

BROAD-PLG: Modelo Computacional para Construção de Jogos Educacionais

Gevã Martins, Fernanda Campos, Regina Braga, José Maria N. David

Universidade Federal de Juiz de Fora – Programa de Pós-graduação em Ciência da Computação – Núcleo de Pesquisa em Engenharia do Conhecimento
Juiz de Fora – MG – Brasil

geva.martins@ice.ufjf.br, fernanda.campos@ufjf.edu.br,
regina.braga@ice.ufjf.br, jose.david@ufjf.edu.br

Abstract. *The computational model BROAD-PLG, based on Software Products Line (SPL) approach, was developed to assist the development of educational games. The model consists of a high-level architecture, domain modeling based on features that describe three different faces of educational games and an application engineering tool that allows instantiation of a framework ready to be used in game development. The separation of concerns splits the domain into sets of features that include educational aspects, game mechanics and gamification elements. To evaluate the proposal a prototype was developed as well as an application of the gamification model.*

Resumo. *O modelo computacional BROAD-PLG, baseado na abordagem de Linha de Produtos de Software (LPS), tem o objetivo de auxiliar no desenvolvimento de jogos educacionais. O modelo é composto por uma arquitetura de alto nível, modelagens de domínio com características de três tipos dos jogos educacionais, e uma ferramenta de engenharia de aplicação que permite instanciação de um arcabouço para o desenvolvimento desses tipos de jogos. A separação de interesses divide o domínio em características que englobam aspectos educacionais, mecânica de jogos e elementos de gamificação. Para avaliar a proposta, foi desenvolvido um protótipo e construída uma aplicação demonstrando o modelo de gamificação.*

1. Introdução

Educadores de todos os níveis enfrentam um grande desafio na tentativa de capturar a atenção e de fomentar o interesse do aluno para os conteúdos educacionais que necessitam ser ensinados (Binotto, 2012). Os objetos de aprendizagem podem atender a essas demandas especialmente quando implementados na forma de jogos educacionais. Desenvolver jogos de qualidade é uma tarefa complexa e difícil. Nos jogos educacionais essa complexidade é evidenciada pela multidisciplinaridade envolvida. Qualidade está frequentemente relacionada a restrições, por exemplo, de tempo, custo, segurança, disponibilidade, portabilidade, acessibilidade e escalabilidade (Bittencourt et al., 2012).

A abordagem de Linha de Produtos de Software (LPS) (Clements; Northrop, 2001) tem sido proposta na literatura para apoiar a construção de software, com o objetivo de promover a qualidade, a produtividade e, ao mesmo tempo, reduzir os custos. Neste contexto, objetos de aprendizagem podem ser desenvolvidos e reutilizados considerando-se as características que são comuns e aquelas que são variáveis. Uma LPS é uma família de sistemas que compartilham uma determinada quantidade de

características, artefatos e um domínio (Tan, et al. 2012).

O termo gamificação vem sendo apresentado como uma alternativa para o engajamento dos alunos. Gamificação pode ser definida como a utilização das mecânicas de jogos em diferentes contextos. A abordagem também atua como uma catalisadora dos processos de atenção, promove um aumento das oportunidades de interação social, melhora as habilidades técnicas relacionadas às atividades exercitadas, permite o estabelecimento de objetivos de curto e longo prazo, incentiva o trabalho em equipe, proporciona uma percepção de progresso e, por fim, fornece um feedback rápido, frequente e preciso (Thomas et al, 2013).

Nesse artigo, apresentamos o BROAD-PLG uma infraestrutura computacional, baseada em uma Linha de Produtos de Software, para apoiar a construção de objetos de aprendizagem do tipo jogos educacionais ou com características de gamificação, foco de desse artigo.

Este trabalho está organizado da seguinte forma: na seção 2 são apresentados alguns conceitos sobre gamificação. Na seção 3 são apresentados os trabalhos relacionados. Na seção 4 é apresentado o projeto BROAD-PLG, as *features* que compõem a linha de produtos, a proposta de arquitetura e a avaliação. Ao final, são apresentadas as considerações finais e trabalhos futuros.

