

UNIVERSIDADE FEDERAL DE JUIZ DE FORA
INSTITUTO DE CIÊNCIAS EXATAS
PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

BROAD-PLG: Modelo Computacional para Construção de Jogos Educacionais

Gevã Schaefer Pereira Martins

JUIZ DE FORA, MG - BRASIL.
SETEMBRO DE 2014

BROAD-PLG: Modelo Computacional para Construção de Jogos Educacionais

Gevã Schaefer Pereira Martins

Universidade Federal de Juiz de Fora

Instituto de Ciências Exatas

Departamento da Ciência da Computação

Mestrado em Ciência da Computação

Orientadora: Fernanda Claudia Alves Campos, D. Sc.

JUIZ DE FORA, MG - BRASIL.

SETEMBRO DE 2014

BROAD-PLG: Modelo Computacional para Construção de Jogos Educacionais

Gevã Schaefer Pereira Martins

DISSERTAÇÃO SUBMETIDA AO CORPO DOCENTE DO INSTITUTO DE
CIÊNCIAS EXATAS DA UNIVERSIDADE FEDERAL DE JUIZ DE FORA, COMO
PARTE INTEGRANTE DOS REQUISITOS NECESSARIOS PARA A OBTENÇÃO
DO GRAU DE MESTRE EM CIÊNCIA DA COMPUTAÇÃO.

Aprovada por:

Profa. Fernanda Claudia Alves Campos, D. Sc.

Prof. José Maria Nazar David, D. Sc.

Profa. Vera Maria Benjamim Werneck, D. Sc.

JUIZ DE FORA, MG - BRASIL
SETEMBRO DE 2014

AGRADECIMENTOS

À Deus, pelas forças.

À professora Fernanda Claudia Alves Campos, por estar presente nas três fases do programa: na banca de seleção, nas disciplinas e na orientação.

Aos membros da banca, pelo tempo e atenção desprendidos.

Aos meus colegas de mestrado, pelos bons momentos.

Aos meus colegas de trabalho, pela compreensão.

À Universidade Federal de Juiz de Fora, pelos incentivos.

À minha família, pelo carinho.

“All schools will end up using game metrics in the future.”

- Nolan Bushnell.

RESUMO

As vantagens da utilização de jogos com objetivos educacionais podem ser consideradas como um consenso entre professores e alunos. No entanto, jogos sérios constituem-se objetos multimídia complexos e caros de se produzir. A natureza multidisciplinar dos jogos educacionais pressupõe o envolvimento e coordenação de uma equipe especializada. Com o objetivo de auxiliar no desenvolvimento de jogos educacionais é proposto o modelo computacional BROAD-PLG. O modelo é composto por uma arquitetura de alto nível, modelagens de domínio baseadas em características que descrevem três diferentes tipos dos jogos educacionais, e uma ferramenta de engenharia de aplicação, que permite instanciação de um arcabouço pronto para ser utilizado no desenvolvimento desses tipos de jogos. A separação de interesses divide o domínio de jogos educacionais em conjuntos de características que englobam aspectos educacionais, mecânica de jogos e elementos de gamificação. Ao final do trabalho são construídos três exemplos distintos de aplicações demonstrando esses três aspectos. Avaliando-se os exemplos pode-se concluir que o BROAD-PLG, apesar de estar na versão inicial, demonstra grande potencial para ser utilizado como uma ferramenta tanto na forma de geração de aplicações, quanto como referência na modelagem de domínio do problema.

Palavras-chave: Jogos. Jogos Educacionais. Jogos Sérios. Educação. Modelagem de domínio. Gamificação. LPS. Linha de Produtos de *Software*. Metadados. BROAD.

ABSTRACT

The advantages of using games for educational purposes can be considered as a consensus among teachers and students. However, serious games constitute complex and expensive to produce multimedia objects. The multidisciplinary nature of educational games requires the involvement and coordination of a specialized team. With the objective of assisting in the development of educational games is proposed the computational model BROAD-PLG. The model consists of a high-level architecture, domain modeling based on features that describe three different faces of educational games and application engineering tool that allows instantiation of a framework ready to be used in game development. The separation of concerns splits the domain of educational games into sets of features that include educational aspects of game mechanics and gamification elements. At the end of the work three different application examples are constructed demonstrating these three aspects. In reviewing the examples it can be concluded that the broad-PLG despite being in the initial stage shows great potential to be used as a tool in generating applications as reference in modeling the problem domain.

Keywords: Games. Educational game. Serious games. Education. Domain modeling. Gamification. SPL. Software Products Line. Metadata, BROAD.

LISTA DE TABELAS

Tabela 2.1 - Casos de sucesso de implementação de LPS	17
Tabela 2.2 – Definições de <i>features</i>	20
Tabela 2.3 – Comparação entre elementos na definição de jogos. Adaptado de (Salen et al, 2004).	23
Tabela 3.1 – Comparação de características entre os trabalhos relacionados	50
Tabela 4.1 – Possíveis combinações e seus produtos na seleção de features.	54

LISTA DE FIGURAS

Figura 2.1: Comparação visual entre <i>Geography Quest</i> e GTA.	26
Figura 2.3: Visão parcial dos metadados do BROAD. Adaptado de (Souza et al, 2010).	34
Figura 3.1: Diagrama de Casos de Uso da Arcade Game Maler	39
Figura 3.2: Diagrama de Features da Arcade Game Maker. Adaptado de SEI (2009).	40
Figura 3.3: Diagrama de <i>Features</i> de iMA (Dalmon, 2012).	41
Figura 3.4: Diagrama de componentes do arcabouço da iMA (Dalmon, 2012).	42
Figura 3.5: Tela de execução do iTangram	42
Figura 3.6: Diagrama de Features da LPS de STI. Adaptado de (Silva et al ,2012).	44
Figura 3.7: Mapeamento de um processo de produção (Chimalakonda, 2010)	46
Figura 3.8: Diagrama de Features de domínio de jogos móveis do gênero RPG.	47
Figura 3.9: Diagrama parcial de <i>features</i> do domínio de jogos móveis.	48
Figura 4.1: Visão Geral Arquitetura BROAD-PL (Campos et al., 2012).	52
Figura 4.2: Diagrama de Características de Jogos	55
Figura 4.3: Diagrama de Características Educacionais	58
Figura 4.4: Diagrama de Características de Gamificação.	59
Figura 4.5: Arquitetura do BROAD-PLG.	63
Figura 4.6: Execução da IDE da linguagem <i>Processing</i> .	65
Figura 4.7: Interface <i>web</i> para seleção de características de jogos.	65
Figura 4.8: Interface <i>web</i> para seleção de características educacionais.	66
Figura 4.9: Interface <i>web</i> para seleção de características de gamificação.	66
Figura 4.10: XML gerado com a configuração do produto.	66
Figura 4.11: Criação do arcabouço utilizando a camada do meio.	67
Figura 4.12: Seleção de características para o jogo <i>Goal Keeper</i> .	68
Figura 4.13: O jogo <i>Goal Keeper</i> em execução.	69
Figura 4.14: Término do jogo <i>Goal Keeper</i> .	69
Figura 4.15: Seleção de características de jogos do produto <i>Monty Cup</i> .	71
Figura 4.16: Seleção de características educacionais do produto <i>Monty Cup</i> .	71
Figura 4.17: Configuração inicial do produto <i>Monty Cup</i>	72
Figura 4.18: O jogador consegue marcar um gol no produto <i>Monty Cup</i> .	72
Figura 4.19: Seleção das características de gamificação para o produto <i>Game Work</i> .	73
Figura 4.20: Classificação geral dos funcionários.	73

