



Desenvolvimento e Avaliação de um Perfil UML para Modelagem de Jogos Educacionais Digitais

Title: *Development and Assessment of an UML Profile for Digital Educational Games Modeling*

Lucas Rodrigues de Oliveira

Curso de Ciência da Computação –
Universidade Federal de Goiás/Regional Jataí
lukimrodrigues@gmail.com

Heitor Augustus Xavier Costa

Departamento de Ciência da Computação –
Universidade Federal de Lavras (DCC/UFLA)
heitor@dcc.ufla.br

Ana Carolina G. Inocêncio

Curso de Ciência da Computação –
Universidade Federal de Goiás/Regional Jataí
anacarolina.inocencio@gmail.com

Paulo Afonso Parreira Júnior

Departamento de Ciência da Computação –
Universidade Federal de Lavras (DCC/UFLA)
paulo.a.junior@dcc.ufla.br

Resumo

UML (Unified Modeling Language) é uma linguagem de propósito geral para modelagem de software, sendo uma das mais conhecidas e utilizadas para a representação gráfica de modelos. Sendo assim, apenas os elementos pré-existentes nessa linguagem podem não ser suficientes para facilitar a identificação e o entendimento dos conceitos inerentes a domínios de software mais específicos. Um exemplo é o domínio de Jogos Educacionais Digitais (JEDs). Neste trabalho é apresentado o desenvolvimento e a avaliação do UP4EG, um perfil UML para modelagem de JEDs, por meio de diagramas de classes UML. O UP4EG reúne diversos estereótipos e “tagged values” distribuídos em seis pacotes, que representam os principais elementos que um JED deve apresentar. A partir de um estudo experimental conduzido sobre o perfil proposto com estudantes de ciência da computação, percebeu-se que a quantidade de erros e o tempo gasto para identificação dos principais elementos de um JED podem ser significativamente melhorados, quando esse perfil é usado na modelagem desse tipo de software.

Palavras-Chave: Perfil UML, Jogos Educacionais Digitais, Modelagem de Software.

Abstract

UML (Unified Modeling Language) is a general purpose language for software modeling, moreover, is the language most known and used for graphical representation of models. Just the elements preexisting in the UML (Unified Modeling Language) may not be sufficient to facilitate understanding of specific domain models. An example of specific problem domain is the Digital Educational Games (DEGs). In this paper, we describe the UP4EG, a UML profile for DEGs modeling, using UML class diagrams. The UP4EG gathers many stereotypes and tagged values divided into six packages that represent the main elements that a DEG should present. From an experimental study on the proposed profile with computer science students, it was realized that number of errors and the time spent for the identification of the main elements of a DEG can be improved when this profile is used in the modeling of this type of software.

Keywords: UML Profile, Digital Educational Games, Software Modeling.



1 Introdução

A modelagem de um software é uma das fases mais importantes do seu processo de desenvolvimento. Ela consiste na criação de modelos que visam a facilitar o entendimento do domínio do problema para o qual o software está sendo desenvolvido (Booch, 2005). Modelos de software são representações simplificadas de conceitos e objetos do mundo real, ressaltando suas principais características, de acordo com as necessidades do projeto (Sommerville, 2011). Modelos de softwares podem ser visualizados por meio de representações gráficas (Sommerville, 2011) e, de acordo com Fowler (2004), uma das linguagens mais conhecidas e utilizadas para representação gráfica de modelos de software é a UML (*Unified Modeling Language*) (OMG, 2015).

Pelo fato de a UML ser uma linguagem genérica, isto é, que não foi desenvolvida para a representação de modelos de um tipo específico de software, os elementos de modelagem tradicionais dessa linguagem podem não ser suficientes para facilitar a identificação e o entendimento dos principais conceitos inerentes a domínios de software mais específicos (Gérard, 2011; Lisboa Filho *et al.*, 2010). Por exemplo, em um diagrama de classes UML, todas as entidades de um software que são representadas como classes possuem uma mesma representação visual, *i.e.*, um retângulo com três compartimentos, sendo o primeiro para o nome da classe e os outros dois para a representação de seus atributos e métodos, respectivamente (mais detalhes sobre diagramas de classes são apresentados na Seção 2.1). No contexto do desenvolvimento de um software para o domínio de sistemas embarcados, por exemplo, caso a UML fosse utilizada para modelagem desse software, não haveria distinção visual entre uma classe que representa um atuador do sistema e uma classe que representa um sensor. Essa falta de identidade visual para entidades semanticamente diferentes pode prejudicar o entendimento do projeto (*design*) software, bem como sua manutenção. Além disso, as possíveis restrições existentes entre entidades de um domínio específico (por exemplo, não pode existir um “atuador”, sem que este esteja conectado a uma entidade de controle do sistema) não seriam tratadas de forma nativa pela UML.

Uma das alternativas propostas para contornar essa limitação é *estender* essa linguagem por meio de um conjunto de mecanismos incluídos dentro do conceito de perfis UML (OMG, 2014). Segundo Lisboa Filho *et al.* (2010), um perfil UML é um conjunto de elementos que permitem alterar a sintaxe/semântica dos elementos da UML, sem violar sua semântica original – mais detalhes sobre perfis UML são apresentados na Seção 2.1 desse artigo. Um exemplo de domínio de software específico é o de Jogos Educacionais Digitais (JEDs). Uma modelagem tradicional, apenas com os elementos básicos da UML, pode não ser suficiente para representar tal domínio de forma clara. Isso faz com que possam surgir algumas dúvidas durante a modelagem, o entendimento e a manutenção de um JED, tais como: (i) quais elementos do jogo referem-se à identidade do jogador? (ii) em quais componentes estão encapsuladas as fases do jogo? (iii) em quais classes se encontram os desafios? entre outros.

Na literatura, é possível encontrar diversos trabalhos referentes à confecção de perfis UML para diferentes domínios, como Geoprocessamento (Lisboa Filho *et al.*, 2012), Persistência de Dados (Torres *et al.*, 2015), Orientação a Aspectos (Uetanabara Júnior *et al.*, 2010; Evermann, 2007), entre outros. Entretanto, há escassez de trabalhos relacionados à modelagem conceitual de JEDs, principalmente, quanto ao desenvolvimento e à avaliação de perfis UML para esse domínio.

Neste trabalho, tem-se como hipótese que a eficácia e a eficiência na identificação dos principais elementos de um JED podem ser aprimoradas por meio da utilização de um perfil UML específico para esse domínio. Sendo assim, desenvolveu-se um perfil UML, denominado



UP₄EG (*UML Profile for Educational Games*), composto de trinta e um elementos de modelagem, confeccionados com base na descrição dos principais elementos de um JED, proposta por Anetta (2010), bem como na modelagem de trinta e quatro jogos educacionais disponíveis na literatura e em repositórios de software livre. Com o intuito de verificar a hipótese levantada anteriormente, o perfil UP₄EG foi avaliado por meio de um estudo experimental com alunos de graduação em Ciência da Computação, cujo intuito foi comparar a quantidade de erros e o tempo gasto para identificação dos principais elementos de um JED utilizando o UP₄EG e outra proposta para modelagem conceitual de JEDs, disponível na literatura. Como resultados, percebeu-se que a quantidade de erros e o tempo gasto para identificação dos principais elementos de um JED podem ser significativamente melhorados, quando esse perfil é utilizado para modelagem desse tipo de software.

O restante deste trabalho está organizado da seguinte forma: na Seção 2 são apresentados alguns conceitos básicos sobre JEDs e Perfis UML. Na Seção 3 estão os principais trabalhos relacionados ao tema dessa pesquisa. Na Seção 4 encontra-se uma descrição detalhada do perfil UP₄EG, bem como da metodologia utilizada para sua confecção. Na Seção 5 estão o planejamento, a execução e a análise dos resultados de um estudo experimental conduzido sobre o perfil proposto. Por fim, na Seção 6 estão as considerações finais e as propostas para trabalhos futuros.

2 *Background*

2.1 UML

Modelos de software são utilizados para auxiliar os usuários e desenvolvedores na visualização, compreensão e comunicação das ideias do projeto a outras pessoas, sendo considerados como uma representação simplificada da realidade (Booch *et al.*, 2005). Um bom modelo abrange as principais características do projeto e omite as informações consideradas irrelevantes para a confecção do mesmo (Sommerville, 2011). Uma das linguagens mais conhecidas e utilizadas para a representação gráfica de modelos de software é a UML (*Unified Modeling Language*). UML é uma linguagem de modelagem visual para especificação, construção e documentação de sistemas complexos de software (Booch *et al.*, 2005).

Um diagrama tem como função representar graficamente elementos de um modelo, sendo elaborado para fornecer a visualização do sistema em diferentes perspectivas (Sommerville, 2011). A UML oferece 13 (treze) diagramas para representação de diversos tipos de modelos. Contudo, de acordo com uma pesquisa realizada com os usuários da UML (Erickson, 2007 *apud* Sommerville, 2011, p. 83), verificou-se que apenas cinco desses diagramas são considerados suficientes para a representação de sistemas de software, são eles: (i) **diagrama de atividades**: determina o fluxo de controle entre uma atividade e outra do software; (ii) **diagrama de casos de uso**: descreve cenários de uso, que representam a funcionalidade do software do ponto de vista do usuário; (iii) **diagramas de sequência**: descreve o comportamento dos objetos do software ao longo de um processo, apresentando as mensagens trocadas por eles; (iv) **diagramas de classes**: descreve a estrutura e os relacionamentos entre as classes do software; e (v) **diagramas de estado**: representa os estados em que o software pode estar, de acordo com o fluxo de execução do mesmo. De acordo com essa mesma pesquisa, o diagrama de classes é, em geral, o mais utilizado para o desenvolvimento de modelos de softwares (Erickson, 2007 *apud* Sommerville, 2011, p. 83).