2. Gamificação

Kapp (2012) define gamificação como “... o uso de mecânicas, estética e modo de pensar dos jogos para engajar pessoas, motivar ações, promover aprendizado e resolver problemas.”. O processo de gamificação é proposto como uma forma de capturar esses fatores e utilizá-los em um contexto diferente.

O ponto em comum entre os conceitos de jogos e gamificação é o objetivo de engajar usuários na realização de determinadas atividades. Um jogo possui características como tempo finito e declaração de vencedores e perdedores. Gamificação por outro lado trás elementos utilizados em jogos para um processo pertencente a um outro domínio e não necessariamente transforma esse processo em um jogo. Para a Ciência da Computação jogos são importantes por apresentarem desafios tecnológicos como a necessidade de ambientes distribuídos, comunicação em tempo real, gráficos em estado da arte e realidade virtual.

Enquanto jogos possuem a habilidade de motivar os jogadores, Gamificação propõe ir além da motivação e alcançar o engajamento. O engajamento é uma relação de afetividade e comprometimento com algo. Enquanto um aluno pode estar motivado a aprender um determinado tópico através de um Jogo Educacional, são as propriedades desenvolvidas através da Gamificação que realizam a captura da atenção no tempo livre e podem fazer com que o aluno compartilhe suas atividades e interesses com os amigos e nas suas redes sociais.

3. Trabalhos Relacionados

Nesta seção são apresentados trabalhos relacionados à construção de jogos e artefatos de software educacionais utilizando-se a abordagem de Linha de Produtos de Software. A linha de produtos Arcade Game Maker (AGM) é um exemplo de LPS de jogos, criada pelo SEI (Software Engineering Institute) (2014). O método de desenvolvimento

incremental é sugerido e consiste na criação dos ativos que são necessários e suficientes para a instanciação do primeiro produto. A cada novo produto novos ativos são criados e adaptados. O grau de reusabilidade vai crescendo ao longo do ciclo de vida da LPS. A documentação do AGM representa a variabilidade da linha através de um diagrama de *features*. Em (Dalmon, 2012) é proposta uma linha de produtos para facilitar o desenvolvimento de Módulos de Aprendizagem Interativos (iMA). Esses aplicativos educacionais foram, originalmente, concebidos para serem construídos através de ferramentas de autoria e utilizados junto com Sistemas Gerenciadores de Curso (SGC). Possuem a particularidade de oferecerem as funcionalidades de avaliação automática e de retroatividade (feedback). Em (Silva et al., 2012; Silva, 2011) é proposta uma linha de produtos de software para Sistemas de Tutores Inteligentes (STI). Um STI típico tem que modelar três aspectos básicos: Domínio, Aprendiz e Estratégias Pedagógicas. A arquitetura da linha é baseada no processo de desenvolvimento de software baseado em componentes UML. A arquitetura proposta por este processo constiu-se de quatro camadas: Interface de Usuário, Sessão do Usuário, Serviços de Sistema e Serviços de Negócio. Para a fase de Engenharia de Aplicação o trabalho propõe uma ferramenta de autoria. Chimalakonda et al. (2013) descrevem a utilização de técnicas de LPS para a criação massiva e customizada de aplicativos de ensino. Os aplicativos atendem ao problema da alfabetização de adultos na Índia. A solução proposta pelos autores tem o nome de Adult Literacy Programme (ALP). A fase de Engenharia de Aplicação é baseada em uma ferramenta de autoria e apresenta um mapeamento restrito das variabilidades. Zhang et al. (2005) utilizam a abordagem de LPS para a geração de jogos do gênero RPG para aparelhos móveis. Através da experiência dos especialistas de negócio os pontos em comum e variabilidades foram mapeados e representados em um diagrama de *features*. Com o mapeamento das características (*features*) é realizada uma divisão em componentes utilizando uma técnica de meta-programação que cria componentes genéricos parametrizáveis. Com os componentes definidos o código é otimizado antes de ser reutilizado. Na fase de Engenharia de Aplicação as meta-variáveis são substituídas com o código das *features* selecionadas. Como resultado final, os autores afirmam haver reduzido o tempo de desenvolvimento de um novo jogo. Em (Cho et al, 2008) é proposta uma LPS para jogos móveis com a adição de padrões de arquitetura. Segundo os autores o domínio de aplicações para dispositivos móveis conta com restrições de interface, de capacidade de memória e do poder de processamento, entretando, a LPS é uma solução para atender as diversas características. Outra dificuldade para a adoção da LPS no desenvolvimento móvel é o rápido ciclo de desenvolvimento. Os autores sugerem que para ampliar a adoção de LPS para plataformas móveis é necessária a criação de padrões de arquitetura.