Figura 4.21: Quadro de medalhas de um funcionário que possui todas as medalhas. 74

Figura 4.22: Tela de seleção de tarefas na visão do funcionário. 75

LISTA DE ABREVIACÕES

ALP	<i>Adult Literacy Programme</i>
DLC	<i>Downloadable Content</i>
DOM	<i>Data Object Model</i>
ESA	<i>Entertainment Software Association</i>
ESI	<i>European Software Institute</i>
FODA	<i>Feature Oriented Domain Analysis</i>
FPS	<i>First Person Shooter</i>
GTA	<i>Grand Theft Auto</i>
IA	<i>Inteligência Artificial</i>
iMA	<i>Módulo de Aprendizagem Inerativa</i>
JME	<i>Java Micro Edition</i>
JVM	<i>Java Virtual Machine</i>
KSI	<i>Korean Software Institute</i>
LPS	<i>Linha de Produto de Software</i>
MPAA	<i>Motion Picture Association of America</i>
MVC	<i>Model View Controller</i>
NLM	<i>National Literacy Mission</i>
NPC	<i>Non Playable Character</i>
OA	<i>Objeto de Aprendizagem</i>
RPG	<i>Role Playing Game</i>
SEI	<i>Software Engineering Institute</i>
SGC	<i>Sistemas Gerenciadores de Curso</i>
SO	<i>Sistema Operacional</i>
SPLC	<i>Software Product Line Conference</i>
STI	<i>Sistema de Tutor Inteligente</i>

SUMÁRIO

1	INTRODUÇÃO	13
1.1	Motivação	14
1.2	Metodologia	16
1.3	Objetivos	16
1.4	Organização	16
2	PRESSUPOSTOS TEÓRICOS	17
2.1	Linha de Produtos de Software	17
2.1.1	Engenharia de Domínio	20
2.1.2	Engenharia de Aplicação	22
2.2	Jogos e Jogos Educacionais	22
2.2.1	Jogos	23
2.2.2	Jogos Educacionais	24
2.2.3	Gamificação	27
2.2.4	Metadados	32
2.2.5	Considerações Finais do Capítulo	34
3	TRABALHOS RELACIONADOS	37
3.1	Arcade Game Maker	37
3.2	Linha de Produtos de Software para Módulos de Aprendizagem Interativa	40
3.3	Linha de Produtos de software web semântica para Tutores Inteligentes.....	43
3.4	Linha de ProdutoS de E-Learning	45
3.5	Linha de ProdutoS para Jogos Móveis	47
3.6	Considerações Finais Do Capítulo.....	49
4	BROAD-PLG: MODELO COMPUTACIONAL PARA CONSTRUÇÃO DE JOGOS EDUCACIONAIS	51
4.1	Introdução	51
4.2	BROAD-PLG – BROAD Product Line for Educational Games	54
4.2.1	Features.....	54
4.2.2	Arquitetura BROAD-PLG	61
4.2.3	Modelo Computacional	63
4.2.4	Avaliação da Proposta	67
4.2.5	Conclusões e lições Aprendidas	75
5	CONSIDERAÇÕES FINAIS.....	77
5.1	Limitações e Dificuldades	77
5.2	Trabalhos Futuros e Perspectivas	79

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 tópicos que necessitam ser ensinados (Binotto, 2012). Com a recente ascensão da computação móvel ocorreram mudanças no comportamento de estudantes de todos os níveis. O impacto dessas mudanças é sentido na sala de aula onde a explicação do professor compete pela atenção dos alunos com aplicativos digitais em smartphones e tablets. Essa intensa exposição gera no aluno uma expectativa de que o recurso educacional seja tão rico quanto os aplicativos que usam no dia-a-dia.

Os Grandes Desafios da Computação, definidos no II Seminário sobre Grandes Desafios da Computação no Brasil (SBC, 2009) envolveram discussões na área da Educação englobando, entre outros desafios, a necessidade de transformar os sistemas e-learning em ambientes mais atraentes e apropriados para diferentes perfis de usuários.

Nesse universo contemporâneo de informação e entretenimento ubiquamente disponíveis, onde as mídias ricas e interativas são amplamente utilizadas, torna-se interessante que o processo educacional também se adeque a essa realidade. Para tanto, é percebido que os objetos de aprendizagem digitais podem corresponder a essa demanda especialmente quando implementados na forma de jogos educacionais.

Jogos apresentam um problema que deve ser solucionado pelo jogador cujas ações são limitadas por um conjunto de regras (Saucé apud Chua, 2012). Ao tentar solucionar o problema o jogador é exposto a um aprendizado. Quando esse aprendizado é relacionado com um propósito educativo o jogo se caracteriza como um jogo educacional (Diaz, 2012).

Desenvolver jogos educacionais de qualidade, entretanto, é uma tarefa complexa e difícil. Embora qualidade seja um atributo relativo, ela está frequentemente relacionada a restrições, por exemplo, de tempo, custo, segurança, disponibilidade, portabilidade, acessibilidade e escalabilidade (Bittencourt et al., 2012). A complexidade aumenta quando o público alvo da aplicação é heterogêneo e possui diferentes contextos.

Em se tratando de Objetos de Aprendizagem (Oas) os requisitos funcionais e suas variabilidades demandam um conjunto de recursos necessários para seu desenvolvimento. 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, mantendo-se os padrões estabelecidos para o produto. Estas variabilidades são representadas através de modelos que poderão auxiliar, posteriormente, no projeto de novos artefatos. Para compor a LPS, são necessários mecanismos para gerenciar o núcleo de artefatos, bem como para apoiar o processo de reutilização envolvido no momento do desenvolvimento de novas aplicações (Weiss et al, 2009). Para tanto, torna-se necessário investigar soluções para o desenvolvimento e gerenciamento de um Núcleo de Artefatos capaz de apoiar o desenvolvimento de objetos de aprendizagem. A partir da definição dessas soluções novos objetos poderão ser desenvolvidos e reutilizados para a composição de conteúdos educacionais.

Para auxiliar e tratar as questões acima é proposta a utilização da abordagem de Linha de Produto de Software aplicada aos OAs. Uma LPS é uma família de sistemas que compartilham uma determinada quantidade de características, artefatos e um domínio (Tan, et al. 2012). A utilização de LPS permite a economia de tempo de desenvolvimento, controle da qualidade e tempo para o mercado (*time-to-market*), o que pode representar um diferencial na transformação dos ambientes educacionais.

O termo gamificação vem sendo apresentado como uma alternativa para o engajamento dos alunos (Thomas et al, 2013). Gamificação pode ser definida como a utilização das mecânicas de jogos em diferentes contextos como, por exemplo, o fazer acadêmico de uma sala de aula do ensino superior. As vantagens do processo de gamificação, no entanto vão além do aumento do engajamento. 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.

