COSTA, José. et al. Desenvolvimento de Jogos 2D. Criação de Desenvolvimento de Jogos 2D. 2013, p.28.

ALMEIDA, Rubens Queiroz de Almeida. In: Software Livre na Educação. Disponível em: <<http://www.dicas-l.com.br/dicas-l/20000205.php>>. Acesso em: 19 de dez de 2017.

AMAZONAS, S. D.; JUNIOR, Glaudiney Moreira Mendonça. Desenvolvimento de Jogos 3D em Java com a Utilização do Motor Gráfico Irrlicht, 2007.

Blender. [Online] [Citado em: 01 de Novembro de 2017]. Disponível em: <<http://www.blender3d.org/e-shop/>>

BITTENCOURT, João Ricardo. Promovendo a ludicidade através de jogos livres. XV Simpósio Brasileiro de Informática na Educação, p. 43-63, 2005. BOX2DJS, 2008.

CELstart. [Online] [Citado em: 18 de dez de 2018]. Disponível em: <<https://blenderartists.org/forum/showthread.php?137310-Celstart>

COSTA, Edes Garcia da, PENTEADO, Rosângela; SILVA. et al. Padrões e Métodos Ágeis: Agilidade no Processo de Desenvolvimento de Software. 2005. Disponível em: <<http://sugarloafplop2005.icmc.usp.br/papers/9673.pdf>>. Acesso em: 25 mar 2017.

COSTA, Edes Garcia da. Métodos Ágeis. 2006. Disponível em: <<http://www.dc.ufscar.br/~junia/MetAgEds.pdf>>. Acesso em: 30 mar 2007.

CLUA, Esteban Walter Gonzalez; BITTENCOURT, João Ricardo. Desenvolvimento de jogos 3D: concepção, design e programação. In: XXIV Jornadas de Atualização em Informática (JAI) Part of XXIV Congresso da Sociedade Brasileira de Computação. 2005. p. 22-29.

FALKEMBACH, Gilse A. Morgental. O lúdico e os jogos educacionais. Mídias na Educação. Disponível em: http://penta3.ufrgs.br/midiasedu/modulo13/etapa1/leituras/arquivos/Leitura_1.pdf. Acesso em: 10/08/2017

FAGUNDES, Priscila Bastos. Framework para Comparaçao e Análise de Métodos Ágeis. Santa Catarina. Florianópolis: 2005.

FARIAS, Anália Rodrigues de. O desenvolvimento da criança e do adolescente segundo Piaget. Ed. Ática, 3º edição, 1995.

FSF - Fundação de Software Livre [Online] [Acesso em: 19 de dez de 2017]. Disponível em: <<http://www.fsf.org/about/>>.

FORTUNA, Tânia Ramos. Sala de aula é lugar de brincar. Planejamento em destaque: análises menos convencionais. Porto Alegre: Mediação, p. 147-164, 2000.

GALLOTTA, Alexandre Objetos de aprendizagem a serviço do professor. Disponível em <http://www.microsoft.com/brasil/educacao/parceiro/objeto_texto.mspx>. Acesso em 10 de maio de 2017.

GALVIS, A. H. - Ambientes de Enseñanza Aprendizaje Enriquecidos con Computador. Boletín de Informática Educativa, p. 117 - 139. Bogotá, 1988.

Garagem Games: [Online] [Acesso em 18 de dez de 2017] Disponível em: <http://www.garagegames.com/products/torque-2d>.

Get Alternative. [Online] [Acesso em: 19 de dez de 2017]. Disponível em: <<http://getalternative.net/software/plib>>

Godot Game Engine. [Online] [Acesso em: 17 de dez de 2017.] Disponível em: <<https://godotengine.org/>>

GIL, Antonio Carlos. Como elaborar projetos de pesquisa. 4. Ed. São Paulo: Atlas, 2008

GUTIERREZ, Suzana de Souza. Distribuição de conteúdo e aprendizagem on-line. RENOTE - Revista Novas Tecnologias na Educação, v. 2, p. 1-14, 2004.

HILMAN, Jonh. Metodologias Ágeis. Disponível em: <http://www.redes.unb.br/material/ESOO/Metodologias%20%c1geis.pdf>. Acesso em: 20 abr 2007.