O presente trabalho se destaca dos demais em relação aos seguintes aspectos: i) A necessidade da Análise de Domínio que culminou na separação de assuntos em três interesses distintos: jogos, jogos educacionais e gamificação. Cada interesse tem seu próprio modelo de *features*, com uma relação sinérgica entre eles. ii) Ao fato de a linha permitir a criação de até oito tipos de aplicação (Tabela 1), cujos modelos são independentes e isso se reflete na fase de Engenharia de Aplicação. iii) A arquitetura da linha proposta não define nenhuma tecnologia específica ou restrições não funcionais. A arquitetura é composta por quatro camadas sem restrições para implementação. iv) A LPS proposta tem possibilidade de aproveitar motores de jogos produzidos por terceiros.

4. BROAD-PLG: Modelo Computacional para Construção de Jogos Educacionais

O projeto BROAD utiliza conceitos de LPS para viabilizar o reuso sistemático de OAs . Nesse contexto, o BROAD-PL (Product Line), permite a personalização dos objetos de aprendizagem, através de uma arquitetura apoiada semanticamente em uma rede de ontologias. A ferramenta Easy Tutorial E-learning (EasyT) foi um primeiro passo para a construção de objetos de aprendizagem via LPS. A infraestrutura BROAD-PLG, uma extensão da arquitetura BROAD-PL, propõe um modelo computacional para a construção de objetos de aprendizagem do tipo jogos educacionais.

A abordagem de LPS foi escolhida como facilitadora para a produção de Jogos Educacionais por três motivos: possuir uma abundante literatura que comprova através de estudos a eficácia da abordagem mesmo sobre domínios complexos, Jogos Educacionais constituem um domínio cujos elementos possuem evidentes características em comum que podem ser reutilizadas de maneira sistemática através da LPS e a LPS permite a criação de arquiteturas de alto nível que comportam implementações para uma grande variedade de ambientes de desenvolvimento e execução.

4.1 Modelo de *Features*

As *features* da Linha de Produtos da proposta BROAD-PLG foram divididas em três modelos distintos. Cada modelo mapeia um aspecto específico da linha: características de jogos, características educacionais e características de gamificação. A vantagem de se dividir o modelo de *features* em três subdomínios é a possibilidade de se obter diferentes tipos de produtos de forma independente, coesa e de baixo acoplamento. A Tabela 1 apresenta as possíveis categorias de produtos, tratadas dessa proposta.

Tabela 1. Categorias de produtos na seleção de *features*.

Jogos	Educacional	Gamificação	Produto
			O produto gerado é um arcabouço com as características do motor de jogos e/ou da plataforma selecionadas.
X			O produto final é um jogo.
	X		O produto final é um objeto de aprendizagem.
		X	O produto final é um sistema que implementa o processo de gamificação.
X	X		O produto final é um jogo educacional.
X		X	O produto final é um jogo que contém elementos motivacionais baseados em jogos.
	X	X	Um objeto educacional com elementos de engajamento.
X	X	X	O produto final é um jogo educacional com elementos especialmente projetados para se obter engajamento.

Não é pretensão realizar a caracterização definitiva dos aspectos de um jogo, jogo educacional ou processo de gamificação. O foco do modelo computacional gerado são os jogos educacionais e objetos educacionais com características de gamificação, e não a geração automatizada de quaisquer tipos de objetos de aprendizagem. O mapeamento das características de jogos teve como princípio a busca por mecânicas de alto nível e independência de hardware, plataforma e protocolos. São eles: Gênero, Arquétipo, Número de Jogadores, Seleção de Dificuldade, Regras, Condição de Vitória, Vitória por Pontos, Número de tentativas, Ambiente virtual, Objetivo, Tempo e Controle. As *features* das características educacionais, destacam os Metadados

(Giunchiglia et al, 2013) e as características: Avaliação, Exercício, Conteúdo e Referências Externas. As características de gamificação tem como objetivos comuns a fomentação da motivação através de mecanismos práticos e a promoção do engajamento. A Figura 1 mostra o diagrama de *features* de gamificação.