1.1 MOTIVAÇÃO

Segundo a ESA (Entertainment Software Association), associação que reúne empresas como a Microsoft, Sony e Warner Bros, a média de idade dos jogadores de jogos eletrônicos (*gamers*) é de 30 anos. A maioria dos *gamers* é maior de 18 anos (64%) e a diferença demográfica entre os gêneros não é grande, ficando os jogadores masculinos com 55% de presença no mercado (ESA, 2013). Nas plataformas *mobile* e em *tablets* a presença feminina é majoritária (61%) e a média de idade cai para 26 anos (EEDAR, 2013).

Os motivos para se jogar variam de acordo com cada pessoa. Algumas utilizam-se dos jogos como uma forma de diversão, escapismo, para treinar uma habilidade específica, para se socializar ou simplesmente passar o tempo. Nas plataformas *mobile* os motivos são: passar o tempo enquanto esperam, dar uma pausa em outras atividades e relaxar e se conectar com amigos e familiares (EEDAR, 2013).

A motivação para se desenvolver jogos encontra respaldo na significativa quantidade de receita arrecadada pelo mercado. Considerando apenas a indústria de jogos eletrônicos, sem contar com os jogos de mesa, a consultora Gartner (Gartner, 2013) estimou o faturamento de mercado em \$93 bilhões de dólares para 2013 e prevê um crescimento para \$111 bilhões de dólares em 2014. Comparando-se com a indústria de esportes, categoria que satisfaz a definição de jogos de (Salen et al, 2014), a consultora Pricewaterhouse Cooper (Forbes, 2013) prevê o alcance de um faturamento de \$67 bilhões de dólares apenas para 2017. A tradicional indústria do cinema obteve menos da metade do faturamento da indústria de jogos eletrônicos. A receita arrecadada com as exhibições das salas é de \$36 bilhões de dólares de acordo com o relatório anual da Motion Picture Association of America (MPAA) (MPAA, 2013).

Além dos fatores pedagógicos os jogos educacionais representam também uma significativa importância em termos de movimentação econômica. De acordo com a firma de pesquisa de mercado Ambient Insight, que tem como clientes empresas como Microsoft, Disney e Dell, o mercado de ensino baseado em jogos e simulações alcançou uma receita aproximada de \$4 bilhões de dólares em 2012 com uma previsão de faturamento de \$9 bilhões em 2017 (Adkins, 2013).

Os jogos, no entanto, possuem alguns riscos. Quando desprovidos de um objetivo produtivo podem se tornar um empecilho à realização de atividades profissionais e acadêmicas. O vício em jogos de azar é um problema bem conhecido e documentado. Com a chegada dos jogos online, a incidência entre os jovens aumentou devido à facilidade de acesso. O problema afeta a sociedade de diversos países e é alvo de discussões e legislações específicas (Kim et al, 2013). Outro aspecto negativo dos jogos, principalmente os eletrônicos, é a possibilidade da influência negativa no comportamento de crianças e jovens. A exposição precoce a jogos violentos está relacionada a casos de comportamentos anti-sociais variando de *bullying* a homicídios. A discussão levanta a hipótese de que crianças com problemas mentais pré-existentes sejam mais vulneráveis mas existem poucas evidências que suportem essa afirmação (Ferguson et al, 2013).

1.2 METODOLOGIA

A metodologia de pesquisa incluiu a revisão bibliográfica, a proposta de uma arquitetura capaz de gerar um arcabouço de um jogo educacional a partir da seleção das características pelo usuário, desenvolvimento da primeira versão e provas de conceito para avaliação da proposta.

Na revisão bibliográfica foram revistos os conceitos de linha de produtos de software, jogos, jogos educacionais, gamificação e metadados de objetos de aprendizagem.

Após essa caracterização foram identificadas e descritas as características de jogos, jogos educacionais e gamificação.

A proposta da arquitetura BROAD-PLG (BROAD Product Line for Educational Games) foi apresentada, a infraestrutura foi desenvolvida e foram realizadas provas de conceito, planejadas para gerar jogos educacionais.

1.3 OBJETIVOS

Desenvolvimento de uma infraestrutura computacional, baseada em uma linha de produtos de software, para apoiar a construção de objetos de aprendizagem do tipo jogos educacionais, com modelo de *features* de jogos, jogos educacionais e gamificação que irão integrar a sessão de escolha pelo usuário das características do jogo a ser desenvolvido.

1.4 ORGANIZAÇÃO

Para atingir o objetivo proposto, este trabalho está organizado da seguinte forma: este capítulo apresentou uma introdução, a motivação que orienta a busca da solução do problema a ser tratado, a metodologia, os objetivos e a organização da monografia. No capítulo 2 são apresentados os pressupostos teóricos. No capítulo 3 são apresentados os trabalhos relacionados, suas análises e comparações. No capítulo 4 é apresentado o projeto BROAD-PLG, as características que compõem a linha de produtos, a proposta de arquitetura e as provas de conceito. Ao final são apresentadas as considerações finais e as referências bibliográficas.

2 PRESSUPOSTOS TEÓRICOS

2.1 LINHA DE PRODUTOS DE SOFTWARE

Linha de Produtos de Software (LPS) “... é caracterizada como um conjunto de componentes que compartilham características comuns e são gerenciadas com o objetivo de satisfazer um determinado cenário de segmento de mercado ou missão.” (Castro et al, 2012). Essa abordagem busca, de forma gerenciada e sistemática, apoiar o desenvolvimento de novos produtos através do reaproveitamento de artefatos de software projetados para atender as especificidades de um determinado domínio (Scheidt et al, 2012). A esta definição também é acrescentado por (Mohalik et al, 2012) que os produtos da LPS são construídos de forma eficiente e econômica.

O desenvolvimento dos conceitos que lançaram as bases para a adoção dos processos de construção de sistemas através de Linhas de Produtos de Software, conhecidos mais comumente na Europa pelo termo famílias de software, data dos trabalhos seminais dos autores Dijkstra e Parnas respectivamente nas décadas de 60 e 70 (Reis, 2007). A SPLC (Software Product Line Conference), conferência anual sobre o tema mantém em seu site o Hall da Fama (<http://splc.net/fame.html>) que registra casos de sucesso da indústria. A tabela 2.1 mostra cinco desses casos com um resumo da implantação em cada empresa. Todas as empresas mostradas na tabela figuram entre as maiores do mundo de acordo com a revista americana Forbes (Forbes, 2014). A SPLC é patrocinada pela ESI (European Software Institute), SEI (Software Engineering Institute) e KSI (Korean Software Institute). Entre seus membros diretores estão autores consagrados da área de LPS como Linda Northrop, Klaus Pohl, Frank van der Linden e Paul Clements.

