

## ***Frameworks* para Desenvolvimento de Jogos Educacionais: uma revisão e comparação de pesquisas recentes**

**Rháleff Nascimento Rodrigues de Oliveira, Rodrigo Pennella Cardoso, Juliana  
Cristina Braga, Rafaela Vilela da Rocha Campos**

Pós-Graduação em Ciência da Computação - Universidade Federal do ABC (UFABC)

CEP: 09210-580 – Santo André – SP – Brasil

{rhaleff.nascimento, juliana.braga}@ufabc.edu.br,

{rodrigopennella, rafaela.vilela}@gmail.com

**Abstract.** *Frameworks have been used in game development to reduce time and cost. However, there are not many frameworks for the development of educational games. This article compares the main frameworks for the educational games development, published in the last years. The result is the comparison of 13 frameworks in relation to the characteristics of application, life cycle, pedagogical and gameplay elements.*

**Resumo.** *Frameworks têm sido usados no desenvolvimento de jogos para reduzir tempo e custo. Entretanto, não há muitos frameworks para o desenvolvimento de jogos educacionais. Neste artigo são revisados e comparados os principais frameworks para desenvolvimento de jogos educacionais, publicados nos últimos anos. O resultado é a comparação de 13 frameworks em relação à aplicação, ensino, ciclo de vida, elementos pedagógicos e de jogabilidade.*

### **1. Introdução**

O uso de *frameworks* para o desenvolvimento de jogos reduz o tempo e recursos financeiros, por eles oferecerem uma representação com alto nível de abstração que modela os fatos do mundo real, por meio de um esquema conceitual, além de também poder maximizar a reutilização de códigos já testados por outros desenvolvedores [Lima et al., 2014; Rocha et al., 2001].

Segundo Rocha et al. (2001), um *framework* pode ser considerado conceitual e/ou de implementação. O “*framework* conceitual” é um esquema abstrato utilizado em um domínio específico, uma representação de alto nível que modela os fatos do mundo real, suas propriedades e seus relacionamentos. Um “*framework* de implementação” consiste em um conjunto de sistemas, que contém funcionalidades prontas para serem utilizadas, sem que os programadores as tenham de reimplementar para cada aplicação. O termo *framework* adotado neste trabalho define uma estrutura conceitual básica que pode incluir desde elementos pedagógicos até artefatos e geração de código, ou seja, considera ambas abordagens. Diante dos benefícios do uso de um *framework*, torna-se necessário identificar, dentre os existentes, quais poderiam apoiar o desenvolvimento de jogos educacionais [Lima et al, 2014], levando em considerações suas peculiaridades, principalmente aquelas relacionadas aos elementos pedagógicos e de jogabilidade.

Nesse contexto, este artigo visa comparar *frameworks* para o desenvolvimento de *games* educacionais, considerando as características de aplicação, ensino, ciclo de vida e elementos pedagógicos e de jogabilidade. Dessa forma, a principal contribuição é

apresentar informações que possam direcionar a equipe de desenvolvimento a escolher o *framework* mais adequado para cada propósito.

Este artigo está organizado da seguinte maneira: na seção 2 é apresentado o referencial teórico e trabalhos relacionados. Na seção 3 é descrita a metodologia utilizada. A análise comparativa dos *frameworks* encontrados na literatura é descrita na seção 4. Na seção 5, têm-se as discussões, seguido das considerações finais na seção 6.

## 2. Referencial Teórico

Os *frameworks* podem servir como arcabouço para informar aos *designers* e desenvolvedores de jogos educacionais sobre quais características e elementos devem ser levados em consideração na construção de um jogo [Malliarakis, 2014]. Existem *frameworks* que são aplicados para criação de jogos genéricos ou específicos, como para *games* de programação. Há também os que variam quanto ao público alvo, como ensino básico, ensino fundamental, ensino médio ou ensino superior; ou voltados para o ensino em geral. Alguns *frameworks* têm seu ciclo de vida focado no *design*, em quanto outros apresentam todo o processo de criação do jogo. Existem *frameworks* no qual há envolvimento do aluno no processo de desenvolvimento do jogo e outros o aluno é apenas usuário. Dentro dos elementos pedagógicos, os *frameworks* podem definir estruturas para a presença de objetivos de aprendizagem, como por exemplo, “90% dos alunos arrastarão e soltarão corretamente as linhas de código referentes a declaração ‘sentão’”. Existem *frameworks* que apresentam teoria de aprendizagem, como construtivismo, humanismos, etc. O perfil do aluno, como idade e contexto social, também pode ser levado em consideração. Alguns dão suporte para a definição de contexto, das atividades desenvolvidas no jogo, e da avaliação de aprendizagem. Alguns *frameworks* têm seus elementos de jogabilidade bem definido, outros não.