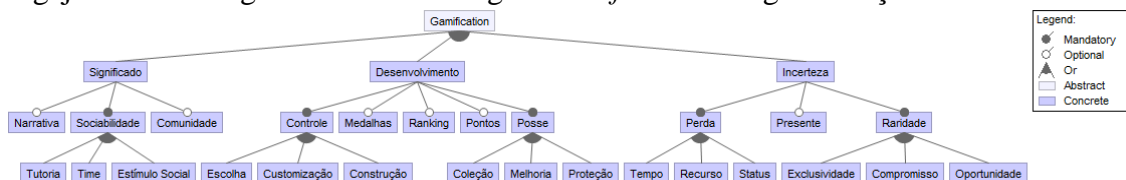


Figura 1: Diagrama de *Features* de Gamificação.

- Significado:** consiste em meios de explicitar os propósitos que dirigem as tarefas que o usuário está desempenhando. Uma das formas de tornar o significado explícito é através da característica Narrativa, que consiste em elaborar uma justificativa envolvente que forneça um motivo pelo qual as tarefas estão sendo desempenhadas. Opcionalmente, pode-se selecionar a característica Comunidade. Essa característica envolve ações em favor de uma determinada sociedade ou ambiente no qual o usuário está inserido. Sociabilidade provê mecanismos que promovem a motivação através do contato entre pessoas. A característica Tutoria fornece mecanismos para que um usuário mais experiente possa acompanhar, ensinar e aconselhar um usuário menos experiente. A característica Time coloca usuários trabalhando juntos em um mesmo objetivo. O Estímulo Social representa a apreciação de uma determinada comunidade pelo trabalho do usuário. É uma característica recorrente das redes sociais implementar uma funcionalidade que permita demonstrar essa apreciação na forma de “Likes”, “+1” e “Gostei”. Permitir a expressão da depreciação, no entanto, pode gerar, de acordo com (Cheng et al, 2014), um ciclo vicioso de represálias e apatia.
- Desenvolvimento:** À medida que esforço e dedicação vão sendo colocados pelo usuário nas atividades que completa, a característica Desenvolvimento provê o feedback necessário para motivar o usuário a continuar persistindo nos seus esforços. A forma mais comum de demonstrar o desenvolvimento do jogador é através da distribuição de pontos. O jogador consegue medir seu progresso pela velocidade da conquista de Pontos. A distribuição de Medalhas (Badges) serve para guiar o jogador em direção à realização de tarefas que o projetista deseja que o usuário realize. Distribuindo-se pontos e medalhas é possível organizar os usuários em Rankings, estimulando a competição. O Controle é, segundo (Pink, 2011), um dos três principais fatores de motivação para o ser humano junto com Maestria, o domínio técnico em estado da arte de uma atividade, que é satisfeito pela característica Desenvolvimento, e Propósito, que é satisfeito pela característica Significado. O controle é passado para o jogador através da possibilidade de realizar escolhas que tenham significado, de personalizar os artefatos que estiver controlando ou de modificar o ambiente através da construção de novos artefatos. Posse é a característica que consegue despertar no usuário o sentimento de preocupação com relação ao status de elementos que estejam presentes apenas no sistema. O despertar desse sentimento de preocupação pode ser feito através da implementação de mecanismos de Coleção, de Melhoria de um determinado elemento ou do dever de Proteger e cuidar de algum elemento do jogo.
- Incerteza:** É a característica que lida com eventos inesperados, tanto os positivos quanto os negativos. Uma das formas de se implementar Incerteza é através de uma recompensa surpresa. Ao realizar uma tarefa, o jogador ganha uma recompensa aleatória (Presente). Como a recompensa pode não ser o que o jogador espera, ele refaz a tarefa na esperança de receber a recompensa específica que deseja. Aversão à Perda é um viés humano que exerce pródiga influência sobre as decisões do cotidiano (Neumann et al, 2014). A característica lida com esse viés de modo a induzir o comportamento do jogador. Mecanismos comuns incluem a ameaça de perda de tempo, recursos e status. Raridade utiliza o apelo ao que é exclusivo, único ou difícil de conseguir. O valor que um objeto tem está ligado à demanda e à oferta. Quanto menor a quantidade de um item maior o seu valor e quanto maior o seu valor maior a demanda por aquele item (Ariely, 2009). Colocar restrições para a conquista de algo faz com que o jogador tenha vontade de obtê-lo.