Tabela 2.1 - Casos de sucesso de implementação de LPS

Empresa	Linha de Produto de Software
Hewlett-Packard	A linha desenvolvida envolve a criação de uma cooperativa de desenvolvimento de <i>firmware</i> denominada Owen. Equipes sediadas em dois estados diferentes desenvolvem ativos para impressoras, copiadoras e <i>scanners</i> que seguem especificações para garantir a compatibilidade dos componentes.
Boeing	Denominada Bold Stroke a linha desenvolvida pela Boeing compreende uma ampla gama de ativos variando de ferramentas e artefatos de software a

	arquiteturas e processos. Os produtos desenvolvidos são sistemas de missão crítica, distribuídos em tempo real para controle das funções do avião e do <i>cockpit</i> dos pilotos.
Philips	O grupo Philips recebe os benefícios da utilização de LPS em vários segmentos como o desenvolvimento distribuído para software de máquinas médicas de aquisição de imagens como máquinas de raios-x, imagem de ressonância magnética, tomografia computadorizada e ultrassom.
Toshiba	A Toshiba é conhecida popularmente pelos equipamentos eletrônicos que produz, porém uma de suas áreas de atuação é o gerenciamento de usinas de força. A LPS que produz software customizado para mais de 150 usinas ao redor do mundo lida com a complexidade de dezenas de milhares de entradas de dados em média para cada usina.
Nokia	Antes de adotar o Windows Phone a Nokia produziu centenas de modelos de aparelhos celulares com os sistemas operacionais Nokia OS, Symbian OS e Symbian. Esses modelos foram desenvolvidos com o auxílio de uma LPS principalmente na parte de interface gráfica.

A abordagem de LPS permite a criação de sistemas de qualidade customizados em massa e, dentre suas vantagens, estão a redução do custo de desenvolvimento, a melhoria da qualidade dos produtos desenvolvidos e a redução do tempo para o mercado (Souza, 2014). A redução do custo de desenvolvimento é alcançada quando os ativos (*assets*) são reutilizados várias vezes nos produtos desenvolvidos. Normalmente o custo de desenvolvimento da linha se paga quando três ou mais produtos são criados (Clements et al, 2002). Quando os artefatos são reutilizados, testes são realizados, falhas corrigidas e o resultado é uma melhoria de qualidade geral dos produtos. No início da implementação da LPS o tempo de desenvolvimento é maior do que o tempo gasto em aplicações *ad-hoc* mas esse tempo é reduzido drasticamente conforme os produtos são desenvolvidos resultando em um menor tempo para o mercado (Ribeiro, 2012).

LPS não possui apenas benefícios. Iniciar uma Linha de Produto de Software em uma empresa ou organização não é uma tarefa simples. A LPS irá alterar radicalmente o modo tradicional pelo qual os sistemas eram feitos dentro da estrutura anterior. Existem, no entanto, métodos que permitem mitigar os riscos envolvidos com a implantação de uma LPS através do planejamento, previsão e priorização de características chaves desejadas pelos *stakeholders* como é o caso da aplicação matemática descrita em (Gillain et al, 2012). O método proposto tem como uma de suas metas o planejamento e a provisão do retorno esperado com a implantação de uma nova LPS realizada tanto a partir da fundação quanto através do reaproveitamento de ativos retirados de aplicações legadas.

Em uma aplicação tradicional, independentemente do processo de desenvolvimento adotado, deve-se primeiro reunir os requisitos a serem atendidos pelo sistema. Os requisitos são levantados pelo analista de sistemas ou desenvolvedor a partir das necessidades explicitamente levantadas pelos envolvidos com o processo de software (*stakeholders*). Na abordagem de desenvolvimento das Linha de Produtos de Software, no entanto, os requisitos não são pensados para atender a apenas um sistema mas a um domínio.

Um domínio pode ser definido sinteticamente como uma área de conhecimento. A LPS aborda um subconjunto dessa área de conhecimento. Esse subconjunto possui uma fronteira definida, mesmo que com bordas difusas, que é denominada escopo. Quanto maior o escopo de uma LPS maior a possibilidade de se gerar produtos, porém, menor a possibilidade de se reaproveitar os artefatos da linha. Quanto menor o escopo menos produtos possíveis, mas a possibilidade de reutilização dos ativos aumenta (Apel et al, 2013). O domínio possui uma representação realizada através de um modelo. Essa representação possui um nível de granularidade. A granularidade representa o nível de detalhe da representação do domínio. Quanto maior a granularidade mais detalhada é a representação do domínio, e mais fácil de se criar modelos que automatizem a criação de artefatos de software. Quanto maior a granularidade, no entanto, mais complexa e difícil se torna o gerenciamento da LPS. Uma granularidade menor significa uma redução no nível de detalhes da representação do domínio aumentando a simplicidade, facilidade de gerenciamento e liberdade na criação dos produtos. Por outro lado, uma granularidade menor, significa uma menor precisão ao se automatizar a construção dos produtos.

Ao se iniciar uma nova Linha de Produtos é possível escolher entre três formas de adoção: proativa, extrativa e reativa (Apel et al, 2013). Na adoção proativa os produtos a serem gerados são previamente conhecidos e servem como uma guia para todos os passos de projeto da linha e da criação de ativos. Com a adoção extrativa há uma refatoração dos produtos existentes que aos poucos vão sendo agregados na forma de uma Linha de Produtos. Seguindo a adoção reativa uma pequena Linha de Produtos é criada e aos poucos os ativos e os detalhes do domínio vão sendo acrescentados. Independente da forma de adoção escolhida o processo de implantação de uma LPS é composto por dois processos: A Engenharia de Domínio e a Engenharia de Aplicação. Em (Royer et al, 2012) esses processos também são denominados de Engenharia para Reuso e Engenharia com Reuso respectivamente. Os artefatos a serem reutilizados, denominados ativos (*assets*), não são necessariamente construtos de software mas podem ser diagramas, documentação, especificações de testes, especificações de requisitos, estimativas de custo e de esforço e, virtualmente, quaisquer

outras peças que façam referência à Linha de Produtos.

2.1.1 ENGENHARIA DE DOMÍNIO

A Engenharia de Domínio compreende a definição do escopo, a fronteira dentro da qual está contida a missão ou segmento de mercado atendido pela LPS, os requisitos a serem atendidos pelos produtos e as características (*features*) tanto as comuns, que estão presentes em todas as instâncias, quanto as variáveis, que caracterizam a individualidade e implementam a customização. Os artefatos reutilizáveis e específicos também são construídos durante a fase de Engenharia de Domínio.

A *feature* é uma característica chave da LPS e está presente tanto na fase de Engenharia de Domínio quanto na fase da Engenharia de Aplicação. A tabela 2.2 mostra algumas das definições descritas na literatura organizadas por ano de publicação (a coluna autor cita apenas aquele primeiramente creditado na obra).

Tabela 2.2 – Definições de *features*

Autor	Ano	Definição
Kyo Kang	1990	Um aspecto visível pelo usuário distinto ou proeminente, qualidade ou característica de um sistema ou sistemas de software.
Paul Clements	2000	Uma unidade lógica de comportamento especificada por um conjunto de requisitos funcionais e não funcionais.
Hassam Gomaa	2004	Um requisito ou característica que é providenciado por um ou mais membros da linha de produto.
Klaus Pohl	2005	Um requisito abstrato.
Sven Apel	2013	Característica ou comportamento visível para o usuário final de um sistema de software.