A Figura 1 apresenta o exemplo de um diagrama de classes que representa o relacionamento entre docentes e discentes em um software acadêmico hipotético. Uma classe é composta por três repartições, a primeira (superior) refere-se ao nome da classe, como por exemplo, Pessoa; a segunda (meio) é responsável pelos atributos (características) dos objetos



que serão gerados a partir dessa classe. Neste exemplo, um objeto do tipo “Pessoa” possui os atributos nome, CPF e dataNascimento; a terceira divisão (inferior) é onde são encontrados os métodos que implementam o comportamento dos objetos da classe. Por exemplo, objetos do tipo Pessoa possuem o método obterIdade, que é responsável por retornar a idade de uma determinada pessoa.

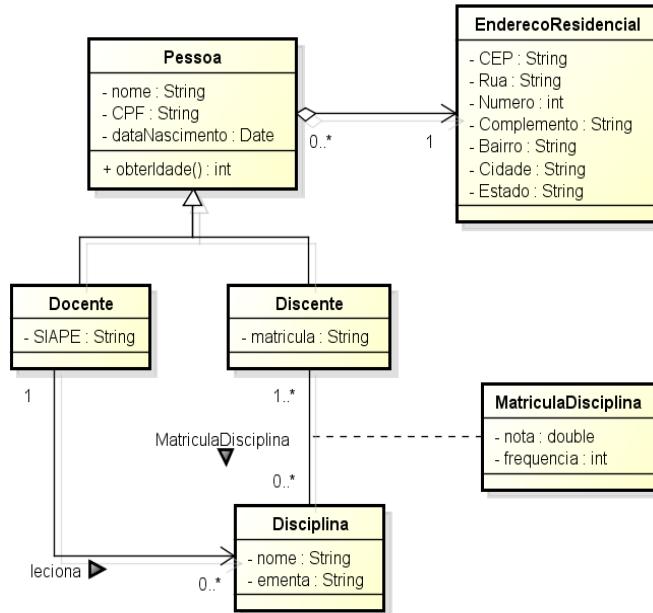


Figura 1. Exemplo de diagrama de classes.

Além dos elementos do tipo classe, o diagrama de classes oferece alguns tipos de associações para apresentar o relacionamento entre os objetos do software. Segundo Booch *et al.* (2005), essas associações podem ser classificadas como:

- **Generalização:** uma classe compartilha seus atributos, métodos e relacionamentos com outras classes de nível mais baixo na hierarquia de classes; na Figura 1, as classes Docente e Discente herdam os atributos, métodos e relacionamentos da classe Pessoa;
- **Associação:** representa uma ligação simples entre duas classes, representando que elas relacionam entre si. Por exemplo, o relacionamento entre Discente e Disciplina é representado por uma associação, expressando que um discente pode se matricular em zero ou várias disciplinas;
- **Agregação:** é uma associação especial, de forma que uma das classes do relacionamento representa objetos do tipo “todo” e a outra representa os objetos do tipo “parte”. Por exemplo, um objeto EnderecoResidencial faz parte de um objeto Pessoa (todo); e
- **Composição:** a composição também é um relacionamento onde um objeto-todo contém outros objetos-parte. Contudo, os objetos-parte dependem do objeto-todo para existir, ou seja, se o objeto-todo for removido, o objeto-parte também deverá ser removido. Simbolicamente, a composição é representada como uma agregação, isto é, uma seta com um losango do lado da classe que representa o objeto todo. Entretanto, no caso da composição, este losango é preenchido com a cor preta.

O diagrama de classes contempla ainda o conceito de multiplicidade. Segundo Booch *et al.* (2005), multiplicidade refere-se à quantidade de objetos de uma classe que podem estar relacionados com objetos das classes associadas a ela. Existem três diferentes tipos de uso para definir a multiplicidade entre classes associadas, sendo elas:



- **um para um:** um objeto de um lado do relacionamento só pode estar relacionado a um objeto da classe do outro lado. Por exemplo, em uma empresa cada funcionário deve usar apenas um computador e cada computador só pode ser usado por um funcionário;
- **um para muitos:** neste tipo de relacionamento um objeto de um lado do relacionamento deverá ter no mínimo uma incidência de objetos da classe do outro lado. Por exemplo, na Figura 1 há um relacionamento desse tipo entre Docente e Disciplina, pois um docente poderá lecionar zero ou várias disciplinas, mas cada disciplina só poderá ser lecionada por um docente; e
- **muitos para muitos:** um objeto de um lado do relacionamento poderá estar relacionado a várias ocorrências da classe do outro lado e vice-versa. Esse tipo de relacionamento ocorre entre as classes Discente e Disciplina, no qual um discente pode se matricular em várias disciplinas e uma disciplina poderá conter vários discentes.

Entretanto, como foi exposto na introdução deste artigo, apenas a utilização da UML tradicional na modelagem de domínios de problemas específicos pode prejudicar a abstração do mesmo por conter detalhes do projeto a ser desenvolvido que a UML não oferece mecanismos para modelagem. Para resolver este problema a UML permite a extensão de seus elementos para ampliar o significado dos modelos por meio de perfis UML.

2.2 Perfil UML

Apesar de a UML ser uma linguagem de propósito geral, *i.e.*, que pode ser utilizada em diferentes domínios, oferecendo ampla flexibilidade e expressividade na modelagem de sistemas de software, existem situações em que os elementos desta linguagem não são suficientes para expressar todas as particularidades de domínios específicos (Gérard, 2011).

Uma das possibilidades oferecidas para se definir linguagens de domínios específicos é a extensão da própria UML, por meio de um conjunto de mecanismos incluídos dentro dos chamados perfis UML (OMG, 2015). Um perfil UML é um conjunto de elementos que permitem a extensão da sintaxe e da semântica dos elementos pré-existentes na UML. Os principais elementos de um perfil UML são: *estereótipos*, *restrições* e *tagged values*. *Estereótipos* são como rótulos adicionados aos elementos de modelagem para destacar seu significado. Em geral, o nome de um estereótipo é representado entre os símbolos « » e ». *Restrições* são responsáveis por aplicar regras aos estereótipos, definindo o que pode e o que não pode ser feito em relação à modelagem de um determinado domínio. *Tagged Values*, por sua vez, permitem definir informações adicionais a um determinado estereótipo.

Para exemplificar os conceitos apresentados anteriormente, suponha que em uma aplicação de *e-commerce*, os produtos do carrinho de compras do usuário são apresentados em uma tabela. No diagrama de classes desse software, haveria então, uma classe denominada CarrinhoDeCompras, que corresponderia a este carrinho de compras. Em um perfil UML, pode-se criar um estereótipo denominado «Tabela», que, quando colocado sobre o identificador da classe ListaDeCompras, ampliaria o significado dessa classe (Figura 2), informando que, além de conter os produtos da compra de um usuário, ela será representada na interface do software por meio de uma tabela. Um possível exemplo de *tagged value* para o estereótipo «Tabela» seria maxResultados, cujo valor definiria a quantidade máxima de produtos a serem apresentados ao usuário de uma só vez. Além do estereótipo e do *tagged value*, poderia ser confeccionada uma restrição que impedisse o usuário de, durante a modelagem de um software com esse perfil, definir um valor para maxResultados menor do que 1 (um). Restrições, em geral, são representadas por meio da linguagem *Object Constraint Language* (OMG, 2017).

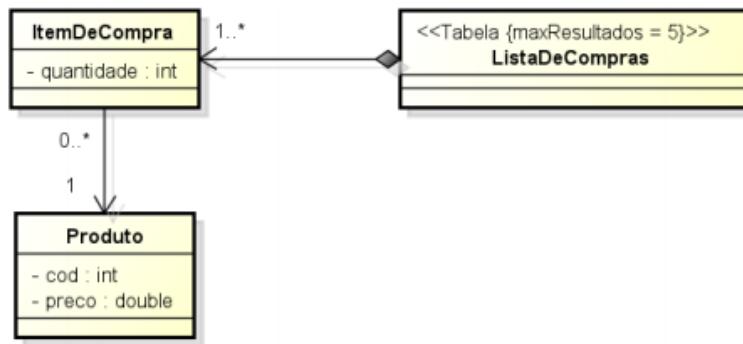


Figura 2. Exemplo de perfil UML hipotético para o domínio “e-commerce”.

2.3 Jogos Educacionais Digitais

Jogos Educacionais Digitais (JEDs) são softwares aplicados ao ensino de determinados conteúdos a grupos específicos de aprendizagem. Anetta (2010) apresenta um conjunto de seis elementos considerados essenciais para um bom jogo: *identidade*, *imersão*, *interatividade*, *aumento de complexidade*, *ensino informado* e *didática*.