Trabalhos como Ahmad et al. (2014 e 2015) e Malliarakis (2014) comparam *frameworks* e modelos de *game design* para o desenvolvimento de jogos educacionais. Eles comparam elementos comuns, tais como, evolução do jogo, gamificação, evidência pedagógicas, controle, engajamento, *feedback* contínuo, contexto narrativo, teorias de aprendizagem, conflito e interação. Ahmad et al. (2015) descreve que o *EGM framework* tem componentes consistentes da categoria de projeto de desenvolvimento de um jogo educacional. Malliarakis (2014), após análise, desenvolveu um novo *framework*, nomeado *CMX design framework*, para criação de jogos para o curso de programação. Este artigo analisa elementos diferentes e complementares aos trabalhos relacionados, conforme critérios apresentados na seção 4.

## 3. Metodologia

O objetivo desse artigo é revisar e comparar as pesquisas recentes sobre *frameworks* para desenvolvimento de *games* educacionais, de modo a responder as questões: (Q1) “Quais são os principais *frameworks* de desenvolvimento de *games* educacionais?” e (Q2) “Quais são as principais semelhanças e diferenças desses *frameworks* (em relação à aplicação, ensino, ciclo de vida, elementos pedagógicos e de jogabilidade)?”.

Uma revisão da literatura foi conduzida para obter uma compreensão do estado da arte e dos conceitos essenciais dos *frameworks*. Foram pesquisados artigos nos principais motores de busca nacionais e internacionais, congressos e periódicos: ACM DL, IEEE Explore, SBIE, CBIE, SBGames, RBIE, RENOTE e Google Acadêmico. As

chaves de busca utilizadas foram: (1) português: (*framework*) AND (*jogos OR games*) AND (*educacionais OR sérios OR serious*); e (2) inglês: (*framework*) AND (*games*) AND (*educational OR serious*).

Os textos encontrados foram analisados usando quatro critérios de inclusão: (i) publicado em inglês ou português; (ii) publicado entre 2008 e 2018; (iii) contém as palavras chaves descritas; e (iv) apresenta *frameworks* para o desenvolvimento de jogos educacionais ou faz uma análise comparativa deles. Os três critérios de exclusão foram: (i) é resumo, palestra, curso, tutorial ou afim; (ii) não está disponível livremente para consulta ou *download*; ou (iii) não é aplicado a jogo educacional.

As pesquisas retornaram 28 publicações que foram seguidas de uma leitura de seus resumos. Foram incluídos 10 artigos (seis em inglês e quatro em português), oito que descrevem o desenvolvimento de *frameworks* [Alves, 2011 e 2017; Ibrahim e Jaafar, 2009; Lima et al., 2014, Malliarakis, 2014; Medeiros et al., 2014; Moser, 2013; Westera et al., 2008] e dois que fazem uma análise comparativa de *frameworks* e modelos de *game design* [Ahmad et al., 2014 e 2015]. Da leitura desses artigos, cinco *frameworks* foram incluídos nos resultados desta pesquisa [Freitas e Jarvis, 2006; Kiili, 2005; Winn, 2007; Yue, Zin e Jaafar, 2009; Yusoff et al., 2009], totalizando 13 artigos revisados e comparados. Apesar dos *frameworks* de Killi (2005), Freitas e Jarvis (2006) e Winn (2007) não estarem dentro do critério de inclusão do período de publicação, eles foram incluídos na análise por serem relevantes e referenciados nos outros trabalhos.

Os critérios de análise e comparação dos *frameworks* foram definidos baseado na leitura dos trabalhos relacionados e artigos revisados. Após a leitura, observou-se uma variedade quanto a aplicação e nível de ensino dos *frameworks*. Foram selecionados os elementos mais citados nos artigos revisados para os elementos pedagógicos e de jogabilidade.

#### 4. Resultados

Esta seção apresenta uma análise dos *frameworks* e os resultados das duas questões de pesquisa propostas. Eles são descritos a seguir, ordenados por ano de publicação. Cada um deles é identificado pela letra F seguindo de um número. Essa identificação será utilizada para apresentar os resultados comparativos de cada um.

**F1:** *Experimental Gaming Model Framework (EGMF)*, proposto por Kiili (2005), pode ser usado para projetar e analisar jogos educativos. Contudo, ele funciona apenas como um elo entre a teoria educacional e o *game design* e não fornece os meios para todo o projeto de *design* de jogos. O processo de aprendizado é realizado por um *loop* experiencial, no qual ele é adaptado ao resultado da solução do primeiro problema e, em seguida, fornece um desafio e continua o *loop* para experimentar problemas e resolver todo *loop*.