Para auxiliar na definição de escopo, na captura de requisitos e na elicitação de funcionalidades pode-se utilizar um método de análise de domínio, como, por exemplo, o FODA (Feature Oriented Domain Analysis), introduzido por Kyo Kang (1990) no trabalho seminal “*A Feature-oriented domain analysis (FODA) feasibility study*”. O método desenvolvido por Kang é composto por três fases: análise do contexto, modelagem do domínio e modelagem da arquitetura.

A análise de contexto delimita as fronteiras do domínio através da definição dos objetivos, padrões a serem seguidos e nomeação de entidades externas com a qual é relacionado. Se a LPS faz parte de um ecossistema pré-existente as características em comum e variantes dos sistemas existentes devem ser especificadas para se obter uma otimização na capacidade de reuso dos ativos. O cálculo dos riscos também é uma característica importante,

sendo portanto levantadas as informações sobre a quantidade de conhecimento disponível sobre o domínio, quais recursos podem ser levantados e quais recursos não possíveis de se obter (Kang et al, 1990).

Ainda segundo Kang (1990), a modelagem de domínio é sub-dividida em três processos: análise de características, modelagem entidade-relacionamento e análise funcional. A análise de características coloca em um modelo a visão geral das capacidades que o usuário final irá usufruir. O modelo que realiza essa visão é denominado modelo de característica (*feature model*). O modelo de características é comumente representado por um diagrama de características. A criação do modelo de características é iniciada a partir da reunião das fontes de informação levantadas a cerca do domínio. Sequencialmente as características são levantadas de acordo com quatro categorias: ambientes operacionais, capacidades, tecnologia e implementação técnica. A categoria “capacidades” ainda se sub-divide em características funcionais, características operacionais e características de apresentação. Definidas as características o processo tem prosseguimento com a modelagem de entidade-relacionamento. Assim como no modelo de características no modelo entidade-relacionamento não existem entidades isoladas. Todas as entidades se relacionam ou são relacionadas por outra entidade. Exemplos de relações comuns são: “é-um” e “consiste-de”. A última etapa da modelagem de domínio é a geração do modelo funcional. O modelo funcional descreve o comportamento do sistema de acordo com cada característica. No modelo funcional podem ser tomadas decisões sobre a implementação da variabilidade através da separação por componentes, utilização de parâmetros ou herança. De acordo com a notação do modelo é possível a utilização de outros paradigmas como a orientação a aspecto.

A última etapa do método FODA é a modelagem da arquitetura. Nessa fase é definida uma arquitetura de referência que será compartilhada pelos produtos da linha. A definição da quantidade de camadas, e suas responsabilidades, é descrita em um modelo. Os componentes são projetados de acordo a otimizar a possibilidade de reutilização. A decisão quanto à granularidade dos componentes deve levar em consideração que os de nível mais baixo, mais granulares, são mais fáceis de se reutilizar, e que os de alto nível, menos granulares, fornecem maior produtividade ao serem reutilizados. Se forem definidas características selecionáveis em tempo de execução deve constar no projeto que o módulo deve permitir essa funcionalidade.

2.1.2 ENGENHARIA DE APLICAÇÃO

O objetivo da Engenharia de Aplicação é desenvolver um novo produto da LPS, através do reuso do núcleo de ativos e exploração da variabilidade da linha, de acordo com o plano de produção (Pohl et al, 2005; Royer et al, 2013). A geração ideal de um novo produto envolve 100% de reutilização dos componentes construídos na fase de Engenharia de Domínio. A reutilização total, no entanto, não é sempre possível devido a fatores que não podem ser previstos durante a fase de Engenharia de Domínio, como a expansão do escopo por exemplo. Quando surge a necessidade de se acrescentar características a fase de Engenharia de Domínio é executada novamente. As técnicas de implementação, o modo como o aplicativo é codificado para refletir a variabilidade, podem variar mesmo entre produtos de uma mesma linha puramente por questões técnicas (Dalmon, 2012). O processo de criação do produto é denominado Derivação do Produto.

O ideal da Engenharia de Aplicação do desenvolvimento de Linhas de Produto de Software é derivar produtos de forma automática sem a necessidade de intervenção manual por parte dos desenvolvedores. Esse alvo nem sempre é possível de ser alcançado pois depende de fatores como arquitetura, domínios e abordagens (Apel et al, 2013). Por exemplo, em um desenvolvimento voltado para componentes conectores precisam ser criados para realizar a comunicação entre eles.

2.2 JOGOS E JOGOS EDUCACIONAIS

Jogos fazem parte dos componentes culturais de uma sociedade. Estão ubiqüamente presentes desde a infância, adolescência, idade adulta e maturidade. Os jogos colaboram como um componente agregador e gerador de identidade dentro de uma comunidade. Assumem diversas formas, conforme o extrato da sociedade, passando por uma cantiga de roda infantil a um esporte de contato pleno (Sylvester, 2013). Nas seções seguintes são definidos, diferenciados e caracterizados os aspectos de jogos, jogos educacionais e a recente tendência na utilização de gamificação.

2.2.1 JOGOS

A definição do termo jogo enfrenta desafios. Alguns conceituados autores, pesquisadores e profissionais da indústria de jogos, antigos e contemporâneos, afirmam que nenhuma definição formal deveria ser possível (Parlett, 1999; Cage, 2014). Outros autores, no entanto, não se furtam de realizar a descrição da natureza dos jogos e separá-los de outras atividades. Salen et al, (2004) realizou uma intercessão entre algumas definições ao comparar alguns critérios utilizados na literatura, tabela 2.3, antes de formular uma nova definição: “Um jogo é um sistema no qual jogadores entram em um conflito artificial, definido por regras, que resulta em um resultado quantificável”. A afirmação de que jogos são sistemas infere uma característica recorrente: assim como sistemas podem conter e estar contidos em outros sistemas um jogo pode conter outros jogos e estar contido em um jogo.

Tabela 2.3 – Comparação entre elementos na definição de jogos. Adaptado de (Salen et al, 2004).

Elementos de Jogos	Parlett	Abt	Huizinga	Caillois	Suits	Crawford	Costikyan	Avedon
Regras	x	x	x	x	x	x		x
Conflito	x					x		x
Objetivo	x	x			x		x	x
Tomada de Decisões		x				x	x	
Ambiente Artificial			x	x		x		
Voluntário				x	x			x
Evento		x			x			x

Para (Rigby et al, 2011) os jogos satisfazem três necessidades psicológicas básicas, e essa satisfação é o que motiva as pessoas a jogar. As necessidades básicas são: Competência, Autonomia e Relacionamento. A necessidade de competência está relacionada à satisfação de vencer desafios e ter a percepção de progressão nas habilidades requeridas. Um ditado recorrente na indústria de jogos é: “fácil de aprender e difícil de dominar”. A necessidade de autonomia é satisfeita quando o jogador tem liberdade para tomar decisões. Franquias de jogos bem sucedidas fornecem grandes mundos abertos para serem explorados, e decisões importantes para serem tomadas, dando ao jogador uma sensação de liberdade e autonomia apesar de haver uma pré-determinação invisível ao jogador. A última necessidade básica a ser satisfeita é a de se relacionar.

Tradicionalmente jogos são experiências sociais. Surgidos recentemente dentro das redes sociais, Jogos Sociais (*Social Games*), constituem uma nova categoria de jogos online.