HUIZINGA, J. Homo Ludens: o jogo como elemento da cultura. 5o. ed. [S.I.]: Perspectiva, 2003. p. 256.

HUIZINGA, J. Homo Ludens. Tradução de J. P. Monteiro. São Paulo, Perspectiva, 1971.

IEEE Comitê Padrão de Tecnologia de Aprendizagem. Metadados de Objetos de Aprendizagem Disponível em <<http://ltsc.ieee.org/wg12/>> acesso em 10 maio de 2017.

IWASAKI, Eliane Yumi. Movimento Open Source. A importância da comunicação e da relação entre empresas e comunidades para o mercado. São Paulo - SP, 2008. Disponível em:<https://www.dicas-l.com.br/download/movimento_open_source.pdf>. Acesso em: 01 dez de 2017.

JACOBSEN, Alessandra de Linhares. Metodologia Científica: (Orientação ao TCC). Florianópolis: Universidade Federal de Santa Catarina, 2016. Disponível em:

<<http://cursodegestaoelideranca.paginas.ufsc.br/files/2016/03/Apostila-ao-TCC.pdf>>. Acessado em: 01 de nov de 2017.

LÖVE. [Online] [Acesso em: 10 de dez de 2017] Disponível em:<<https://love2d.org/>>.

LUCENA, M. A Gente é uma Pesquisa: Desenvolvimento Cooperativo da Escrita Apoiado pelo Computador; Departamento de Educação, PUC-Rio; Rio de Janeiro: 1992. Disponível em:< <http://www.seer.ufrgs.br/renote/article/view/13629>>. Acesso em 10 de set de 2017.

LUCCHESE, Fabiano; RIBEIRO, Bruno. Conceituação de jogos digitais. Disponível em:<<http://www.dca.fee.unicamp.br/~martino/disciplinas/ia369/trabalhos/t1g3.pdf>> Acesso em: 12 de set de 2017.

LÜDKE, Menga; ANDRÉ, Marli EDA. Pesquisa em educação: abordagens qualitativas. Disponível em: <http://www.lite.fe.unicamp.br/papel/2003/>. Acesso em: 25 de out de 2017.

KISHIMOTO , Tizuko M.O Jogo e a Educação Infantil . São Paulo: Ed. Pioneira, 1994.

KOPER, R. Combinando recursos de aprendizagem reutilizáveis a unidades de aprendizagem pedagógicas propositais. Londres: Ed. Kogan Page, 2003.

MALUF, Ângela Cristina Munhoz. BRINCAR, Prazer e Aprendizado. 3^a ed. Petrópolis: Vozes, 2004.

MARCONI, Marina de Andrade, LAKATOS, Eva Maria. Fundamento de Metodologia Científica. 7. ed. São Paulo, Atlas, 2010.

MARINHO, Herminia Regina Bugeste. Pedagogia do movimento universo lúdico e psicomotricidade. 2. ed. Curitiba: IBPEX, 2007.

MENDES, R. M.; SOUZA, V. I.; CAREGNATO, S. E. A propriedade intelectual na elaboração de objetos de aprendizagem. In: Cinform – Encontro Nacional de Ciência da Informação, 5. 2004, Salvador. Anais, Salvador: UFBA, 2004.

MICHELAZZO, Paulino. Software livre e inclusão digital. São Paulo: Conrad Editora do Brasil, 2003.

MOYLES, Janet R. Só brincar? O papel do brincar na educação infantil. Tradução: Maria Adriana Veronese. Porto Alegre: Artmed, 2002.

NETO, Oscar Nogueira de Souza. Análise Comparativa das Metodologias de Desenvolvimento de Softwares Tradicionais e Ágeis. Trabalho de Conclusão de Curso. Universidade da Amazônia. Belém: 2004.

NUNES, Luiz. Motores para Desenvolvimento de Jogos Digitais. Relatório da Disciplina de Projeto. Engenharia Informática. Escola Superior de Tecnologia e de

Gestão. Disponível em:<<http://projinf.estig.ipb.pt/~LuisNunes/relatorio.pdf>> Acesso em: 10 de set de 2017.