Com o objetivo de tratar as características previamente apresentadas, uma arquitetura foi projetada no contexto de gamificação.

4.2 Arquitetura BROAD-PLG

Considerando-se a multiplicidade de plataformas, *frameworks*, *engines* e linguagens de desenvolvimento a opção escolhida foi a de desenvolver uma arquitetura de alto nível que se abstenha de selecionar protocolos ou padrões de tecnologia. Ao realizar a abstenção de restrições técnicas a arquitetura permite, por exemplo, que a LPS seja especializada para trabalhar na plataforma web utilizando-se de serviços ou em uma plataforma *mobile* transmitindo dados através de múltiplos protocolos e múltiplas redes. A Figura 2 mostra a visão geral da arquitetura:

- **Client Tier:** representa os meios disponíveis para acesso ao BROAD-PLG. Na primeira versão o acesso é implementado para funcionar sobre a web.
- **Middle Tier:** representa os serviços do sistema e de negócio. É composta por Domain Tier, responsável por providenciar um meio do usuário selecionar a configuração do produto através das *features* disponíveis na linha. O usuário pode compor a configuração do produto através da seleção de características dos três modelos: jogos, educacional e gamificação. Após a seleção das características, uma representação do modelo composto pelas *features* escolhidas pode ser enviada para a Application Tier que recebe a configuração enviada pela Client Tier e o repassa para o Factory Connector que, por sua vez, utilizando os motores de jogos e de plataforma, constrói o arcabouço para a aplicação. O Factory Connector é um módulo que consegue se comunicar com os dois tipos de motores da linha, ler a configuração enviada pela Client Tier, e ainda recuperar ativos da Backend Tier. O Factory Connector tem a missão de automatizar ao máximo a produção de um novo produto.
- **Backend Tier:** armazena os ativos da linha e a base de conhecimento. São separados os repositórios que armazenam arquivos de projeto e de multimídia para facilitar que o Factory Connector consiga recuperar os arquivos de multimídia de forma automática.

4.3 Modelo Computacional

Para avaliar a proposta foi implementado um protótipo da arquitetura contendo uma camada cliente que permite a seleção de *features* de jogos, educacionais e de gamificação. A camada cliente foi construída em cima da plataforma web com a linguagem de programação *Processing* que se utiliza da JVM (*Java Virtual Machine*) para compilar e executar o código. Após a seleção das *features* desejadas o sistema exporta a seleção em um arquivo XML. A implementação da camada do meio é feita através de um aplicativo Java que lê o arquivo XML, valida e instancia o arcabouço. Foi desenvolvido um motor de plataforma para HTML5 que está dentro da aplicação que instancia o arcabouço.

Só podem ser selecionadas características do tipo OU, opcionais ou alternativas. Somente pode ser selecionada uma opção alternativa de cada pai. Se um filho de uma característica do tipo OU ou alternativa é selecionada então o pai é automaticamente selecionado se não for uma característica obrigatória. Se o irmão de uma característica alternativa é selecionado então o outro irmão previamente selecionado é desselecionado. Se o pai que possui filhos com características do tipo OU ou alternativa é desselecionado então os filhos são automaticamente desselecionados. Após a seleção de uma árvore de características o usuário pode passar para as próximas árvores selecionando os ícones nos campos superiores esquerdo e direito. Ao terminar com a seleção nas árvores de características o usuário deve pressionar o botão “Gerar Configuração do Produto” no canto inferior direito. Após o pressionamento do botão o sistema gera o XML com a configuração do produto (Figura 3).