Nessa categoria de jogos, para avançar é necessário, ou pelo menos conveniente, pedir ajuda a uma rede de amigos. O sentimento derivado de ajudar e ser ajudado, de fazer parte de uma comunidade e de ter um espaço nela produz bem estar (Chua et al, 2012).

Os jogos possuem algumas estruturas abstratas em comum: regras, ambientes artificiais, conflitos e incerteza (Salen et al, 2004). A característica mais proeminente dos jogos são as regras. Elas definem a forma do jogo, limitam a liberdade dos jogadores e fornecem desafios. Boas regras devem ser explícitas e não ambíguas. Os jogos são executados em ambientes artificialmente preparados. Os jogadores devem interagir dentro desse ambiente fisicamente ou através de peças que representam o jogador. O ambiente é separado do mundo real mesmo que através de linhas imaginárias. O conflito se dá em torno do objetivo. O objetivo é um recurso limitado que nem todos os jogadores conseguem alcançar. Mesmo em jogos cooperativos existe um conflito, comumente com relação ao tempo ou a adversários controlados pelo computador. O último elemento abstrato caracterizador dos jogos é a incerteza. O resultado de uma partida não é pré-determinado, pois depende de fatores aleatórios e de decisões. Um grau muito baixo de elementos aleatórios em um jogo diminui o elemento surpresa, e um grau muito alto de aleatoriedade passa ao jogador a sensação de que suas intervenções não interferem no resultado da partida (Bisadi et al, 2014).

2.2.2 JOGOS EDUCACIONAIS

No tratado “*Some Thoughts Concerning Education*”, de 1693, o filósofo inglês John Lock declara: “Eu sempre tive a idéia de que o aprendizado pode ser transformado em jogos e recreação para as crianças...”. Um pouco mais de três séculos depois o potencial dos jogos educacionais sobre o ensino é declarado como consenso (Arnab et al, 2014).

Também denominados Jogos Sérios (*Serious Games*) os jogos educacionais podem ser classificados como um subconjunto de jogos. O que difere um jogo educacional de um jogo tradicional é o objetivo educacional (Belloti et al, 2013). Enquanto que jogos tradicionais têm como objetivo último o entreterimento, o objetivo dos jogos educacionais é de que o aluno adquira uma unidade de informação.

Os jogos educacionais têm sido utilizados com sucesso em uma ampla gama de domínios e objetivos. Do ensino fundamental ao ensino superior a literatura é farta em casos de uso que demonstram sua eficácia no apoio ao aprendizado. No ambiente corporativo fornecem auxílio para o treinamento e avaliação dos funcionários. Na área da saúde os jogos

sérios contribuem para a reabilitação e integração social dos pacientes e para o treinamento de técnicos, enfermeiros e médicos. Notadamente uma das maiores contribuições para a área é o projeto “*fold-it*” (<http://fold.it>). O projeto conseguiu um grande alcance midiático ao ser anunciada em 2011 a descoberta, feita pelos jogadores, da estrutura de uma proteína dos retrovírus. Essa descoberta abre a possibilidade do desenvolvimento de novas drogas para combater doenças como a AIDS (Khatib et al, 2011).

A literatura fornece evidências de que o aprendizado é facilitado pelo uso de jogos educacionais quando comparado com os métodos tradicionais de instrução (Giannakos et al, 2013; Shin et al, 2012). Fatores como imersão, estimulação sensorial através de multimídias, interatividade e resposta imediata estimulam a motivação do aluno. A motivação de um estudante, e seu desempenho acadêmico, estão correlacionados positivamente: quanto maior a motivação maior o desempenho acadêmico (O’Rourke et al, 2014). Motivação, portanto, se configura um aspecto fundamental para o melhor aproveitamento dos jogos educacionais. Para se alcançar a motivação é necessário que haja uma correspondência entre os jogadores e o jogo. Uma forma de alcançar essa identificação é utilizar a participação do público alvo no projeto de desenvolvimento (Diaz et al, 2012). Ao ser motivado, o aluno voluntariamente aumenta seu tempo de exposição ao conteúdo educacional. Pelos motivos apresentados, a associação de jogos, com objetivos educacionais, tem como alvo mais importante o engajamento do aluno.

Segundo (Schell, 2008) os jogos educacionais apresentam três desvantagens principais. A primeira é que a produção de jogos necessita de muitos recursos pois combina as características de um artefato multimídia com a complexidade de um sistema de informação. Dependendo da complexidade os jogos possuem uma coleção de arquivos de diversas mídias como música, imagens, animações e modelos em três dimensões que necessitam ser produzidos por profissionais capacitados com habilidades específicas. O jogo mais vendido de 2013, Grand Theft Auto V (GTA), necessitou de uma equipe de mais de 1000 profissionais divididos em 5 anos de produção ao custo de \$265 milhões de dólares (Wikipedia, 2014). A figura 2.1 mostra a comparação entre uma tela do jogo educacional Geography Quest (Horachek, 2014), desenvolvido por apenas um programador com ativos gráficos *open source*, e uma tela do GTA. Há uma correlação positiva entre a quantidade de investimento e a qualidade do jogo. Jogos de baixa qualidade não conseguem provocar nos alunos o engajamento necessário. É necessário então obter recursos humanos e financeiros suficientes para alcançar a qualidade desejada.

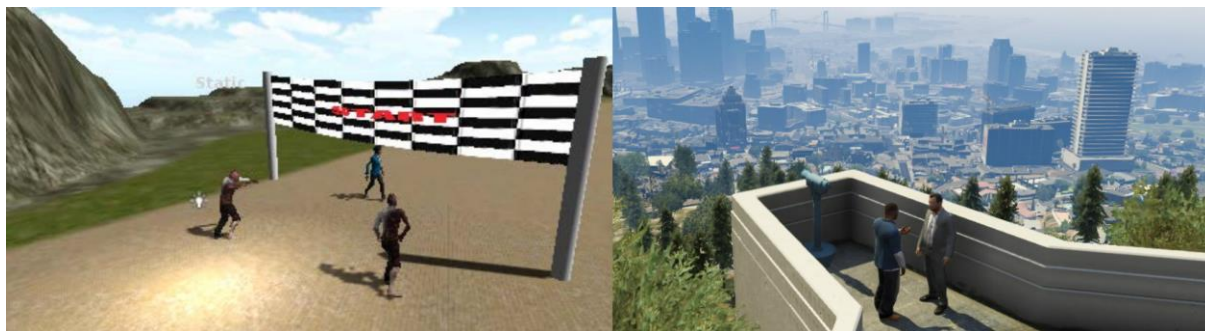


Figura 2.1: Comparação visual entre *Geography Quest* e GTA.