OLIVEIRA, Esdras Rocha de. O uso de engines para o Desenvolvimento de Jogo eletrônico. Disponível em: <<http://www2.uesb.br/computacao/>> Acesso em: 19 de dez de 2017.

Panda 3D. [Online] [Acesso em: 08 de Novembro de 2017.] <http://www.panda3d.org/>.

PASSERINO, L. M. Avaliação de jogos educativos computadorizados. Taller Internacional de Software Educativo 98 – TISE' 98. Anais. Santiago, Chile, 1998.

PETRILLO, 2008) PETRILLO, F.S. Práticas ágeis no Processo de Desenvolvimento de Jogos Eletrônicos. 168 f. Dissertação (Mestrado em Computação) – Universidade Federal do Rio Grande do Sul. Orientador: Dr. Marcelo Soares Pimenta. Porto Alegre – RS, 2008.

PRESSMAN, R.S. Engenharia de Software – Abordagem de um praticante. Porto Alegre: Pearson Makron Books. 5º. ed, 2001.

PRESSMAN, R. S. Engenharia de Software. Porto Alegre: Pearson Makron Books. 7º. ed. 2011.

PINHEIRO, Walter. Software livre e inclusão digital. São Paulo: Conrad Editora do Brasil, 2003.

RAVELLI, Thadara Garcia. O Jogo como Oportunidade de Aprendizagem em Ciências Biológicas. Disponível em: <<http://www.mackenzie.br/>> Acesso em: 02 de jan de 2018.

Ren'Py Visual Novel Engine. [Online] [Acesso em: 19 de dez de 2017] <Disponível em: <<https://www.renpy.org/>>

SAVI, Rafael; ULBRICHT, Vania Ribas. Jogos digitais educacionais: benefícios e desafios. RENOTE, v. 6, n. 1, 2008.

SEGUNDO, Volmar P.; MOURA, Alexandre M. Introdução de Metodologias Ágeis de Desenvolvimento de Software nos Currículos de Referência do Ensino Universitário. Ed Fontes. São Paulo, 2004.

SILVA, Renildo Franco da. Novas tecnologias e educação: A evolução do processo e ensino e aprendizagem na sociedade contemporânea. Disponível em: <<http://www.fvj.br/revista/wp-content/uploads/2014/12/2Artigo1.pdf>> Acesso em: 21 de dez de 2017

SILVA, IK de O.; MORAIS II, Marçal José de Oliveira. Desenvolvimento de jogos educacionais no apoio do processo de ensino-aprendizagem no ensino fundamental. Holos, v. 5, 2011.

SINGH, H. Introduction to Learning Objects. 2001. Disponível em: <www.imsproject>

org/content/packing/ims-cp-bestv1p1.html>. Acesso em 21 de set de 2017.

SOARES, Michel dos Santos. Comparação entre Metodologias Ágeis e Tradicionais Para o Desenvolvimento de Software. 2004.

Disponível em: <<http://www.dcc.ufla.br/infocomp/artigos/v3.2/art02.pdf>>. Acesso em: 11 de jan de 2017.

STANG, Bendik. Game engines: Características e possibilidades. Instituto de Informática e Modelagem Matemática na Universidade Técnica da Dinamarca, 2003. Disponível em: <<http://www.acervodigital.ufpr.br/>>. Acesso em: 01 de nov de 2017.

SCHMITT, Peterson Ricardo Maier. Aplicação web utilizando API GOOGLE MAPS. Disponível em:<http://repositorio.roca.utfpr.edu.br/jspui/bitstream/1/_06.pdf> Acesso em: 19 de dez de 2017.

Scratch Brasil. [Online] [Acesso em: 01 de Jan de 2018] Disponível em: <<http://www.scratchbrasil.net.br/index.php/sobre-o-scratch.html>>

TORI, R.; BERNARDES, J.; BIANCHINI, R.; CUZZIOL, M.; JACOBER, E.; NAKAMURA, R. Jogos Eletrônicos e Realidade Virtual In: TORI, R.; KIRNER, C. (Eds.) Realidade Virtual: Conceitos e Tendências – livro do pré-simpósio - VII Symposium on Virtual Reality, São Paulo: Mania de Livro, 2004. Cap.10, p.159-176.