Identidade é o fator capaz de “fazer o jogador acreditar” que ele ou ela é um indivíduo dentro do ambiente do jogo. Em muitos jogos, a identidade dos jogadores é representada por meio de um único personagem, denominado *avatar*. **Imersão** significa que os jogadores têm um senso de presença por meio de sua identidade, estão envolvidos no conteúdo e, com isso, ganham motivação para alcançarem o objetivo do jogo. **Interatividade** significa que os jogos devem permitir que seus jogadores se comuniquem socialmente, tanto com outros jogadores (por exemplo, em um ambiente *multiplayer*), quanto com a máquina, comunicando-se com agentes computadorizados não controlados por humanos, também conhecidos como NPC (*Non-Player Character*). Segundo o conceito de **aumento de complexidade**, um bom jogo deve possuir diversos níveis. A conexão entre esses níveis em um jogo oferece uma plataforma para o aumento da complexidade dos conceitos em um jogo educacional. **Ensino Informado** refere-se à análise constante das informações dos estudantes no jogo (*feedback*). Isso oferece uma forma de se analisar o modo como os alunos estão lidando com os conceitos ensinados. O elemento **didática** consiste na forma como são dispostos os elementos educacionais necessários para o aprendizado do aluno. Eles devem ser organizados de uma maneira clara para que os jogadores tenham condições de identificar as atividades que devem ser realizadas em determinado momento do jogo.

3 Trabalhos Relacionados

Esta seção discute alguns dos principais trabalhos relacionados ao tema deste artigo. Lisboa Filho et al. (2010) apresentam o “GeoProfile”, um perfil UML construído para a modelagem conceitual de bancos de dados geográficos. Segundo os autores, existem modelos consolidados na literatura para a modelagem conceitual de banco de dados geográficos, mas esses modelos não abordam todas as características consideradas relevantes para que se possa ter um entendimento adequado dos conceitos desse domínio. O trabalho de Torres et al. (2015), por sua vez, teve como objetivo a criação do perfil “MD-JPA”, que oferece uma extensão da UML para representação dos conceitos mais importantes sobre persistência de dados em Java, com base na especificação JPA (*Java Persistence API*). JPA é uma especificação da plataforma Java, que descreve uma interface comum para o desenvolvimento de frameworks de persistência de dados do tipo ORM (*Object-Relational Mapping*).

Já no trabalho de Uetanabara Júnior et al. (2010), os autores propuseram o perfil “UML-AOF”, cujo objetivo era tornar as características dos frameworks Orientados a Aspectos (FOA) mais evidentes. De todos os trabalhos comentados, esse foi o único em que os autores realizaram



uma avaliação do perfil proposto, comparando-o com outro perfil para modelagem de software orientado a aspectos, proposto por Evermann (2007). Os resultados dessa avaliação mostraram que, ao utilizar o perfil “UML-AOF”, menos erros foram cometidos durante a identificação dos principais elementos de um *framework* OO e que o tempo necessário para o entendimento de um modelo de classes de um FOA era menor, se comparado ao tempo gasto para realização da mesma tarefa com o perfil proposto por Evermann (2007). No contexto da modelagem conceitual de jogos, Zagal *et al.* (2005) apresentam a ontologia GOP (*Game Ontology Project*), cujo objetivo é descrever e classificar os elementos essenciais de um jogo.

Dos trabalhos comentados anteriormente, nenhum deles tem como objetivo a confecção de um perfil UML para Jogos Educacionais Digitais. Quanto à ontologia GOP, apesar de seu intuito ser auxiliar no entendimento dos principais elementos de um jogo, seu uso não é totalmente adequado para a modelagem de JEDs, uma vez que faltam elementos característicos do domínio educacional. Além disso, nenhum perfil UML foi proposto com base nessa ontologia, o que impede a utilização direta desses conceitos durante a modelagem de JEDs com a linguagem UML.

4 UP₄EG (*UML Profile for Educational Games*)

Especificar quais dos elementos citados anteriormente estarão presentes em um JED, logo na fase de concepção e modelagem do mesmo, pode levar os Engenheiros de Software a uma melhor reflexão sobre o que pretende-se desenvolver, bem como sobre possíveis problemas de *design* do jogo, antes mesmo de ele ser implementado. Com o intuito de se alcançar esses benefícios, desenvolveu-se um perfil para apoiar a modelagem de JEDs, denominado UP₄EG (*UML Profile for Educational Games*).

Para a criação do perfil UP₄EG, foi realizada a análise da documentação de trinta e quatro Jogos Educacionais Digitais (Tabela 1), com o intuito de se identificar como os elementos descritos na Seção 2 deste trabalho são modelados nesses jogos. É importante salientar que perfis UML podem ser elaborados para qualquer tipo de diagrama disponibilizado na UML. Neste trabalho, escolheu-se confeccionar um perfil para o diagrama de classes da UML, pois, segundo Sommerville (2011), este é um dos diagramas mais utilizado para a modelagem de software em geral. Além disso, a documentação da maioria dos JEDs utilizados como base para a confecção do perfil proposto neste trabalho disponibilizava apenas seu diagrama de classes.

Tabela 1. JEDs utilizados como base para a confecção do perfil UP₄EG.

Categoría	Jogos Educacionais Digitais
Esporte	Billiard Game Physics (Curi, 2012), Ahunt (2017).
Estratégia	Abalone (2017), Dalek World (Duggan, 2015), Sokoban (Li, 2014).
Infantil	Eduka (2017), Ens. De Matemática (Mwangi, 2011), Ens. De Geografia (Lahoud, 2011), Contar é o Bicho (Mendes Junior, 2012), Heróis da Matemática (Mendes Junior, 2012), Qual letra é, Jacaré? (Mendes Junior, 2012), Maior ou Menor (Mendes Junior, 2012), Silabando (Mendes Junior, 2012), Tabuada Contra o Tempo (Mendes Junior, 2012), THOT (Teramoto, 2008).
Investigação	Jogo das Sete Falhas (Diniz, 2011), Mathemá (Aguiar, 2012), SimInvestigador (Lessa Filho, 2014).
Quis	Fisicare (Jácome, 2003), Sócio-Histórico (Avelar, 2012), Multiagente (Vilela, 2011), SIEP (Silva, 2012), JOE (Paschoal, 2014).
Missão	ABC (Mendes Junior, 2012), Macaco Acadêmico (Lobo, 2012), Treinamento para Gerentes de Projeto de Software (Dantas, 2013).
Simulação	CidTrans (Azzolini, 2014), Touck Brush Game (Rodrigues, 2011), Educação no Trânsito (Cuba, 2009), ITest Learning (Sousa, 2012), Percepção Musical (Matunobu, 2010), Tricky Trig (Tresohlavy, 2007), Monsterismus (2015), Reeducação Alimentar (Jardim, 2013), Software Development Manager - SDM (Kohwalter, 2011).

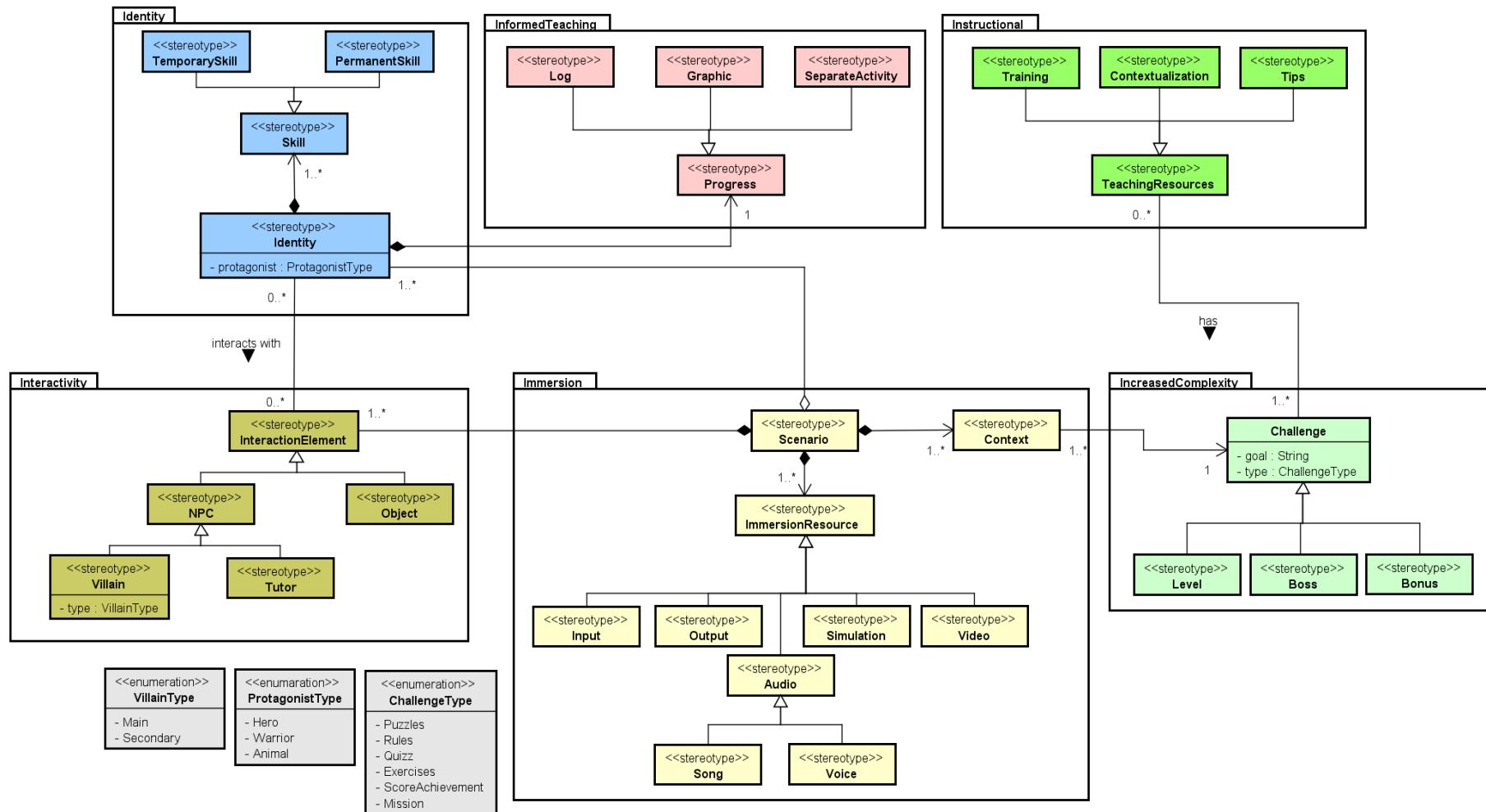


Para obtenção desses jogos, inicialmente, foi realizada uma busca em repositórios de código aberto, como *Github* (2017), *Sourceforge* (2017), entre outros. Foram encontrados quatro jogos nesses repositórios e, como nenhum deles disponibilizava seu diagrama de classes, foi realizada a confecção manual dos mesmos, a partir do código fonte disponível. Posteriormente, realizou-se uma busca por propostas de JEDs em repositórios de monografias, dissertações, teses e artigos científicos. A partir disso, trinta jogos educacionais foram encontrados, os quais já apresentavam um diagrama de classes elaborado pelos autores dos jogos.