**F2:** *Framework Four-Dimensional (FDF)*, proposto por Freitas e Jarvis (2006), é composto por quatro princípios básicos: (i) Contexto: define características de tipo do jogo, infraestrutura requerida, especificações técnicas, local de uso, etc.; (ii) Pedagógico: contexto de aprendizagem; (iii) Aprendiz: suas características dentro do jogo, como idade, preferências, disponibilidade e nível de conhecimento prévio; e (iv) Representação: que deve ser retratada no jogo, como por exemplo, avatares que terão características específicas baseadas no contexto do jogo.

**F3:** *Design, Play and Experience (DPE) framework*, desenvolvido por Winn (2007), visa descrever a relação entre o *designer* e o jogador, além de especificar os conceitos que correspondem a cada camada projetada dependendo da fase e do tipo de usuário. As camadas são quatro: aprendizagem, narrativa, jogabilidade e experiência, sendo que há três fases: *Design, Play e Experience*. O *designer* identifica os objetivos de aprendizagem que guiarão as atividades (métricas de avaliação final). Todas as outras camadas são projetadas com base nesses objetivos e depois configuradas e refinadas a partir do *feedback* do envolvimento com o jogo (*Experience*). A fase *Play* representa a interação dos jogadores com os recursos do jogo.

**F4:** Westera et al. (2008) apresenta um *framework* para *design* de jogos sérios baseados em cenários, que visa reduzir a complexidade do *design* nos níveis conceitual, técnico e prático. No nível conceitual, identifica os elementos básicos que compõem a configuração do jogo estático e descreve a dinâmica do jogo (mudanças de estado dos vários componentes ao longo do tempo). No nível técnico, apresenta uma arquitetura básica do sistema, que compreende várias ferramentas de construção. No nível prático, um conjunto de princípios de *design* (estrutura do jogo, *feedback* e representação do jogo) é apresentado para controlar e reduzir a complexidade do *game design*.

**F5:** *Digital Game Based Learning (DGBL)*, proposto por Yue, Zin e Jaafar (2009), é um *framework* para educação de história que inclui o currículo do país, os valores morais e psicológicos, e as teorias de aprendizagem e elementos educacionais. Ele utiliza o jogo como meio para introduzir os conteúdos de aprendizagem.

**F6:** *Educational Game Design Model (EGDM)*, proposto por Ibrahim e Jaafar (2009), é um *framework* que combina elementos de aprendizagem (pedagogia e conteúdo de aprendizagem) com elementos de desenvolvimento (*game design*). Os autores enfatizam a usabilidade que compreende eficiência, competência e satisfação, tanto no *game design* quanto no conteúdo de aprendizagem.

**F7:** *Conceptual Framework*, proposto por Yusoff et al. (2009), fornece um guia de referências para o *design* de jogos educativos. O autor afirma que os resultados de aprendizagem (*intended learning outcomes*) representam quais objetivos os alunos devem ser capazes de alcançar depois de concluir com êxito todas as tarefas atribuídas. Um resultado de aprendizagem está intimamente conectado com as habilidades que os alunos devem desenvolver por meio de sua interação com o jogo dentro do processo de aprendizagem (*capability*) e ligado também ao fato do jogo estar em conformidade com o material educacional que os alunos têm que aprender (*instructional content*).

**F8:** *InVision Framework*, proposto por Alves (2011), contém um conjunto de bibliotecas com componentes funcionais de jogos e é extensível por meio de *scripts*. É dividido em quatro partes: (i) camada de suporte nativo, (ii) subcamada que transforma chamadas de funções em objetos e ações, (iii) conjunto de bibliotecas e funcionalidades, e (iv) aplicação para construção de protótipos.

**F9:** *Child-Centered Game development Framework (CCDG)*, proposto por Moser (2013), visa fornecer aos pesquisadores, desenvolvedores e *designers* uma coleção de abordagens para o envolvimento ativo de crianças de 10 a 14 anos durante as fases de desenvolvimento (análise, conceito, *design*, implementação e avaliação). É baseada na ideia de que é importante a parceria com os usuários para entender o que é necessário para desenvolver jogos para eles. As crianças são incentivadas a assumir o papel de usuários, testadores, informantes, etc.

**F10:** Medeiros et al. (2014) apresenta um *framework* para criação de jogos voltados para o ensino de lógica de programação, apresentando a concepção e o desenvolvimento por meio da integração do ambiente visual de programação *blockly* (biblioteca *JavaScript* desenvolvida pela ©Google) com cenários criados em HTML5. Os jogos criados por meio desse *framework* possuem uma estrutura com três componentes: *game canvas*, *blockly editor* e *game mode selector*.

**F11:** *BasisJED*, criado por Lima et al. (2014), visa auxiliar o desenvolvimento de novos jogos sérios, buscando manter o equilíbrio entre aprendizado e entretenimento. Os autores levaram em consideração características essenciais de jogos sérios, como universo do jogo, fases, desafios, desempenho do jogador, entre outros.