A segunda desvantagem é a dificuldade inata de se tornar transparente o conteúdo a ser ensinado. Enquanto que jogos sérios podem ser capazes de fornecer motivação, as unidades de informação permanecem as mesmas. A solução mais comum da indústria e da literatura é adicionar uma camada, ou filtro, de fantasia através de um tema. O aluno consegue, no entanto, enxergar através dessa camada e perceber seu conteúdo. Se existir uma pré-existente resistência ao conteúdo ministrado o aluno transfere essa resistência para o jogo. Por exemplo, o jogo educacional de realidade aumentada *The Table Mystery* (Boletsis et al, 2013) possui um enredo interessante e que pode ser encontrado na indústria cinematográfica. No desenvolver do jogo, no entanto, os jogadores são apresentados de maneira crua à tabela periódica. Se o aluno teve dificuldades ao ser introduzido à tabela através do ensino convencional, ele pode também encontrar as mesmas dificuldades ao percebê-la no jogo.

A terceira desvantagem dos jogos educacionais é caracterizada pela rápida degradação técnica e tecnológica. Pela visão tecnológica à medida que novas plataformas surgem e atingem popularidade outras plataformas, perdem popularidade e caem em desuso. Um jogo é projetado para rodar em uma ou mais plataformas específicas. Portar um jogo para uma nova plataforma pode se mostrar uma tarefa inviável. Sem a possibilidade de portabilidade todo o investimento despendido é desperdiçado. Pela visão técnica, um jogo educacional se torna obsoleto se o seu conteúdo é defasado. As mudanças de currículo ocorrem com mais frequência à medida de se move do ensino fundamental até o ensino superior. Se o conteúdo não puder ser modificado, ou adaptado, todo o produto pode ser jogado fora.

2.2.3 GAMIFICAÇÃO

Embora o termo gamificação tenha sido cunhado recentemente as estruturas que o compõem têm sido utilizadas há muito tempo e em diversos segmentos (Heizen, 2013). Possivelmente a mais popular menção à gamificação seja a canção “*A Spoonful of Sugar*” do filme *Mary Poppins*: “*In ev'ry job that must be done / There is an element of fun / You find the fun and snap! / The job's a game*” (Sherman et al, 1964).

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.”. Os jogos e os jogos sérios, como descritos nos capítulos anteriores, possuem uma série de fatores sinérgicos que favorecem a participação, interatividade e criatividade. O processo de gamificação é proposto como uma forma de capturar esses fatores e utilizá-los em um contexto diferente. As mecânicas utilizadas em jogos foram refinadas e evoluídas ao longo do tempo e formam uma sólida base para a construção do projeto de *gamification*. São constituídas de regras e sistemas que promovem balanceamento, competitividade e resultados quantificáveis.

A estética dos jogos é cuidadosamente projetada para auxiliar na satisfação de um objetivo tanto prático quanto emocional. Máquinas caça-níquel, por exemplo, são esteticamente projetadas para promover excitação através do uso de sons altos, luzes de várias cores piscando em alta frequência e com números e símbolos girando. Essa excitação gera no jogador uma ansiedade e expectativa com relação à próxima tentativa. O último elemento destacado por Kapp (2012) é o modo de pensar dos jogos. Criatividade, estratégia, reflexos e cooperação são alguns dos requisitos necessários para superar obstáculos do cotidiano. Os jogos permitem o treinamento de habilidades que auxiliam para a melhoria de um indivíduo nesses aspectos. Em um ambiente seguro o jogador pode experimentar, se arriscar e tirar conclusões sobre os resultados. Quando estiver em uma situação, onde o resultado tem consequências reais, o jogador se utiliza de recursos com os quais, em situações similares, obteve um melhor resultado.

Tão importante quanto definir o termo gamificação é definir o que não faz parte desse contexto. Apenas utilizar um sistema de recompensas e penalidades pode não ser o suficiente para caracterizar a utilização do processo. Como estipulado por (Kohn, 1999) o uso do sistema “*Carrot and Stick*” é uma solução de curto prazo. A expressão “*Carrot and Stick*” é utilizada para designar quaisquer sistemas que recompensem o comportamento ou

desempenho desejado e punam o comportamento ou desempenho indesejado. A expressão é derivada da metáfora na qual o proprietário de uma carroça vê seu burro empacado, persistentemente se recusando a mover. O dono do burro então tem duas opções: segurar uma cenoura na frente do burro de modo que o burro se mova para tentar pegá-la ou utilizar a vara e fazer o burro se mover através do castigo físico. Esse sistema está tão presente no cotidiano que pode parecer absurdo parar para pensar se ele realmente funciona da forma como é esperado. No entanto, experiências realizadas pela área de Economia Comportamental comprovam que o sistema não funciona sempre da forma prevista. Um exemplo é a conclusão, reforçada por diversos experimentos de pesquisadores diferentes, em épocas diferentes, de que em atividades que envolvam a criatividade e resolução de problemas as pessoas têm um desempenho menor quando é oferecida uma recompensa (Ariely, 2010).

Em vez de engajar os participantes a punição, por falha, gera aversão pela atividade que se transforma em um gerador de sofrimento para aqueles que não conseguem alcançar uma meta arbitrariamente definida. Em contrapartida, também está fora do escopo do processo de gamificação a trivialização do ensino e demais atividades. Matérias complexas continuam sendo complexas e atividades extenuantes continuam extenuantes. O mesmo esforço e dedicação são necessários para se dominar um assunto ou habilidade após sua implantação. O que muda é a atitude e comprometimento no relacionamento com a atividade. Uma das características desejáveis no *design* de um processo de gamificação é capacidade compelir o usuário a alcançar um estado mental denominado, pelos psicólogos, fluxo (Nah et al, 2014). Csikszentmihalyi (1990) define fluxo como sendo, um estado de consciência, onde o indivíduo está completamente absorto em uma atividade alcançando um nível de performance ótimo. Quando uma pessoa entra no estado de fluxo ela perde a noção de tempo e de sintomas fisiológicos como fome, cansaço e sono. Um dos pré-requisitos que despertam o estado de fluxo é a satisfação (*enjoyment*) com a atividade a ser desempenhada. Por exemplo, crianças podem passar várias horas jogando videogame em um estado absoluto de concentração porque obtém prazer dessa atividade, mas não conseguem o mesmo nível de concentração em uma sala de aula de onde não obtém o mesmo nível de satisfação. A figura 2.2 demonstra a relação entre o nível de habilidade e o nível do desafio necessário para haver a caracterização do fluxo.

As fronteiras que definem o que é um jogo educacional e o que é gamificação podem ser nebulosas. O ponto em comum entre os dois conceitos é o objetivo de engajar usuários na realização de determinadas atividades. Jogos sérios pertencem a um subconjunto de jogos. Sendo que um jogo possui características como tempo finito e a 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. Isso é importante por que processos dentro de uma organização apresentam uma natureza de continuidade em que não há vencedores ou perdedores. Embora um dos mecanismos de motivação de gamificação seja a recompensa e a valorização de bons resultados, declarar uma minoria como vencedores e uma maioria como perdedores pode ter o efeito de desmoralizar grande parte de uma equipe (Kohn, 1999).