A camada de aplicação foi implementada como uma aplicação Java CLI que recebe como parâmetro o arquivo de configuração e instancia o arcabouço de acordo com o motor de plataforma e com o motor de jogos caso seja especificado.

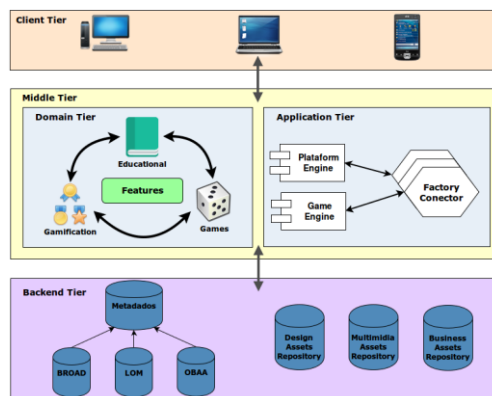


Figura 2: Arquitetura do BROAD-PLG.

4.4 Avaliação da Proposta

A avaliação da proposta da arquitetura BROAD-PLG se deu por meio do desenvolvimento do protótipo e da geração de produtos. Após a construção do protótipo, foram planejadas as provas de conceito seguindo modelos que contemplassem a avaliação da viabilidade da proposta. O produto, descrito a seguir, ressalta as características de gamificação. Para avaliação do protótipo foi desenvolvido o produto Game Work, demonstrando a versatilidade da linha e a sinergia entre os três domínios. O produto desenvolvido é um sistema de controle de atividades gamificado de uma empresa de fabricação de software, para ser utilizado em disciplinas da área de Engenharia de Software. A Figura 3 mostra a seleção de características de gamificação do produto.

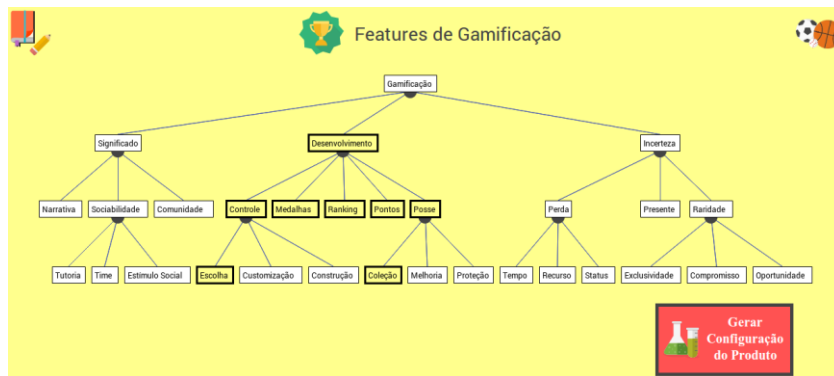


Figura 3: Seleção das características de gamificação para o produto *Game Work*.

A aplicação se concentra em características de desenvolvimento de software. A cada tarefa desempenhada o funcionário/aluno recebe pontos pela conclusão dependendo da dificuldade de cada tarefa. Se a tarefa é concluída dentro do prazo, o funcionário/aluno recebe pontos integrais e, se a tarefa for entregue fora do prazo, ele recebe menos pontos. Pontos extras também são concedidos caso as tarefas sejam entregues subsequentemente, dentro do prazo, sem que o funcionário/aluno quebre a corrente de entregas dentro do prazo. Como exemplo, foi gerada uma simulação de uma empresa-escola com 18 funcionários/alunos. A Figura 4 mostra o quadro com a classificação dos funcionários/alunos.

A concessão de medalhas foi outra característica selecionada para o produto, a cargo gerente de cada equipe. Na implementação foram definidos nove tipos de

medalhas: Funcionário do Mês, Excelência, Crescimento, Iniciativa, Inovação, Sempre Alerta, Liderança, Além do Dever e Trabalho em Equipe. A Figura 5 mostra um quadro individual de medalhas de um funcionário/aluno que conseguiu coletar todas as medalhas possíveis.