**F12:** *CMX Design framework*, proposto por Malliarakis (2014), inclui conceitos de representação para jogos para ensino de programação. Ele é abstrato e detalhado para ser um guia para *designers* e desenvolvedores. Há seis principais conceitos: objetivos de aprendizagem, pedagogia, resultados de aprendizagem, usuário, cenário e atividades.

**F13:** *Framework* “Eu fiz Meu Game”, proposto por Alves (2017), visa o desenvolvimento de jogos digitais com e por crianças. Este *framework* requer a participação ativa dos envolvidos (educadores, acadêmicos e crianças). O processo é composto por quatro etapas: (i) Envolvimento, (ii) Experiência, (iii) Transposição e (iv) Criação de Jogos Digitais. A mediação simbólica é compreendida como condição necessária em cada etapa e atividade a ser realizada, por isso apresenta-se de maneira concomitante a todo processo (fundamentação para o desenvolvimento cognitivo).

O Quadro 1 apresenta uma visão geral da comparação entre os *frameworks* selecionados, usando cinco critérios descritos a seguir.

(i) *aplicação*: refere-se ao uso geral do *framework* ou em disciplinas e conteúdo específicos. Neste elemento, é possível verificar que apenas os *frameworks* **F5**, **F8**, **F10** e **F12** são de uso específico, o primeiro em história e os outros relacionados à programação. Os demais *frameworks* são de uso geral.

(ii) *níveis de ensino*: refere-se ao nível de ensino do público alvo do jogo. Sendo assim, os *frameworks* **F19** e **F13** são focadas no ensino infantil, o **F10** e **F12** para ensino médio e o **F4**, **F6** e **F8** no ensino superior; os demais não especificam o nível de ensino, podendo ser usados como estruturas em todos os níveis.

(iii) *ciclo de vida*: refere-se às fases de desenvolvimento de um jogo educativo, são elas: *design*, implementação e teste e avaliação. Apenas o *framework* **F8** não dá suporte à fase de *design*; sendo que **F9** e **F13** também dão suporte para o envolvimento do usuário nessa fase (*design* participativo). Os *frameworks* **F5**, **F9** e **F13** focam em todo processo de criação; e **F2**, **F3**, **F8** também abordam a fase de teste e avaliação.

(iv) *pedagógicos*: refere-se aos elementos pedagógicos de objetivos de aprendizagem, avaliação de aprendizagem, perfil do aluno, teoria de aprendizagem e definição do contexto e atividade. Os *frameworks* **F6**, **F7**, **F8**, **F10** não consideram a elaboração dos objetivos de aprendizagem. Apenas os *frameworks* **F1**, **F4** e **F6** apresentam elementos de avaliação da aprendizagem. Os *frameworks* que levam em consideração o perfil do aluno são o **F2**, **F3**, **F9**, **F12** e **F13**. Apenas os *frameworks* **F4**, **F8** e **F11** não abordam nenhuma teoria de aprendizagem, e **F6**, **F9** e **F11** não dão suporte para definição de contexto e atividades desenvolvidas no jogo.

(v) *jogabilidade*: refere-se às regras, *feedback* e imersão, que são elementos importantes para a jogabilidade (interações significantes que o jogador tem com o jogo). Os *frameworks* **F1**, **F6**, **F7**, **F8**, **F10** e **F12** não apresentam o elemento regra. Apenas o **F6** não apresenta o *feedback* em sua estrutura e os **F4** e **F7** não apresentam o elemento imersão. Pode-se concluir que os *frameworks* **F6** e **F7** não define bem a jogabilidade.

Quadro 1: Análise comparativa dos frameworks para jogos educacionais

ID	Autor (ano)	Elementos Nome	Aplicação		Nível de Ensino		Ciclo de vida			Pedagógicos				Jogabilidade			
			Genérica	Específica	Genérico	Específico	Design	Implementação	Teste e Avaliação	Objetivos de Aprendizagem	Avaliação da Aprendizagem	Perfil do Aluno	Teorias de Aprendizagem	Definição do Contexto e Atividade	Feedback	Regras	Imersão
F1	Kiili (2005)	EGMF	X		X		X			X	X		X	X	X	X	X
F2	Freitas e Jarvis (2006)	FDG	X		X		X		X	X		X	X	X	X	X	X
F3	Winn (2007)	DPE	X		X		X		X	X		X	X	X	X	X	X
F4	Westera et al. (2008)	Framework for reducing design complexity	X			ES	X			X	X			X	X	X	
F5	Yue, Zin e Jaafar (2009)	DGBL		H		EF2	X	X	X	X			X	X	X	X	X
F6	Ibrahim e Jaafar (2009)	EGDM	X			ES	X				X		X				X
F7	Yusoff et al. (2009)	Conceptual Framework	X		X		X						X	X	X		X
F8	Alves (2011)	InVision		P		ES		X	X					X	X		
F9	Moser (2013)	CCGD	X			EI	X*	X	X	X			X	X	X	X	X
F10	Medeiros et al. (2014)	Framework para jogos de LP		LP		EM	X	X					X	X	X	X	X
F11	Lima et al. (2014)	BasisJED	X		X		X			X				X	X	X	X
F12	Malliarakis (2014)	CMX Design Framework		P		EM	X			X		X	X		X		X
F13	Alves (2017)	Eu fiz meu game	X			EI	X*	X	X	X			X	X	X	X	X