VALENTE, José Armando O professor no ambiente LOGO: Formação e atuação Campinas, SP: UNICAMP/NIED, 1996.

WILEY, D. (ed.). The instructional use of learning objects, 2000. Disponível em: <<http://reusability.org/read/chapters/bannan-ritland.doc>>. Acesso em 02 de maio de 2017.

XAVIER, Danilo Domingos. Software livre na educação. Disponível em: <<http://www.ic.ufmt.br/sites/default/files/field/pdf/Monografia/DaniloXavier.pdf>> Acesso em: 19 de dez de 2017.

ZANATTA, Alexandre. xScrum: Uma Proposta de Extensão do xScrum para Adequação ao CMMI. Dissertação de Mestrado. Universidade Federal de Santa Catarina. Florianópolis: 2004.

APÊNDICE A – PLANO DE ENSINO OFICINA DA I

PLANO DE ENSINO

1. IDENTIFICAÇÃO

Cidade:	Araguatins – TO
Tema:	Godot Engine: Um motor de jogos 2D e 3D livre e <i>Open Source</i>
Local:	Instituto Federal de Educação, Ciências e Tecnologia do Tocantins – Campus Araguatins.

1.2. Curso

Curso	Oficina com licenciandos em computação e alunos do curso técnico em informática.
--------------	--

1.3. Data

Dia e Horário	29/04/2017 – 14h00min
----------------------	-----------------------

1.4. Ministrante

Ministrante	Elisama Martins Gonçalves
--------------------	---------------------------

2. COMPETÊNCIAS

- Conhecer a Godot Game Engine;
- Apresentar a possibilidade de criação de apps educacionais e a inserção do mesmo em sala de aula;
- Criar um exemplo prático de um app através da ferramenta Godot.

3. HABILIDADES

- Identificar os principais recursos da engine;
- Apresentar a importância da ferramenta para a educação.

4. CONTEÚDO PROGRAMÁTICO (E/OU BASE TECNOLÓGICA):

1ª Parte:

- Conceituar o que é a Godot Game Engine;
- Surgimento da ferramenta;

- Características principais;

2^a Parte:

- Oficina prática para apresentação de recursos e criação de um exemplo de jogo.

5. METODOLOGIA

A oficina será ministrada de forma expositiva e interativa por meio de apresentação de slides e discussão com os participantes.

A primeira etapa será voltada para conceituação e explicação sobre a ferramenta.

A segunda etapa será voltada para desenvolvimento de exemplo prático e esclarecimento de dúvidas acerca da *engine*.

6. RECURSOS DE DIDÁTICOS

- Notebook;
- Data Show;
- Slides;
- Computadores

REFERÊNCIAS

Godot Game Engine. [Online] [Acesso em: 20 de abr de 2017]. Disponível em: <<https://godotengine.org/>>

APÊNDICE B – PLANO DE ENSINO OFICINA DA II

PLANO DE ENSINO

1. IDENTIFICAÇÃO

Cidade:	Palmas – TO
Tema:	Godot Engine: Um motor de jogos 2D e 3D livre e <i>Open Source</i>
Local:	Instituto Federal de Educação, Ciências e Tecnologia do Tocantins – Campus Palmas.

1.2. Curso

Curso	Oficina com licenciandos em computação e alunos do curso técnico em informática.
--------------	--

1.3. Data

Dia e Horário	24/11/2017 – 13h30min
----------------------	-----------------------

1.4. Ministrante

Ministrante	Elisama Martins Gonçalves
--------------------	---------------------------

2. COMPETÊNCIAS

- Conhecer a Godot Game Engine;
- Apresentar a possibilidade de criação de apps educacionais e a inserção do mesmo em sala de aula;
- Criar um exemplo prático de um app através da ferramenta Godot.

3. HABILIDADES

- Identificar os principais recursos da engine;
- Apresentar a importância da ferramenta para a educação.

4. CONTEÚDO PROGRAMÁTICO (E/OU BASE TECNOLÓGICA):

1^a Parte:

- Conceituar o que é a Godot Game Engine;
- Surgimento da ferramenta;
- Características principais;

2^a Parte:

- Oficina prática para apresentação de recursos e criação de um exemplo de jogo.