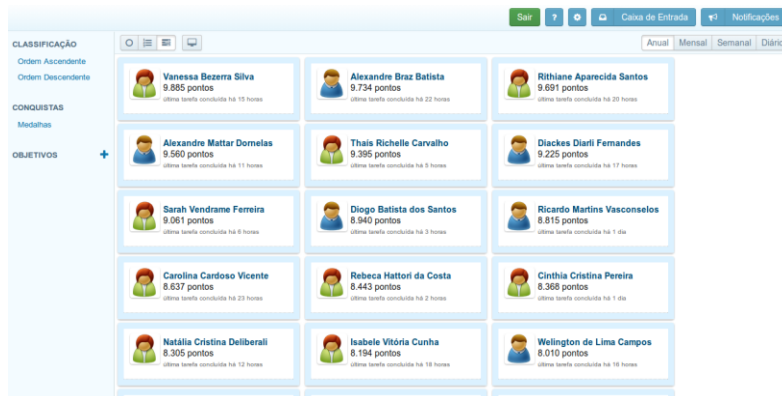


Figura 4: Classificação geral dos funcionários/alunos.



Figura 5: Quadro de medalhas de um funcionário/aluno que possui todas as medalhas.

Outra característica selecionada e implementada na aplicação Game Work é a capacidade de escolha. A implementação possibilita o funcionário/aluno escolher quais tarefas deseja executar. A Figura 6 mostra a tela de seleção de tarefas.



Figura 6: Tela de seleção de tarefas na visão do funcionário/aluno.

O interesse maior no desenvolvimento desse produto foi observar a viabilidade de utilização da solução proposta, e conjunto de *features* para a geração automática, ou semi automática, de jogos educacionais. Investigações futuras quanto à capacidade de evolução e adaptação às outras tecnologias diferentes da plataforma web poderão validar a viabilidade de uso da BROAD-PLG.

A experiência em desenvolver o produto nos mostrou que o principal esforço em se desenvolver aplicativos do tipo jogos, educacionais ou não, reside na engenharia de mídias e sua orquestração. O aplicativo Game Work se beneficiou com a geração do arcabouço: 50% dos arquivos do projeto foram gerados automaticamente. Como cada

jogo possui uma identidade visual diferente, para cada novo projeto novas mídias têm de ser acrescentadas. Apenas elementos menores, como pequenas imagens ou pequenos trechos de som, conseguem ser reutilizadas com maior facilidade por causa da sua natureza genérica e pouco específica.

5. Considerações Finais

O projeto BROAD-PL pretende especificar, projetar, implementar e avaliar uma linha de produtos de software para objetos de aprendizagem e com essa proposta, foi dado mais um passo para que seja alcançado esse objetivo. A identificação de *features* contribui para o desenvolvimento de um núcleo de artefatos reutilizáveis e que irão potencializar a construção de objetos de aprendizagem e depois reutilizá-los para composição de conteúdos contextualizados.

BROAD-PLG é um projeto em andamento, mas alguns passos importantes já foram finalizados como a especificação de *features* para jogos, jogos educacionais e gamificação, a definição de uma arquitetura que poderá compor a linha de produto de software e o desenvolvimento de um modelo computacional capaz de automatizar etapas para a geração de produtos do domínio. Os resultados alcançados com o uso da primeira versão mostraram-se promissores em relação à possibilidade de produção e reutilização em larga escala dos jogos educacionais, para as diversas áreas do conhecimento e sua personalização para diferentes perfis de usuários.

De maneira geral, a LPS conseguiu cumprir com a missão de auxiliar na construção de produtos de jogos, produtos educacionais e de produtos que implementem gamificação. A fase de Engenharia de Domínio conseguiu captar os elementos essenciais de cada domínio e a Engenharia de Aplicação conseguiu prover um mecanismo de alto nível com bastante capacidade de adaptação e potencial de geração de produtos de forma semi-automática. Um problema em aberto é como tratar a evolução do produto e do modelo. Na literatura existem sugestões sobre como abordar a evolução de um LPS, no entanto, não há consenso e certamente nem todas as soluções são adequadas para o BROAD-PLG. Encontrar a melhor solução para o acompanhamento da evolução da linha fornece um desafio para trabalhos futuros.