**Legenda:** \*Design Participativo, EI: Ensino Infantil, EF2: Ensino Fundamental 2, EM: Ensino Médio, ES: Ensino Superior, H: História, LP: Lógica de Programação, P: Programação.

## 5. Discussões

Para análise e comparação dos *frameworks*, de forma geral e abrangente, eles foram organizados em seis grupos, conforme objetivos (aplicação e ensino) e etapas de desenvolvimento (ciclo de vida), como é apresentado no Quadro 2.

Quadro 2: Visão geral dos grupos de frameworks para jogos educacionais

Objetivos	Aplicação	Genérico		G.	Específico
	Ensino	Genérico (i)	Específico (ii)		Específico (iii)
Ciclo de Vida (grupo)	Design (iv)	<b>F1, F2, F3, F7, F11</b>	<b>F4 (ES), F6 (ES), F9 (EI), F13 (EI)</b>		<b>F5 (H, EF2), F10 (LP - EM), F12 (P, EM)</b>
	Implementação (v)		<b>F9 (EI), F13 (EI)</b>		<b>F5 (H, EF2) F10 (LP, EM), F8 (P, ES)</b>
	Teste e Avaliação (vi)	<b>F2, F3</b>	<b>F9 (EI), F13 (EI)</b>		<b>F5 (H, EF2), F8 (P, ES)</b>

**Legenda:** EI: Ensino Infantil, EF2: Ensino Fundamental 2, EM: Ensino Médio, ES: Ensino Superior, G: Genérico, H: História, LP: Lógica de Programação, P: Programação

Os seis grupos são: (i) *aplicações e níveis de ensino genéricos*: foram analisados os *frameworks* **F1**, **F2**, **F3**, **F7** e **F11**; (ii) *aplicações genéricas e ensino específico*: foram analisados os *frameworks* **F9** e **F13** para o ensino infantil (EI), os **F4** e **F6** para o ensino superior (ES); (iii) *aplicação específica*: foram analisados somente *frameworks* para o ensino de programação, **F8**, **F10** e **F12**, pois há apenas um específico para História, **F5**; (iv) *design* (focado no *game design* e *design* pedagógico): foram analisados os *frameworks* **F1**, **F2**, **F3**, **F7** e **F11** que são genéricos em aplicação; além de comparados o **F9** e **F13**, que apresentam *design* participativo, em conjunto com o **F4**

e F6 que também são para ensino específico; (v) *implementação*: foram analisados os *frameworks* **F5, F8, F9, F10 e F13**, que apresentam aspectos de implementação em seu ciclo de vida; e (vi) *teste e avaliação*: têm-se os *frameworks* **F2, F3, F5, F8, F9 e F13**.

No primeiro grupo (i), o **F1** tem sido usado para garantir a experiência de fluxo no *game design* (independente de aplicação), pois enfatiza a importância de considerar vários antecedentes e o fluxo no *game design*: senso de controle, jogabilidade, *feedback* sem ambiguidade, etc. O **F2** foi usado para criar um jogo sério para treinamento de controle de infecção. O **F3** fornece uma estrutura para analisar um jogo e também uma linguagem uniforme na qual um grupo de indivíduos pode discutir e criticar o *design* de um jogo. Desta forma, o **F3** tem sido usado na análise de jogos sérios existentes, pois fornece uma estrutura para decompor os elementos do *design* (aprendizado, narrativa, jogabilidade, experiência do usuário e tecnologia). O **F7** foi usado para criar um jogo sério que visa permitir que alunos estrangeiros reconheçam a localização do destino final (no sistema de ônibus universitário); incluindo teoria de aprendizagem e pedagogia como requisitos. O **F11**, no ano de publicação (2014), estava em fase de implementação e não foi aplicado em nenhum contexto.