A partir da análise dos trinta e quatro JEDs da Tabela 1, chegou-se a um conjunto de pacotes, estereótipos, *tagged values* e enumeradores (*enums*) que podem ser utilizados para a confecção de diagramas de classes UML de JEDs (Figura 3). Os estereótipos do perfil proposto são representados como classes desse diagrama e seus *tagged values*, como atributos. Para gerar o perfil proposto, os seguintes procedimentos metodológicos foram realizados: (i) cada elemento de um bom JED, descrito na Seção 2.3, foi transformado em um pacote; (ii) o código-fonte e/ou diagrama de classes dos jogos selecionados foram analisados pelos autores desse artigo, individualmente, com o intuito de identificar as principais entidades de cada jogo e mapeá-las aos pacotes criados anteriormente. Para esse mapeamento, procurou-se encaixar cada elemento no pacote em que a maioria dos autores indicou (os casos em que houve empates foram resolvidos em uma reunião de consenso); e (iii) por fim, elementos que possuíam características semelhante foram agrupados em uma única entidade. Esse agrupamento foi realizado em reunião de consenso, com a presença de todos os autores do artigo. Por exemplo, o estereótipo *TemporarySkill*, apresentado na Figura 3, foi obtido a partir da observância de vários elementos presentes nos jogos analisados que tinham como finalidade “ser uma habilidade temporária de um jogador”.

Identity (Identidade) - nesse pacote está o conjunto de estereótipos relacionados à identidade de um jogador; o estereótipo «*Identity*» é responsável por destacar em qual classe está modelado o *avatar* que representa o jogador. Por meio do *tagged value* *protagonist*, o engenheiro de software pode modelar o tipo de *avatar* que representará o jogador. Esse tipo é definido pela enumeração *ProtagonistType*, que contém os seguintes valores: *Hero*, *Warrior* e *Animal*. Além disso, um *avatar* poderá apresentar habilidades no jogo, representadas pelo estereótipo «*Skill*»; essas habilidades são divididas em dois tipos: (i) habilidades temporárias, representadas pelo estereótipo «*TemporarySkill*»: são consideradas habilidades necessárias para que o jogador conclua pequenos objetivos dentro do jogo. Por exemplo, um *avatar* do tipo guerreiro precisa derrotar um tipo diferente de vilão em cada fase, e, para isso, necessita de habilidades diferentes; e (ii) habilidades permanentes - estereótipo «*PermanentSkill*»: são consideradas habilidades em que, uma vez conquistadas, o jogador não as perde.

Interactivity (Interatividade) – esse pacote representa o conjunto de estereótipos responsáveis por dar significado aos elementos do jogo que irão interagir com o jogador. Sendo assim, há um relacionamento entre os estereótipos «*Identity*» e «*InteractionElement*». Os elementos de interação são representados por dois estereótipos: (i) «*NPC*», que possui como subclasses, os estereótipos «*Villain*» e «*Tutor*». «*Villain*» representa os vilões do jogo; como em um jogo podem existir diversos vilões, tal informação é representada no perfil por meio do *tagged value* *type* e da enumeração *VillainType*, cujos possíveis valores são *Main* e *Secondary*. Por exemplo, no jogo “Macaco Acadêmico” (Lobo, 2012), o jogador deverá coletar frutas para enfrentar diversos monstros existentes no jogo, que podem ser representados pelo estereótipo «*Villain*» com o valor *Secondary* para o atributo *type*; e (ii) «*Object*», que representa os objetos no mundo do jogo, os quais serão passíveis de interação com o jogador, como por exemplo, uma pedra, uma árvore, uma espada, entre outros.

Figura 3. O perfil UP₄EG.



Immersion (Imersão) – nesse pacote estão os estereótipos responsáveis por caracterizar os elementos relacionados à imersão. O estereótipo «Scenario» pode ser usado para modelar os elementos do jogo que caracterizam o cenário em que se desenrolará todos os seus acontecimentos. Um elemento modelado como «Scenario» pode conter diversos elementos do tipo «InteractionElement» e «Identity» (se for considerada uma plataforma *multiplayer*). Como pode-se perceber na Figura 3, o estereótipo «Scenario» possui relacionamentos de agregação e composição com alguns dos estereótipos considerados importantes para a representação dos elementos de JEDs, esses relacionamentos possuem o mesmo significado dos relacionamentos aplicados em diagramas de classes. Além disso, cada cenário pode estar relacionado a um ou mais elementos do tipo «Context». Por exemplo, no jogo “CidTrans” (Azzolini, 2014), cujo objetivo é apoiar o ensino de leis de trânsito, o jogador deve dirigir um carro em um tráfego simulado. Cada fase do jogo representa um cenário diferente (uma cidade de pequeno, médio ou grande porte), mas em cada cenário, o jogador poderá concluir diferentes desafios (representados pelo estereótipo «Challenge»), de acordo com o contexto atual daquela fase, ou seja, em cada fase poderá haver contextos distintos, com também desafios distintos.

Além disso, o cenário deve ter alguns mecanismos que irão permitir saídas de áudio e vídeo ao usuário, além de suportar a entrada de dados e permitir um *feedback* de acordo com a entrada realizada. Todos esses elementos contribuem para a imersão do jogador e são representados por meio das subclasses do estereótipo «ImmersionResource»: (i) «Input» e «Output» são responsáveis pela entrada e saída de dados no jogo; (ii) «Simulation» tem como objetivo representar simulações existentes em um jogo. Por exemplo, o jogo “Fisicare” (Jácome, 2003), que possui como objetivo auxiliar no ensino de física, permite ao aluno visualizar simulações de experimentos físicos; (iii) «Animation» é utilizado para representar uma pequena história do jogo ao jogador, com o propósito de introduzi-lo no contexto do jogo. Isso é perceptível no jogo “Sócio-Histórico” (Avelar, 2012), em que são apresentadas telas com o intuito de contextualizar o jogador diante de uma história; e (iv) «Audio» é responsável pela representação dos sons do jogo, sendo eles músicas de fundo («Song») ou vozes («Voice»), tanto de personagens como de narradores.

IncreaseadComplexity (Aumento de complexidade) – o estereótipo «Challenge», contido nesse pacote, é responsável por representar os tipos de desafios abordados no jogo, bem como o objetivo dos mesmos, respectivamente modelados pelos seus *tagged values type* e *goal*. Além disso, os estereótipos «Level», «Boss» e «Bonus» são tipos de desafios que podem ser mais bem explicitados por meio do enumerador «ChallengeType». Por exemplo, em um jogo, poderia haver um desafio do tipo “avançar de fase”, que corresponderia ao estereótipo «Level». Mas esse desafio (passar de fase) poderia exigir a realização de um quebra-cabeça (*puzzles*) ou finalização de uma missão (*mission*) dada ao jogador. Assim, o *tagged value type* pode auxiliar na modelagem desse cenário. O *tagged value goal*, por sua vez, representa um texto que descreve o objetivo do desafio, como por exemplo, “finalizar a missão no menor tempo possível”, “finalizar o quebra-cabeça com o menor número de erros”, entre outros.

Instructional (Didática) – o pacote *Instructional* é responsável por representar o conjunto de estereótipos capazes de explicitar quais são os elementos de aprendizagem a serem abordados no jogo. Tais objetos instrucionais são representados por meio do estereótipo «TeachingResources», que tem relacionamento com o estereótipo «Challenge». Isso se justifica, pois como os desafios são uma das principais características para a imersão do jogador, quanto mais envolvido o aluno estiver, mais ele estará aprendendo novos conteúdos e fixando os já obtidos. Contudo, nem todo desafio precisa estar diretamente relacionado a um elemento instrucional.

O estereótipo «TeachingResources» possui três subclasses: (i) «Training», que representa elementos instrucionais do tipo treinamento. Geralmente são usados quando o



jogador não possui familiaridade com o conteúdo instrucional a ser abordado no jogo; (ii) «Contextualization», responsável por contextualizar (problematizar) o conteúdo a ser contemplado no jogo. Esta característica pode ser percebida, por exemplo, nos jogos “SDM” (Kohwalter, 2011), “Jogo das Sete Falhas” (Diniz, 2011) e “Treinamento para Gerentes de Projeto de Software” (Dantas, 2013); e (iii) «Hints», que representa as dicas que são oferecidas durante o jogo para auxiliar no cumprimento de algum desafio. No jogo “Jogo Ortográfico Educacional – JOE” (Paschoal, 2014), por exemplo, quando o aluno comete algum erro, são apresentadas dicas para possíveis soluções da atividade.