Referências

- ARIELY, D. Predictably Irrational, **Revised and Expanded Edition: The Hidden Forces That Shape Our Decisions**. HarperCollins, 2009.
- BINOTTO, S. & Basso, M. **Banco Internacional de Objetos Educacionais: um relato de experiência do Projeto ODIN**. International Bank of Educational Objects: experience report of ODIN Project Revista ACB, 2012, 17, 174-193.
- “**BROAD Project: Semantic Search and Application of Learning Objects**”, IEEE Technology and Engineering Education (ITEE), Dez., vol.7, n.3.
- CASTRO, J.; Nazar, J. M. & Campos, F. **EasyT: Apoiando a Construção de Objetos de Aprendizagem para uma Linha de Produtos de Software**. Conferencias LACLO, 2012, 3, 11.
- CHENG, J.; Niculescu-Mizil, C. D. & Leskovec, J. **How Community Feedback Shapes User Behavior**. CoRR, 2014, abs/1405.1429, 10.

- CHIMALAKONDA, S. & Nori, K. V. **What makes it hard to apply software product lines to educational technologies? Product Line Approaches in Software Engineering (PLEASE)**, 2013 4th International Workshop on, 2013, 17-20.
- CHO, H. & Yang, J.-S. **Architecture Patterns for Mobile Games Product Lines Advanced Communication Technology**, 2008. ICACT 2008. 10th International Conference on, 2008, 1, 118-122.
- CLEMENTS, P. & Northrop, L. Addison-Wesley (Ed.) **Software product lines Addison-Wesley**, 2002.
- DALMON, D. L. **Uma Linha de Produto de Software para Módulos de Aprendizagem Interativa**. Universidade de São Paulo, 2012.
- GIUNCHIGLIA, F.; Khandaker, T. H.; Sheikh Shaugat, A. & Rezwan, A. **Minimalist Metadata Visualization: The Minimal Set of Context Dependent Attributes for Entity Identification**. IEEE 17th International Conference on Intelligent Engineering Systems, Università degli Studi di Trento. Dipartimento di Ingegneria e Scienza dell'Informazione, 2013.
- KAPP, K. Taff, R. (Ed.) **The Gamificação of Learning and Instruction: Game-based Methods and Strategies for Training and Education**. Wiley, 2012.
- NEUMANN, N. & Böckenholt, U. **A Meta-analysis of Loss Aversion in Product Choice**. Journal of Retailing , 2014, 90, 182 – 197.
- PINK, D. **Drive: The Surprising Truth About What Motivates Us**. Penguin Group US, 2011.
- SEI. **Software Engineering Institute**. Disponível em: https://www.sei.cmu.edu/product_lines/ppl/index.html. Acesso em 15 de agosto de 2014.
- SILVA, L. et al. (2011) “**Composer-Science: A semantic service based framework for workflow composition in e-Science projects**”. Information Sciences, p. 186-208.
- SILVA, A.; Costa, E. & Bittencourt, I. **Uma Linha de Produto de Software baseada na Web Semântica para Sistemas Tutores Inteligentes**. Revista Brasileira de Informática na Educação, 2012, 20, 87.
- TAN, W. H.; Neill, S. & Johnston-Wilder, S. **How do professionals’ attitudes differ between what game-based learning could ideally achieve and what is usually achieved**. International Journal of Game-Based Learning (IJGBL), IGI Global, 2012, 2, 1-15.
- THOMAS, C. & Berkling, K. **Redesign of a gamified Software Engineering course Interactive Collaborative Learning (ICL)**, 2013 International Conference on, 2013, 778-786.
- ZHANG, W. & Jarzabek, S. **Reuse without compromising performance: industrial experience from RPG software product line for mobile devices**. Software Product Lines, Springer, 2005, 57-69.
- ZHU, X.-J., L. X.-F. W. G. (2007) “**Ontology based sharing and services in e-learning repository**”, Proceedings - 2007 IFIP International Conference on Network and Parallel Computing Workshops, NPC 2007, pp. 957–962.