No segundo grupo (ii), *frameworks* com aplicação genérica e específica para o ensino superior, a aplicação do **F4** não foi relatada e, segundo o autor, será avaliada em trabalhos futuros. O **F6** não relata o seu uso no desenvolvimento de jogos. Quanto para ensino infantil, o **F9** foi aplicado para o desenvolvimento de vários minijogos por crianças, uma delas participou de todo o processo do “*Crazy Action Fly*”, por exemplo. O **F13** foi aplicado na construção do jogo digital “Os brinquedos que criaram vida”, desenvolvido por quatro crianças de 9 anos, no qual duas tinham deficiência intelectual.

No terceiro grupo (iii), o **F8** foi criado para ser usado no aprendizado de algoritmo, programação, inteligência artificial, redes e grafos, permitindo o uso de *script* em várias linguagens, para o ensino superior. O **F10** utiliza o *Blockly* para criação de jogos voltados para o ensino de lógica de programação (independente do conteúdo) para ensino médio. No **F12**, os elementos de conteúdos têm como objetivo familiarizar alunos do ensino médio, que são iniciantes em programação, com conceitos como variáveis, declaração *if*, *loops* e lógica algorítmica.

No quarto grupo (iv), em relação aos elementos de jogabilidade e *game design*, o **F1** enfatiza a importância de considerar vários antecedentes de fluxo no *design* de jogos educacionais: desafios combinados com o nível de habilidade de um jogador, objetivos claros, *feedback* sem ambiguidade, senso de controle, jogabilidade, gamificação, atenção focada e enredo. O **F2** destaca quatro elementos principais de um jogo sério e seu uso efetivo (conforme seção 4.1), sendo que apenas o elemento de *Representação* está relacionado ao *game design* e inclui níveis de imersão, fidelidade e interatividade. Entretanto, em se tratando da área de simulação, alguns elementos que diminuem a fidelidade podem aumentar o engajamento (como por exemplo, pausar e voltar uma simulação, perde a fidelidade de comportamento, mas pode proporcionar uma visualização detalhada e maior interatividade). O **F3** descreve a relação entre o *designer* e o jogador, são incluídas camadas de aprendizagem, narração, jogabilidade e experiência do usuário; com inserção de elementos de *game design*. Por exemplo, na camada de narrativa é projetada a narração, cenário, personagens e o elemento da imersão, na camada de jogabilidade é definida a mecânica, dinâmica e efeitos, e na camada de experiência tem-se a preocupação com a diversão. O **F4** foca em princípios

de *feedback* e representação dos componentes (personagens, gráficos, animações e sons). O **F5**, específico para o ensino de História, traz elementos de *game design* mais definidos, como enredo, regras, imersão, satisfação, *feedback*, desafios e competição, recompensa e prêmio, e tecnologia multimídia. O **F6** enfatiza a usabilidade (satisfação e eficiência), multimodalidade (multimídia e interação) e diversão (objetivos claros, resultados incertos, desafios e fantasia). O **F7** apresenta algumas diretrizes para o *game design*, por exemplo, na etapa de “Atributos do Jogo”, a interação e *feedback* estão presentes, e na etapa de “Mecânica do Jogo”, a mecânica e as regras são definidas. O **F9** apresenta elementos de narração, cenário, personagens, mecânica e imersão. Os *frameworks* **F9** e **F13** usam *design* participativo. Eles se diferenciam em relação ao público-alvo, uma vez que o **F9** se aplica a pré-adolescentes (crianças de 10 a 14 anos), e o **F13** envolve crianças em fase de alfabetização (EI) e aborda a inclusão de crianças com deficiência intelectual. O **F10**, focado em lógica de programação, não apresenta elementos de *game design* bem definidos, mas é possível identificar a diversão, a imersão e o *feedback* como principais elementos. O **F11** leva em consideração várias características do *design*, como imersão, universo do jogo, fases, desafios e recompensas. Os conceitos mais proeminentes que definem o *game design* do **F12** são: cenário, personagens, imersão, objetivos claros e premiação.