5. METODOLOGIA

A oficina será ministrada de forma expositiva e interativa por meio de apresentação de slides e discussão com os participantes.

A primeira etapa será voltada para conceituação e explicação sobre a ferramenta.

A segunda etapa será voltada para desenvolvimento de exemplo prático e esclarecimento de dúvidas acerca da *engine*.

6. RECURSOS DE DIDÁTICOS

- Notebook;
- Data Show;
- Slides;
- Computadores

REFERÊNCIAS

Godot Game Engine. [Online] [Acesso em: 18 de nov de 2017]. Disponível em: <<https://godotengine.org/>>

GitHub. [Online] [Acesso em 18 de nov de 2017]. Disponível em: <<https://github.com/godotengine/godot>>

APÊNDICE C – INSTRUMENTO DE COLETA DE DADOS



MINISTÉRIO DA EDUCAÇÃO
SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO TOCANTINS
CAMPUS PORTO NACIONAL
CURSO SUPERIOR DE LICENCIATURA EM COMPUTAÇÃO

TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO

1 – Dados de identificação:
TEMA DO TRABALHO: GODOT Game Engine - Um motor de jogo livre como ferramenta para criação de apps educacionais
Pesquisador (a): Elisama Martins Gonçalves
Instituição a que pertence: Instituto Federal de Educação, Ciência e Tecnologia do Tocantins – Campus de Porto Nacional.
Telefone para contato: (63) 98438-7377
Orientador do trabalho: Professor Me. Janio Carlos Nascimento Silva
Instituição a que pertence: IFTO Campus de Porto Nacional

Vossa Senhoria está sendo convidado (a) a participar do Projeto de Pesquisa: **GODOT Game Engine - Um motor de jogo livre como ferramenta para criação de apps educacionais**, de responsabilidade da pesquisadora: Elisama Martins Gonçalves, sob orientação do Professor Mestre Janio Carlos Nascimento Silva, do IFTO Campus de Porto Nacional.

O presente trabalho tem como objetivos:

- Construir conhecimento acerca da ferramenta GODOT Game Engine que possibilite uma metodologia ágil de criação de apps educacionais.
- Estudar e compreender o ambiente e as características do motor de jogo para utilização como ferramenta base para criação de apps educacionais.
- Avaliar a abordagem proposta (Processo e Ambiente) em um contexto de desenvolvimento de apps educacionais.

Garantir-se-á total sigilo da sua identidade, tendo em vista que sua participação neste trabalho é voluntária e que este consentimento poderá ser retirado a qualquer tempo, sem prejuízo ao trabalho.

Eu, _____ RG

Nº_____ declaro ter sido informado (a) e concordo em participar, como voluntário (a), do projeto de pesquisa acima descrito.

Porto Nacional/TO, _____ de novembro d de 2017.

Voluntário



Este instrumento de pesquisa corresponde a uma das fases do trabalho de conclusão de curso intitulado: “Godot Game Engine - Um motor de jogo livre como ferramenta para criação de *apps* educacionais”, após a participação na oficina de introdução a ferramenta Godot, gostaria de sua ajuda no sentido de oferecer impressões acerca da temática apresentada.

1. Você conhece alguma ferramenta que considere eficiente para o desenvolvimento de *apps* educacionais?

- Sim
- Não
- Não sei opinar

Se sim qual (ais)?

2. Você já tinha algum conhecimento sobre a ferramenta Godot?

- Sim
- Não

Se sim, onde você o adquiriu?

- Internet
- Professor
- Colega
-

Outros.

3. Você acredita que é possível fomentar a produção de apps educacionais com a utilização da Godot?

- () Sim
() Não
() Não sei opinar

4. Você acredita que esse tipo de ferramenta fornece alguma vantagem dentro da sua área de formação?

- () Sim
() Não
() Não sei opinar

Se sim, quais?

5. Quais foram as maiores dificuldades durante o desenvolvimento de apps com utilização da Godot?

6. Quais os maiores benefícios percebidos após uma primeira impressão do Godot?

7. Quais as maiores desvantagens percebidas após a primeira impressão do Godot?