InformedTeaching (Ensino informado) – nesse pacote estão os estereótipos responsáveis por representar o desempenho do aluno no jogo. O principal estereótipo desse pacote é «Progress», que possui três subclasses: (i) «Log»: representa as ações do aluno no jogo, geralmente armazenada em arquivos de *log*; (ii) «Graphic»: representa os resultados obtidos pelos alunos no jogo por meio de gráficos, de forma a fornecer ao professor uma interpretação dos dados de forma mais clara, rápida e objetiva; e (iii) «SeparateActivity»: representa o desempenho dos alunos em pequenas atividades no jogo como por exemplo, número de respostas corretas em uma fase, número de moedas coletadas, quantidade de pontos obtidos, entre outras coisas.

5 Aplicação do UP₄EG no jogo educacional “SDM” (*Software Development Manager*)

Software Development Manager (SDM) é um jogo educacional para a disciplina de Engenharia de Software, cujo objetivo é aprimorar a aprendizagem dos conceitos e práticas relacionados à gestão de pessoas, incluindo capacitação, carga de trabalho e o papel de cada empregado no projeto de desenvolvimento de um software. Para isso, o “SDM” propõe a utilização de simulação de um ambiente de trabalho de desenvolvimento de software (Kohwalter, 2011).

Como dito anteriormente, o “SDM” possui como foco a gestão de pessoas, em que os principais elementos do jogo são os empregados. Cada funcionário desempenha uma função dentro do grupo de desenvolvimento, sendo essas características adicionadas ou melhoradas durante o andamento do jogo. Essas características incluem mudanças em possíveis papéis que o empregado poderá desempenhar e os atributos usados para calcular o progresso do mesmo. Outro elemento existente no jogo é a especialização, usada para definir o nível de competência no trabalho de cada empregado. Na Figura 4 é apresentado o diagrama de classes do jogo “SDM” com a aplicação do perfil UP₄EG.

Com a aplicação do UP₄EG, novas informações, inerentes ao domínio de jogos, podem ser identificadas apenas por meio da visualização do diagrama de classes do jogo. Desta forma, as informações originais têm seu significado estendido a fim de melhorar o entendimento sobre o projeto a ser desenvolvido. A classe Employee (Figura 4) é responsável por representar os empregados do jogo. Ela foi estereotipada como «Identity», uma vez que o jogador também faz parte da equipe de desenvolvimento do projeto. Com esse estereótipo, a equipe de desenvolvimento e manutenção do jogo pode perceber mais facilmente que essa classe, além de representar os empregados, também representa a identidade do jogador, informação que não poderia ser obtida apenas visualizando o diagrama de classes puro, ou seja, sem a aplicação do UP₄EG.

A classe Employee se relaciona com a classe Training, estereotipada com «Training», uma vez que essa classe é responsável pelo treinamento dos empregados da equipe, incluindo o próprio jogador. A classe Training, por sua vez, se relaciona com Specialization. Essa última diz respeito às especialidades que cada membro da equipe poderá dominar, como por exemplo, um novo tipo de linguagem de programação, uma metodologia de desenvolvimento de

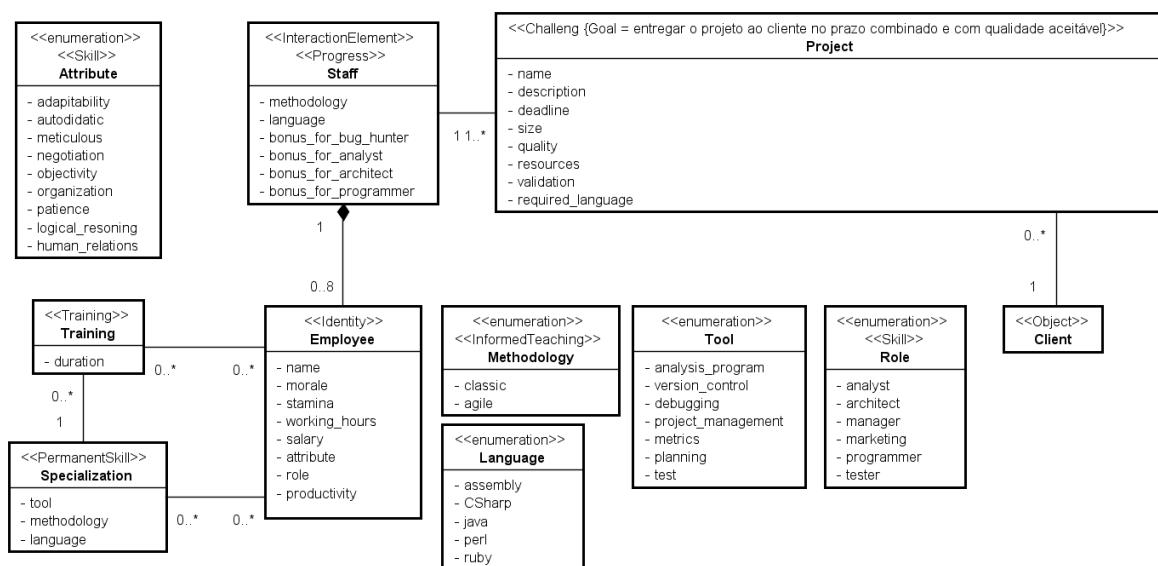


Figura 4. UP₄EG aplicado ao diagrama de classes do jogo “SDM”.

A classe **Staff** é responsável por representar a equipe de desenvolvimento da empresa como um todo. Essa classe recebeu os estereótipos «**InteractionElement**», uma vez que ela possui como característica a representação de um tipo de elemento de interação, isto é, o jogador interage com a equipe de desenvolvimento para atingir o objetivo final. Esta classe também contém informações sobre os resultados (bônus) conquistados pelos funcionários, a partir das atividades completadas com sucesso durante o desenvolvimento do projeto. Sendo assim, ela também foi estereotipada com «**Progress**», pois representa o progresso do usuário no jogo.

O estereótipo «Challenge» se encontra aplicado à classe Project. A *tagged value goal* descreve o real objetivo daquele desafio, que entregar o projeto pronto ao cliente, que é representado pela classe Client, na data combinada e com nível de qualidade aceitável. A classe Client recebeu o estereótipo «Object», pois representa um elemento de interação no jogo, mas um elemento estático, que serve apenas para descrever as características do projeto.

6 Estudo Experimental

O estudo experimental descrito nesta seção teve como objetivo verificar a eficiência e a eficácia da aplicação do perfil UP4EG quanto à correta identificação dos elementos de um JED e quanto ao tempo gasto para a identificação desses elementos. No contexto desse trabalho, eficácia está relacionada à quantidade de respostas corretas dadas pelos participantes do experimento, a perguntas que analisam o seu entendimento com relação aos elementos de um JED modelados em um diagrama de classes UML.

O perfil UP₄EG foi comparado com um perfil UML desenvolvido pelos autores desse trabalho, a partir da ontologia para jogos (GOP), proposta por Zagal *et al.* (2005). Isso foi necessário, pois até o momento da realização deste estudo (Março/2017), não havia outros perfis UML similares ao proposto neste trabalho para serem utilizados durante o processo de avaliação. O perfil desenvolvido foi denominado Perfil GOP, por ser baseado na ontologia GOP (*Game Ontology Project*), cujo objetivo é descrever e classificar os elementos essenciais de um jogo em geral. Assim, foi possível analisar as vantagens e desvantagens trazidas pela aplicação



de um perfil UML confeccionado especificamente para JEDs (UP₄EG) em relação a um perfil de propósito geral.

O Perfil GOP representa as principais características da ontologia GOP, sendo elas: (i) **interface**: responsável por mapear as reações do jogador e a manipulação de entidades no jogo; (ii) **regras**: são características que definem e restringem o que pode e o que não pode ser feito em um jogo; (iii) **objetivos**: consistem em objetivos ou condições que definem o sucesso do jogador em um jogo; e (iv) **manipulação de entidades**: é caracterizada pelas ações que podem ser feitas tanto pelos jogadores, quanto pelas entidades que modificam o estado do jogo. Cada característica possui vários elementos, os quais foram convertidos em estereótipos UML pertencentes ao Perfil GOP. Devido à limitação de espaço, uma descrição mais completa sobre o Perfil GOP foi omitida deste trabalho, contudo, mais informações podem ser encontradas em Oliveira (2016).

6.1 Planejamento do Estudo Experimental

O estudo experimental proposto neste trabalho foi conduzido com 10 (dez) graduandos do curso de Ciência da Computação da Universidade Federal de Goiás/Regional Jataí. Para realizar a avaliação dos perfis, diagramas de classes de dois jogos educacionais foram escolhidos, dentre os trinta e quatro diagramas de classes analisados neste trabalho. A escolha foi feita de forma que os diagramas de classes dos dois jogos possuíssem quantidade aproximada de classes e elementos relacionados aos JEDs.

O estudo experimental foi conduzido com base no projeto apresentado na Tabela 2. Com o intuito de homogeneizar o conhecimento dos participantes do estudo, antes da execução do experimento, foi realizado um **treinamento** sobre modelagem de software, diagramas de classe UML e sobre os perfis UP₄EG e GOP.