Ainda no quarto grupo (iv), em relação aos elementos pedagógicos e *design* instrucional, o **F1** vincula a jogabilidade à aprendizagem experiencial para facilitar a experiência de fluxo, descrevendo a aprendizagem como um processo cíclico por meio da experiência direta no mundo do jogo. Ele ainda enfatiza que a atividade necessária para a aprendizagem não é apenas cognitiva, mas também comportamental. O **F2** apoia-se por perspectivas associativas (instrutivista e centrada em tarefas), cognitivas (construtivismo) e situacionais (aprendizagem em comunidades de práticas). O **F3** propõe o uso da Taxonomia de Bloom para definir os objetivos de aprendizado. O **F4** enfatiza a importância do *feedback* adequado, que foca no progresso geral do aluno e seu desempenho. Propõe também um *feedback* em pares, para estabelecer a reflexão como uma atividade distinta, e assim, melhorar a qualidade da aprendizagem. O **F5** apresenta elementos de componente pedagógico que devem ser levados em consideração, tais como, a definição de metas de aprendizagem, definição da teoria de aprendizagem, psicologia educacional, necessidades curriculares do país, patriotismo e valor moral e a teoria da memorização e esquecimento. Assim como o **F3**, o **F6** propõe o uso da Taxonomia de Bloom para projetar os resultados de aprendizagem. Além disso, o **F6** foi projetado especificamente para autoaprendizagem do aluno com módulos de autoavaliação integrados e, também, é explorado a teoria da motivação e a resolução de problemas. O **F7** se baseia em taxonomias para alcançar os resultados de aprendizagem, como a de Bloom (domínio cognitivo), de Dave (domínio psicomotor) e de Krathwohl (domínio afetivo). O **F8**, por ser um *framework* de implementação, não aborda sobre *design* pedagógico, contudo apresenta exemplos de seu uso em disciplinas do curso de Ciência da Computação, como o ensino de programação, redes e comunicação via Internet, computação gráfica e inteligência artificial. O **F9** se baseia em abordagens de aprendizagem colaborativa, ativa, reflexiva e cultural. O **F10** aborda o uso da teoria de aprendizagem significativa, teoria pedagógica do construtivismo e taxonomia de Bloom. O **F11** sugere definir os objetivos de aprendizagem na fase de concepção do jogo, podendo ser objetivos genéricos ou específicos de programação e, também, os alunos podem definir seus próprios objetivos. O **F12** sugere modelar o jogo com elementos pedagógicos, como por exemplo, um ambiente de sala de aula, desempenho do jogador,

instrução, um assistente que represente um professor e a liberdade para fracassar. Por fim, o **F13** muito se assemelha ao **F9**, os dois propõem uma estratégia pedagógica centrado no aluno. No entanto, o **F13** enfatiza os princípios propostos por Vigotski em relação a mediação simbólica e a atividade criadora e imaginativa.

No quinto e sexto grupos (v. implementação e vi. teste e avaliação), o **F5** apresenta um modelo de *design* instrucional no qual sugere que, para essas fases, o protótipo deve ser desenvolvido e testado completamente, por meio de teste de reprodução, usabilidade, teste *alpha* e *beta*. O *alpha* garantirá que o jogo é jogável do começo até o final. O *beta* testará erros como a interatividade do jogo. O autor sugere usar os *softwares* Studio Max ou Maya para modelar personagens em 3D. O **F8** criou uma aplicação sobre uma arquitetura de *software* extensível através de *scripts* e bibliotecas dinâmicas. Ele foi escrito em C#, com algumas bibliotecas de componentes em C++, contudo, há suporte para executar o *script* em F#, Boo, Visual Basic, Python e Ruby, dando suporte para estudantes criarem seus *games* com *scripts* de forma simples. Além disso, o designer pode usar tais *scripts* para prototipar e testar os jogos, através de teste de profundidade e teste *alpha*. No **F9**, o usuário participa de todo processo de criação do jogo, desde a concepção até o *playtest*. Com isso, a implementação do que foi desenhado pela criança pode ser realizada por meio de ferramentas como Scratch, Kodu ou Zieege. A avaliação consiste em avaliar os protótipos, usando a avaliação heurística, *Structured Expert Evaluation Method (SEEM)* e *Heuristics to Evaluate the Playability (HEP)*. O **F10** propõe que os jogos precisam ser implementados em HTML5 e integrado ao editor *Blockly*. O programador deve criar arquivos em *Javascript* contendo os blocos de comando disponíveis no *Blockly*. No **F13** propõe que a equipe técnica deve avaliar qual a tecnologia adequada para implementar o jogo criado pelas crianças, pois nessa fase, o papel delas é de validar. A fase de avaliação consiste em realizar testes de jogabilidade. Os **F8** e **F13** têm artefatos específicos. O **F8**, por exemplo, produz uma API simplificada para áudio e renderização, que prevê abstrações para a reprodução de áudio, usando a biblioteca nativa *FMod*. O **F13** possui artefatos, como fichas de especificação e avaliação, que devem ser aplicados no processo de desenvolvimento do jogo.

Ainda no sexto grupo (vi. teste e avaliação), o **F2** propõe o uso de teste formativo, avaliação e teste interativo de demonstradores com usuários, em cada versão piloto. O **F3** utiliza o *playtest* para refinar o *design* do jogo e o processo de *design* de interação para desenvolver os jogos, criar protótipo, testar o protótipo e modificar o jogo com base nos resultados de teste. O autor sugere que os componentes do jogo sejam testados inicialmente em um protótipo analógico, como um jogo de tabuleiro.

## 6. Considerações Finais

A utilização de *frameworks* no desenvolvimento de *software* permite otimizar recursos (financeiros, humanos, tecnológicos e tempo). Especificamente para o desenvolvimento de jogos educacionais existem diversas opções de *frameworks*, cada um com suas particularidades e objetivos. Neste artigo foram analisados os 13 principais *frameworks* encontrados na literatura, levando em consideração critérios de: aplicação, nível de ensino, ciclo de vida de desenvolvimento, elementos pedagógicos e de jogabilidade. Foram respondidas as questões de pesquisa que apresentaram as semelhanças e diferenças dos *frameworks*, além das discussões que também apresentam vantagens e desvantagens em relação ao contexto de uso. Pôde-se notar que a maioria dos

*frameworks* foca mais na fase de *design*, tendo oportunidade de desenvolvimento de *frameworks* que apoiem todo o ciclo de vida, desde a concepção até a avaliação dos jogos educacionais; principalmente focados na implementação. Assim, é possível observar que há uma preocupação para que haja uma interação significativa do jogador com o jogo, por meio dos elementos analisados sobre jogabilidade. Entretanto, notou-se que a maioria dos *frameworks* não abordam a avaliação da aprendizagem, sendo uma oportunidade de exploração.

De forma geral, não existe um melhor *framework* para o desenvolvimento de jogos educacionais, mas aquele ou aqueles que melhor se adequam à necessidade do usuário. Nesse sentido, este trabalho pode ajudar o *game designer* e o *designer* instrucional a escolherem um melhor *framework* para o desenvolvimento do seu jogo, a partir dos seus requisitos e dos diferentes elementos desses *frameworks*.

## Referências

- Ahmad, M.; Rahim, L.A.; Arshad, N.I. (2015). “An analysis of educational games design frameworks from software engineering perspective”. JICT, pp. 123-151.
- Ahmad, M.; Rahim, L.A.; Arshad, N.A. (2014). “A review of educational games design frameworks: An analysis from software engineering”. In: ICCOINS, pp. 1-16.
- Alves, A.G. (2017). “Eu Fiz Meu Game: Um framework para criação de jogos digitais por crianças”. In: VI CBIE, pp. 2-11.
- Alves, D.F.M. (2011). “Projeto InVision Framework”. Dissertação de Mestrado. Instituto de Informática, Universidade Federal de Goiás. Goiânia.
- Freitas, S.; Jarvis, S. (2006). “A Framework for Developing Serious Games to meet Learner Needs”. In: I/ITSEC, pp. 1-11.
- Ibrahim, R.; Jaafar, A. (2009). “Educational games (EG) design framework: combination of game design, pedagogy and content modeling”. In: ICELTICS, pp. 293-298.
- Kiili, K. (2005). “Digital game-based learning: Towards an experiential gaming model”. Internet and Higher Education, 8, pp. 13-24.
- Lima, W.A.; Aranha, R.V.; Raimann, E.; Camargo, C.A.X.; Inocencio, A.C.G.; Ribeiro, M.W.S. (2014). “Uma Proposta de Framework para o Auxílio na Criação de Serious Games”. In: XI WRVA, pp. 126-131.
- Malliarakis, C.; Satratzemi, M.; Xinogalos, S. (2014). “Designing educational games for computer programming: A holistic framework”. EJEL, 12 (3), pp. 281-298.
- Medeiros, T.; Brasil, P.; Aranha, E. (2014). “Um framework para criação de jogos voltados para o ensino de lógica de programação”. In: XXV SBIE, pp. 1113-1117.
- Moser, C. (2013). “Child-Centered Game Development (CCGD): Developing Games with Children at School”. Personal and Ubiquitous Computing, 17, pp. 1647-1661.
- Rocha, L. V.; Edelweiss, N.; Iochpe, C. (2001). “GeoFrame-T: A Temporal Conceptual Framework for Data Modeling.” In: 9th ACM SIGSPATIAL, pp. 124-129.
- Westera W.; Nadolski R.; Hummel H.; Wopereis I. (2008). “Serious games for higher education: a framework for reducing design complexity”. Journal of Computer Assisted Learning, 24, pp. 420-432.
- Winn, B. M. (2007). “The Design, Play, and Experience Framework”. In: Ferdig, R. (ed.) Handbook of Research on Effective Electronic Gaming in Education, pp. 1010-1024. IGI Global, London.
- Yue, S.; Zin, N.; Jaafar, A. (2009). “Digital Game-based learning (DGBL) model and development methodology for teaching history”. WSEAS Transactions on Computers, 8 (2), pp. 322-333.
- Yusoff, A.; Crowder, R.; Gilbert, L.; Wills, G. (2009). “A Conceptual Framework for Serious Games”. In: 9th ICALT, pp. 21-23.