Cada jogo continha um diagrama de classes ao qual foram aplicados o UP₄EG e o Perfil GOP. Após o treinamento sobre perfis UML, os participantes do experimento foram divididos em dois grupos de cinco, a saber, *Grupo A* e *Grupo B*. Durante a **primeira fase** do estudo experimental, o *Grupo A* recebeu o diagrama de classes do jogo “CidTrans” (Azzolini, 2014), modelado com base no perfil UP₄EG e o *Grupo B* recebeu o diagrama de classes do mesmo jogo, modelado de acordo com o Perfil GOP. Na **segunda fase**, o *Grupo B* recebeu o diagrama de classes do jogo “SimInvestigador” (Lessa Filho, 2014), modelado com auxílio do perfil UP₄EG, e o *Grupo A* recebeu o mesmo diagrama, porém modelado com o Perfil GOP. Os diagramas de classes dos jogos utilizados no estudo experimental encontram-se no trabalho de (Oliveira, 2016). O jogo “SimInvestigador” possui como objetivo auxiliar o professor no ensino por meio da resolução de casos contextualizados em uma narrativa associada ao conteúdo de uma disciplina.

Tabela 2. Design do estudo experimental.

Fase	Grupos	
	A	B
Treinamento	Modelagem de Software, UP ₄ EG e Perfil GOP	Modelagem de Software, UP ₄ EG e Perfil GOP
1 ^a Fase	Jogo “CidTrans” + Perfil UP ₄ EG	Jogo “CidTrans” + Perfil GOP
2 ^a Fase	Jogo “SimInvestigador” + Perfil GOP	Jogo “SimInvestigador” + Perfil UP ₄ EG

Em cada fase do experimento, foi entregue um questionário aos participantes, que deveria ser respondido após a análise dos diagramas de classes apresentados a eles. Tal questionário continha seis questões relacionadas à identificação e à compreensão dos elementos de um JED modelados nesses diagramas. As questões abordavam: (i) **Questão 1** – a



identificação da identidade do jogador; (ii) **Questão 2** – a identificação de agentes responsáveis pela interação com o jogador; (iii) **Questão 3** – a identificação dos elementos físicos do jogo; (iv) **Questão 4** – a identificação dos objetos instrucionais abordados pelo jogo; (v) **Questão 5** – a identificação dos elementos responsáveis pelo aumento da complexidade; e (vi) **Questão 6** – a identificação dos elementos responsáveis pelos resultados do aluno.

A Tabela 3 apresenta o conjunto completo de questões feitas aos participantes do estudo experimental. As questões utilizadas foram confeccionadas por 2 (dois) pesquisadores com experiência nos temas de JEDs e perfis UML, utilizando como base as questões propostas no trabalho de Uetanabara Júnior et al. (2010). É importante ressaltar que esses pesquisadores não foram participantes do experimento para avaliação do perfil proposto.

6.2 Resultados e Discussão

Na Tabela 4 são apresentados os dados coletados na primeira fase do estudo experimental, isto é, a aplicação dos perfis UP₄EG e GOP no jogo “CidTrans”. Cada célula dessa tabela apresenta a quantidade de participantes do experimento que acertou determinada questão do questionário proposto na Tabela 3. As respostas dadas pelos participantes do experimento foram analisadas e corrigidas por especialistas nos assuntos de modelagem de software e jogos educacionais.

O símbolo “-”, quando presente nesta tabela, indica que a sub-questão analisada não faz parte da questão em análise. Por exemplo, a questão 1 não possui sub-questões E, F ou G, sendo assim, essas colunas receberem o símbolo “-”. A coluna com o símbolo “%” representa a porcentagem de acertos para cada questão, com base na quantidade máxima de acertos por questão.

Por exemplo, a questão 1 possui 4 (quatro) sub-questões, sendo assim, sabendo que a quantidade de participantes por grupo é 5 (cinco), a maior pontuação possível para a questão 1 é 20 (vinte) pontos, no caso de todos os cinco participantes acertarem a todas as sub-questões. Uma vez que a questão 1 obteve 17 (dezessete) acertos, seu percentual de acerto é de 85%.

Tabela 3. Questões feitas aos participantes do estudo experimental.

#	Questões
1	Você consegue identificar algum personagem que representa a identidade do jogador no jogo? Caso sim, responda: A. Que tipo de personagem é esse (pessoa, animal, entre outros)? B. Qual(is) elemento(s) do diagrama são responsáveis por representar a identidade do jogador? C. Quais são as habilidades desse personagem? D. Qual(is) elemento(s) do diagrama é (são) responsável(is) por representar as habilidades do jogador?
2	Há outro(s) personagem(ns) no jogo? Caso sim, responda: A. Que tipos de personagens são esses (vilão, NPC, chefe de fase, entre outros?)? B. Esses personagens podem interagir com o jogador? C. Qual(is) elemento(s) do diagrama são responsáveis pela representação desses jogadores? D. Quais são as formas (mecanismos) de comunicação entre o jogador e os demais personagens do jogo? E. Qual(is) elemento(s) do diagrama são responsáveis pela representação desses mecanismos? F. Quais são as habilidades desse(s) personagem(ns)? G. Qual(is) elemento(s) do diagrama são responsáveis por representar as habilidades desses personagens?
3	Há objetos físicos no jogo? Caso sim, responda: A. Que tipos de objetos são esses (árvores, pessoas, automóveis, entre outros)? B. Qual(is) elemento(s) do diagrama são responsáveis por representar esses objetos? C. Eles podem interagir com o jogador e/ou com outros personagens do jogo? D. Quais são as formas (mecanismos) de interação entre os personagens do jogo e os objetos? E. Qual(is) elemento(s) do diagrama são responsáveis pela representação desses mecanismos?
4	Há objetivos educacionais claros no jogo? Caso sim, responda: A. Que tipos de objetivos são esses (perguntas e repostas, acúmulo de moedas, entre outros)? B. Qual(is) elemento(s) do diagrama são responsáveis por representar esses objetivos?



	C. Eles podem interagir com o jogador e/ou com outros personagens do jogo? D. Quais são as formas (mecanismos) de interação entre os personagens do jogo e os objetivos educacionais? E. Qual(is) elemento(s) do diagrama são responsáveis pela representação desses mecanismos?;
5	O jogo provê mecanismos para escolha do grau de complexidade (como por exemplo, baixa, média, alta)? Qual(is) elemento(s) do diagrama são responsáveis pela representação desses mecanismos?
6	O jogo provê mecanismos para feedback de desempenho do jogador ao professor? Qual(is) elemento(s) do diagrama são responsáveis pela representação desses mecanismos?

Tabela 4. Resultados da primeira fase do estudo experimental.

Grupo 1 – CidTrans + Perfil UP ₄ EG								
Questões	Sub Questões							%
	A	B	C	D	E	F	G	
1 ^a	5	3	4	5	-	-	-	85
2 ^a	4	3	4	5	2	4	4	74
3 ^a	5	2	5	5	3	-	-	80
4 ^a	4	3	3	4	2	-	-	64
5 ^a			4					80
6 ^a			3					60

Grupo 2 – CidTrans + Perfil GOP								
Questões	Sub Questões							%
	A	B	C	D	E	F	G	
1 ^a	5	0	2	4	-	-	-	55
2 ^a	2	0	4	1	1	2	3	37
3 ^a	5	2	4	0	0	-	-	44
4 ^a	0	0	2	0	0	-	-	08
5 ^a			1					20
6 ^a			0					00

A quantidade de acertos na primeira fase do experimento utilizando o UP₄EG foi consideravelmente maior do que utilizando o Perfil GOP. Isso pode ser explicado pelo fato que o perfil UP₄EG pôde ser aplicado a todas as classes do jogo “CidTrans”, permitindo assim, a extensão da informação relacionada a essas classes do diagrama. Por outro lado, como o perfil GOP é genérico, isto é, ele não é específico para JEDs, seus elementos puderam ser aplicados a apenas uma parte do diagrama de classes do jogo. Isso fez com que os participantes tivessem menos informações, além daquelas disponibilizadas no diagrama de classes convencional, para responder às questões propostas.

Um fato que chama a atenção é que tanto o perfil UP₄EG quanto o perfil GOP tiveram desempenhos insatisfatórios para a **Questão 4**, que diz respeito aos objetivos educacionais do jogo. Isso é compreensível para o caso do perfil GOP, que não possui enfoque educacional, e, para o perfil UP₄EG, gera-se um indício de que essa característica de um JED deve ser mais bem explorada e especificada nesse perfil, com o intuito de aprimorar o entendimento dos elementos pertencentes a ela.

Os resultados da segunda fase do experimento estão na Tabela 5. Como pode ser observado, apesar de o perfil UP₄EG ter tido um desempenho melhor do que o do perfil GOP também nessa fase, sua porcentagem de acertos reduziu em praticamente todas as questões, se comparada com o resultado da primeira fase. De modo análogo, o desempenho do perfil GOP também reduziu, em comparação à primeira fase do estudo experimental. Isso pode ter acontecido devido ao tipo de jogo utilizado, isto é, o jogo “CidTrans” pertence a um domínio mais simples e mais próximo da realidade dos participantes do experimento, que é o ensino de leis de trânsito, por isso, pode ter levado a um resultado melhor. Contudo, essa afirmação deve ser mais bem investigada por meio da condução de outros estudos experimentais.

A questão com maior porcentagem de acertos para o perfil UP₄EG, nas duas fases do estudo, é a **Questão 1**, que diz respeito à identificação da identidade do jogador. Isso pode ser explicado pelo fato que a identidade do jogador, em geral, é mais bem modularizada e especificada no projeto dos jogos. Dessa forma, o uso do estereótipo pôde trazer informações claras e úteis para entendimento desse tipo de característica.

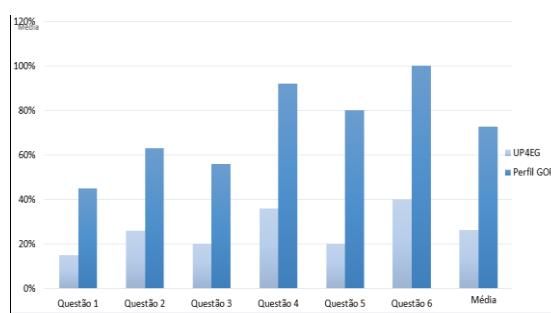
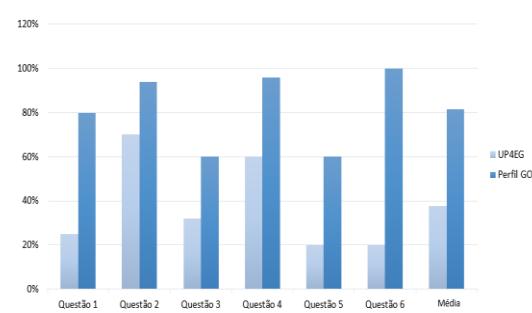


Tabela 5. Resultados da segunda fase do estudo experimental.

Questões	Sub Questões							% %
	A	B	C	D	E	F	G	
1 ^a	2	2	0	0	-	-	-	20
2 ^a	1	1	0	0	0	0	0	05
3 ^a	2	2	2	3	1	-	-	40
4 ^a	1	0	0	0	0	-	-	04
5 ^a			2					40
6 ^a			0					00

Questões	Sub Questões							% %
	A	B	C	D	E	F	G	
1 ^a	4	3	4	4	-	-	-	75
2 ^a	3	2	2	1	2	0	1	31
3 ^a	4	3	4	5	1	-	-	68
4 ^a	3	2	1	2	2	-	-	40
5 ^a			4					80
6 ^a			4					80

Os gráficos das Figuras 5 e 6 sumarizam os resultados obtidos na primeira e segunda fases do experimento, respectivamente. Ao contrário das Tabelas 4 e 5, esses gráficos apresentam a porcentagem de respostas **incorrectas**, por questão, dadas pelos participantes que utilizaram diagramas modelados com os perfis analisados, a saber, UP4EG e GOP.

Figura 5. Resultados - 1^a faseFigura 6. Resultados - 2^a fase

Um resultado que chamou a atenção, devido às discrepâncias existentes entre os valores para os dois perfis, é o da **Questão 5**, na primeira fase do experimento. Essa questão pede ao participante que informe se o jogo possui mecanismos para escolha de seu grau de complexidade e pede que ele aponte quais elementos do diagrama são responsáveis por representar esses mecanismos. A resposta esperada para essa questão seria que o jogo provê tal escolha e que a mesma é representada pela classe Cenario. Apesar de a ontologia GOP tratar da modelagem conceitual de jogos, o elemento “aumento de complexidade” é um conceito importante, principalmente, em jogos educacionais. Sendo assim, tal perfil não possibilitou a correta identificação desse conceito para a maioria dos participantes. Quanto à segunda fase do estudo experimental, a **Questão 6**, relacionada à identificação dos elementos responsáveis pelo *feedback* do progresso do aluno no jogo, foi a questão em que houve maior discrepância entre os dois grupos. De modo análogo ao que ocorreu no exemplo anterior, o perfil GOP não ofereceu recursos adequados para a representação desses tipos de elemento, o que dificultou a resolução dessa questão por parte dos participantes.

Em resumo, a quantidade média de erros cometidos na primeira fase do experimento pelos participantes que utilizaram o perfil UP4EG foi consideravelmente menor do que a quantidade de erros dos participantes que utilizaram o Perfil GOP. Isso ocorreu, pois o perfil UP4EG pôde ser aplicado em todas as classes do jogo “CidTrans”, permitindo assim, a ampliação das informações presentes nessas classes, enquanto que o Perfil GOP pôde ser aplicado em apenas uma pequena parte do diagrama de classes. De modo análogo, a quantidade média de erros na segunda fase do estudo experimental, cometida pelo grupo que utilizou o perfil UP4EG também foi menor. Porém, mesmo utilizando o perfil UP4EG, ainda houve erros. Uma explicação para isso pode ser a falta de conhecimento dos participantes quanto aos jogos apresentados a eles, bem como a ambiguidade gerada pela nomenclatura de alguns estereótipos



desse perfil. Sendo assim, pretende-se replicar o experimento descrito nesta seção com outros participantes e com outros jogos, com o intuito de se obter resultados mais robustos.

Com relação ao tempo gasto pelos participantes na primeira fase para a identificação dos principais elementos dos JEDs abordados na avaliação, o grupo utilizando o perfil UP₄EG terminou o experimento com uma diferença de doze minutos de antecedência para o grupo utilizando o Perfil GOP. Essa diferença reflete na maior compreensão do jogo proporcionada pelo uso do perfil UP₄EG, pois o mesmo permite estender a informação de uma quantidade maior de classes do que o Perfil GOP. Da mesma forma, na segunda fase, o grupo utilizando o perfil UP₄EG terminou o experimento primeiro que o grupo utilizando o Perfil GOP, porém a diferença caiu para quatro minutos de antecedência. Isso pode ter ocorrido pela falta de um padrão para a modelagem de JEDs, o que acarretou na maior dificuldade de compreensão dos elementos do jogo “SimInvestigador”. Entretanto, nas duas fases da avaliação o tempo gasto pelo grupo utilizando o perfil UP₄EG foi menor, o que caracteriza a maior facilidade de identificação e compreensão dos elementos que compõem os JEDs analisados proporcionados pela utilização do perfil UP₄EG.

6.3 Ameaças à Validade

Como principais ameaças à validade do estudo experimental realizado, têm-se:

- (i) a pequena quantidade de amostras utilizadas no experimento e o fato de terem sido utilizados alunos de graduação: tais ameaças impedem a generalização dos resultados obtidos neste estudo, contudo, pretende-se replicar o experimento apresentado neste trabalho com um grupo maior de participantes, bem como com profissionais da área de desenvolvimento de jogos;
- (ii) possíveis problemas existentes nos materiais utilizados no experimento, tais como os diagramas de classe, questionários, entre outros: com o intuito de mitigar tal ameaça, todo material utilizado durante o experimento foi validado por profissionais com experiência em experimentação em engenharia de software, jogos educacionais e modelagem de software. Além disso, o material utilizado no experimento foi avaliado em um teste piloto, realizado com os autores deste artigo. O intuito do teste piloto é encontrar possíveis falhas nestes materiais e corrigi-las antes da execução do estudo experimental; e
- (iii) a escolha dos jogos utilizados durante a avaliação pode ter influenciado nos resultados: com o intuito de mitigar esse tipo de influência, os jogos foram selecionados de forma que tivessem “tamanhos” (quantidade de classes) similares, bem como uma quantidade de elementos educacionais também similar. Além disso, o fato de o projeto experimental contemplar duas fases, nas quais os dois grupos de participantes puderam analisar os dois jogos selecionados, minimiza a possibilidade de que haja qualquer tipo de privilégio de um jogo em favor de algum dos perfis analisados.

Outro ponto importante a ser considerado é que os participantes do estudo experimental não tiveram informações a respeito de qual perfil teria sido desenvolvido pelos autores deste trabalho, evitando assim, que um maior esforço fosse dispendido por parte dos participantes em favor de um ou de outro perfil avaliado.

7 Considerações Finais

Este trabalho apresentou detalhes da confecção de um perfil UML para a modelagem de classes de jogos educacionais digitais. Esse perfil foi elaborado com base em trinta e quatro jogos previamente selecionados na literatura e em repositórios de software livre. Os resultados do



estudo experimental conduzido sobre o perfil UP₄EG, e detalhado neste artigo, mostrou que há indícios de que esse perfil pode facilitar a identificação e entendimento dos principais elementos de um JED, por parte de seus desenvolvedores, a partir de um diagrama de classes UML.

Como trabalhos futuros, pretende-se: (i) implementar o perfil UP₄EG em alguma ferramenta de apoio à modelagem de software, como o Astah* (2017); (ii) incrementar o perfil proposto com novos elementos, que podem ser obtidos a partir da análise de outros tipos de jogos; (iii) propor a utilização do UP₄EG para a detecção de problemas de projeto de JEDs logo no início do desenvolvimento dos mesmos; e (iv) aplicar novos experimentos sobre o perfil proposto, com o intuito de ampliar a capacidade de generalização das afirmações descritas neste trabalho, bem como avaliar o impacto do uso do perfil sobre outras características de qualidade de um software, tais como legibilidade, reusabilidade, manutenibilidade, testabilidade, entre outras coisas.

Referências

- Abalone. Disponível em: <https://github.com/Keruspe/Abalone>. Acessado em: Maio/2017.
- Aguiar, T. e ALEXANDRE, E. O Uso de Jogos Educativos no Ensino da Matemática Elementar. 34 f. Monografia (Bacharelado em Sistemas para Internet) - Centro Universitário Adventista de São Paulo, Campus Engenheiro Coelho - São Paulo. 2012. [[GS Search](#)].
- Ahunt. “Android version of Electronica-IM game”. Disponível em: <https://github.com/mgrebenets/ahunt>. Acessado em: Maio/2017.
- Anneta, L. A. The “I’s” Have It: A Framework for Serious Educational Game Design. American Psychological Association. 2010. doi: [10.1037/a0018985](https://doi.org/10.1037/a0018985). [[GS Search](#)].
- Astah* CASE Tool. Disponível em: <http://astah.net/download>. Acessado em: Maio/2017.
- Avellar, F. T. Desenvolvimento de um jogo educacional de cunho sócio-histórico. 58 f. Monografia (Bacharelado em Ciência da Computação) - Universidade Estadual Paulista - Faculdade de Ciências, Bauru - São Paulo. 2012. [[GS Search](#)].
- Azzolini, V. L. CIDTRANS - Jogo 3D para educação no trânsito. 2014. 99 f. Monografia (Engenharia da Computação) - Universidade Tecnológica Federal do Paraná, Pato Branco. 2014. [[GS Search](#)].
- Booch, G. et al. The Unified Modeling Language User Guide, 2º edição. Rio de Janeiro, Elsevier, 2005.
- Cuba, G. P. Jogos sérios: tecnologia de jogos por computador aplicada ao ensino e aprendizagem. 91 f. Monografia (Bacharelado em Ciência da Computação) - Universidade Federal de Goiás - Campus Catalão. 2009. [[GS Search](#)].
- Curi, C. F. et al. Proposal and Development of a Computer Game for Teaching Physics. SBC - Proceedings of SBGames. 2012.
- Dantas, A. R. Jogos de Simulação no Treinamento de Gerentes de Projetos de Software. Dissertação (Mestre em Ciências em Engenharia de Sistemas e Computação) - Universidade Fereral do Rio de Janeiro, Rio de Janeiro. 2013. [[GS Search](#)].
- Diniz, L. L. Jogo das 7 Falhas: Um Jogo Educacional para o Apoio ao Ensino do Teste de Caixa-Preta. Dissertação (Mestre em Computação Aplicada) - Universidade do Vale do Itajaí, São José - Santa Catarina. 2011. [[GS Search](#)].



Duggan, Brian *et al.* Learning games programming with Dalek World. 7th. International Conference on Computer Games: AI, Animation, Mobile, Educational and Serious Games, CNBDI, 28-30th. November, Magelis, Angouleme, France. 2015. [[GS Search](#)].

Eduka. Disponível em: <https://github.com/janci/Eduka>. Acessado em: Maio/2017.

Evermann, J. A Meta-Level Specification and Profile for AspectJ in UML. Journal of Object Technology. Vol 6, No 7, Aug 2007. doi: [10.1145/1229375.1229379](https://doi.org/10.1145/1229375.1229379) [[GS Search](#)].

Fowler, M. UML Essencial: Um breve guia para a linguagem-padrão de modelagem de objetos, São Paulo, Ed. S.A, 2004. 155 p.

Gérard, S. Papyrus User Guide Series: About UML profiling. 2011. Disponível em: https://www.eclipse.org/papyrus/resources/PapyrusUserGuideSeries_AboutUMLProfile_v1_0.0_d20120606.pdf

Github. Disponível em: <https://github.com/>. Acessado em: Maio/2017.

Jardim, R. S. Jogo Virtual de Reeducação Alimentar Infantil. 78 f. Monografia (Bacharelado em Sistemas de Informação) - Universidade Federal do Estado do Rio de Janeiro - Centro de Ciências Exatas e Tecnologia, Rio de Janeiro. 2013. [[GS Search](#)].

Jácome, T. F. FISICARE - Um Software Educacional Para Auxílio à Aprendizagem em Física (Temperatura e Calor). 45 f. Relatório de Estágio (Sistemas de Informação) - Centro Universitário Luterano de Palmas, Palmas. 2003.

Kohwalter, T. C. *et al.* SDM - An Educational Game for Software Engineering. X SBGames - Salvador - BA, November 7th - 9th, 2011. doi: [10.1109/SBGAMES.2011.10](https://doi.org/10.1109/SBGAMES.2011.10) [[GS Search](#)].

Lahoud, Y. M. S. *et al.* Jogo Educativo para a Disciplina de Geografia. Colloquium Exactarum, v. 3, n. 2, p. 133-145. 2011. [[GS Search](#)].

Lessa Filho, C. A. *et al.* Um Jogo Educativo na Web no Contexto do Ensino Fundamental. CINTED - Novas Tecnologias na Educação, V. 12 Nº 2, dezembro, 2014. doi: [10.22456/1679-1916.53505](https://doi.org/10.22456/1679-1916.53505) [[GS Search](#)].

Li, Z. *et al.* Object-Oriented Sokoban Solver: A Serious Game Project for OOAD and AI Education. Proceedings of the 44th Annual Frontiers in Education Conference (FIE 2014), pp. 1-4, Madrid, Spain, October 22-25, 2014. doi: [10.1109/FIE.2014.7044115](https://doi.org/10.1109/FIE.2014.7044115) [[GS Search](#)].

Lisboa Filho, J. *et al.* (2010). A UML profile for conceptual modeling in GIS domain. DE@CAISE'2010, p.18-31. [[GS Search](#)].

Lobo, A. S. e Mello, L. S. V. Desenvolvimento de um jogo para apoio ao ensino de aritmética. 53 f. Monografia (Bacharel em Ciência e Tecnologia) - Instituto Federal de Educação, Campus Guarulhos. 2012.

Matunobu, Y. Desenvolvimento de um Software Educativo para o Treinamento em Percepção Musical. 81 f. Monografia (Bacharelado em Ciência da Computação) - Centro Universitário Eurípides De Marília - UNIVEM, Marília. 2010.

Mendes Júnior, A. Z. ABC - Sistema de Auxílio à Educação Infantil. 2012. Monografia - IFP - Campus Paranaguá. 2012.

Monsterismus. “An educational Android game to help teach the fundamentals of programming”. Disponível em: <https://github.com/Tape/Monsterismus>. Acessado em: Maio/2017.

Mwangi, Rose W. *et al.* Integrating ICT With Education: Designing an Educational Computer Game for Teaching Functions in Undergraduate Mathematics. Journal of Theoretical and Applied Information Technology. 2011. [[GS Search](#)].



Object Constraint Language (OCL). Disponível: <http://www.omg.org/spec/OCL/>. Acessado em: Maio/2017.

Object Management Group (OMG). Meta Object Facility (MOF) Core Specification. Disponível: <http://www.omg.org/spec/MOF/2.4.2/PDF>. Documento: formal/2014-04-03.

Object Management Group (OMG). Unified Modeling Language™ (UML®). Disponível: <http://www.omg.org/spec/UML/>. Documento: formal/2015-06.

Oliveira, L. R. “UP4EG (UML PROFILE FOR EDUCATIONAL GAMES) - Um Perfil UML para Modelagem de Jogos Educacionais Digitais”. Monografia de Graduação. Universidade Federal de Goiás/Regional Jataí. Jataí/GO. 2016.

Paschoal, Luan *et al.* JOE: Jogo Ortográfico Educacional. III Congresso Brasileiro de Informática na Educação (CBIE 2014). XXV Simpósio Brasileiro de Informática na Educação (SBIE 2014). doi: [10.5753/cbie.sbie.2014.652](https://doi.org/10.5753/cbie.sbie.2014.652) [GS Search].

Rodrigues, H. F. Aplicando Sistemas Hapticos em Serious Games: Um Jogo para a Educação em Higiene Bucal. 156 f. Dissertação (Mestrado em Informática) - Departamento de Informática - Universidade Federal da Paraíba, João Pessoa. 2011. [GS Search].

Silva, D. O. SIEP: Sistema Interativo Educacional Pedagógico. Monografia. Faculdade Cenecista de Capivari. São Paulo. 2012.

Sommerville, I. Engenharia de Software. 9º Edição. São Paulo: Pearson Prentice Hall, 2011. 529 p.

Sourceforge. Disponível: <https://sourceforge.net/>. Acessado em: Maio/2017.

Sousa, V. F. ITest Learning: Um jogo para o apoio ao ensino de testes de software. Monografia (Bacharel em Sistemas de Informação) - Universidade Federal do Ceará, Campus Quixadá. 2012. [GS Search].

Teramoto, E. H. I. *et al.* TOTH: Jogo Eletrônico para a Aprendizagem da Matemática. 80 f. Monografia (Engenharia da Computação) - Universidade de São Paulo - Escola Politécnica, São Paulo. 2008. [GS Search].

Torres, A. *et al.* Towards a UML profile for model-driven object-relational mapping. SBES. 2009. doi: [10.1109/SBES.2009.22](https://doi.org/10.1109/SBES.2009.22) [GS Search].

Tresohlavy, K. Educational Game: Tricky Trig. Object-Oriented Theory and Programming I. 2007.

Uetanabara Júnior, J. *et al.* An Overview and an Empirical Evaluation of UML-AOF. ACM SAC 2010. doi: [10.1145/1774088.1774564](https://doi.org/10.1145/1774088.1774564) [GS Search].

Vilela, J. F. F. Projeto e implementação de um Software Educativo Multi-Agente com Tropos e JADE. 129 f. Monografia (Engenharia da Computação) - Universidade Federal do Vale do São Francisco - UNIVASF, Juazeiro - Bahia. 2011.

Zagal, José P. *et al.* Towards an Ontological Language for Game Analysis. 2005. Disponível em: <http://www.gameontology.com/>.