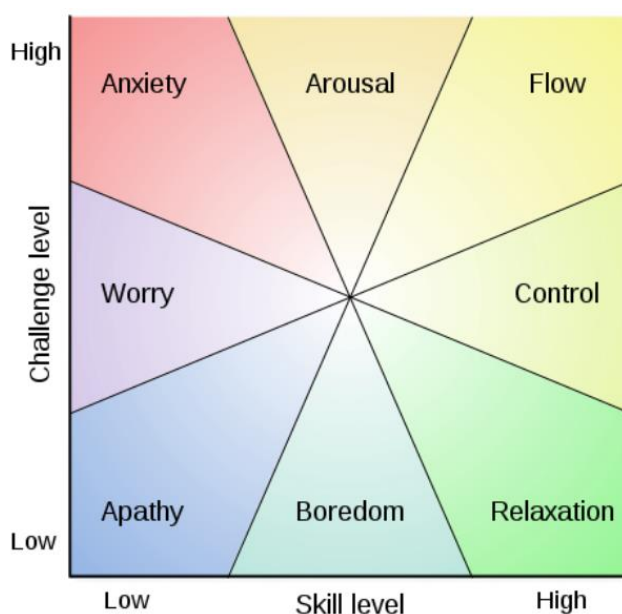


Figura 2.2: *Flow*: Nível do Desafio vs Habilidade (Beatson, 2008).

Determinar exatamente como os jogos conseguem fomentar o engajamento não é uma tarefa trivial. Boyler et al (2012) redigiu uma revisão sistemática sobre a identidade do engajamento produzido por jogos digitais. Os resultados da pesquisa de Boyler et al (2012) sugerem uma série de fatores que podem aumentar o nível de engajamento dos jogadores. Jogos mais rápidos tendem a ser mais imersivos e recebem mais atenção por parte do jogador porém aumentam seu nível de *stress*. Tantos jogos com temas violentos quanto não-violentos podem receber dos jogadores o mesmo nível de engajamento. Jogadores preferem jogar com outras pessoas a jogar com inteligências artificiais. O nível de detalhamento gráfico influencia o engajamento. Quanto mais elaborado o visual de um jogo maior a satisfação do jogador, assim como maior velocidade de resposta e o controle sobre o jogo. Uma história de plano de fundo ajuda no processo de identificação do jogador com o jogo. Bouvier et al (2013) sugerem estudar as interações do aluno para a realização de um mapeamento do nível de engajamento do jogador.

O processo de gamificação tem sido adotado com sucesso por organizações de diversos tamanhos e áreas de atuação com resultados positivos. Da área militar, como uma forma de auxiliar no recrutamento e treinamento, à medicina terapêutica no combate à dor e como catalizadora da recuperação psíquica e emocional. Na indústria, gamificação tem sido utilizado para treinamento e aumento de produtividade e no setor de serviços como uma ferramenta para aumentar a inovação, atrair novos talentos e diminuir o ritmo de saída dos empregados (*turnover*). A seguir, são descritas algumas bem sucedidas implementações de gamificação em diferentes áreas.

A fundação sem fins lucrativos Khan Academy é uma das maiores fontes de materiais de ensino à distância disponíveis. Em 2013, o governo brasileiro comprou o material traduzido para o português como parte do conteúdo de 400 mil *tablets* distribuídos para professores do ensino médio (Lemann, 2013). A Khan Academy utiliza gamificação como um auxílio para diminuir a evasão, notadamente alta em cursos a distância, e aumentar a motivação. A cada sequência de exercícios, respondidos corretamente, o aluno tem sua progressão imediatamente marcada em um *dashboard*. O aluno tem diferentes níveis de domínio em cada habilidade que variam de “não iniciado” a “dominado”. Ao assistir a um vídeo ou realizar um exercício o aluno ganha “pontos de energia” e medalhas que possuem pré-requisitos para se conquistar.

Em janeiro de 2012 a empresa de artigos esportivos Nike lançou o acessório *Fuel Band*. A pulseira sincronizada com um *smartphone* registra o nível de atividade física do usuário. Pelo exercício físico o portador da pulseira é recompensado automaticamente com *Fuel Points*. O usuário se compromete com uma determinada meta diária e se esforça para cumpri-la. Quando a meta é alcançada o resultado é publicado *online* para que os amigos do usuário possam ver. Também é possível participar de um *leaderboard* onde o usuário compete com amigos disputando o melhor resultado em um determinado período de tempo. A tentativa bem sucedida de gamificação esportiva resultou em uma base de 11 milhões de usuários e um aumento nos lucros de 18% (Knowledge, 2013).

Na indústria alimentícia se destacam a campanha *Monopoly* do McDonalds e a *Shake It* da Coca-Cola. Iniciada em 1987, e realizada todos os anos no mês de outubro, a campanha *Monopoly* consiste em colecionar *tickets* ganhos a cada compra e completar todas as casas do jogo para ganhar prêmios em dinheiro, viagens e vales-compra. A campanha *Shake It* é voltada apenas para os consumidores de Hong Kong, e consiste de um aplicativo a ser instalado em um *smartphone* que, se sacudido na hora em que passa um comercial da Coca-Cola, permite ao consumidor a chance de ganhar prêmios. Ao final da campanha, o aplicativo

obteve a marca de 9 milhões de *downloads*, ou seja, 1.28 vezes a população de Hong Kong.

Em termos de participação de mercado gamificação ainda não possui uma expressão tão significativa quanto os jogos sérios. Em 2013, um faturamento de \$421.3 milhões de dólares foi alcançado com uma projeção de crescimento para \$5.50 bilhões de dólares em 2018 (Market, 2014).

As desvantagens na utilização de gamificação são caracterizadas por dois fatos: não é fácil de se desenvolver e nem todas as pessoas se sentem motivadas pelos elementos presentes nos jogos. Acrescentar elementos a um processo, em primeiro lugar, pressupõe a modelagem desse processo. Nem todas as organizações possuem processos bem definidos ou mesmo modelados. Em segundo lugar, é necessário conhecer muito bem o público alvo. A apresentação estética e a dinâmica de um jogo estão estritamente atreladas a um tipo de público específico. Deve-se levar em consideração que socialização, competição e riscos são fatores, ubiquamente presentes nos jogos, que trazem ansiedade a uma parcela da população e em alguns casos até mesmo pânico.

Em (Hamari et al, 2014) é realizada uma revisão sistemática sobre sistemas de gamificação e a conclusão é de que não há “... *frameworks* teóricos bem definidos ou discursos unificados”. De fato embora a literatura seja abundante em pontos comuns não existe até o presente momento uma unanimidade ou consenso geral sobre o que caracteriza o processo. No entanto existem propostas que se aproximam de uma solução definitiva como é o caso do *framework* conceitual Octalysis (Chou, 2013). Proposto por Yu-Kai Chou, consultor em gamificação desde 2003, o *framework* Octalysis propõe oito dimensões ou categorias de técnicas de gamificação: significado, desenvolvimento, controle, sociabilidade, incerteza, perda, raridade e posse. A dimensão Significado compreende a imersão do jogador em um objetivo nobre e coletivo. Estar envolvido em uma tarefa que tenha um significado maior como ajudar o próximo que esteja em necessidade, é um dos fatores que promovem o engajamento e a sensação de bem estar. Ter a sensação de pertencer a um grupo, que tem um objetivo comum, fornece segurança e a caracterização de identidade.

Desenvolvimento é a dimensão que demonstra, visualmente, e através de regras as etapas percorridas e o progresso conquistado pelo jogador. A maneira mais comum de demonstrar progresso é através da concessão de pontos. Quanto mais pontos o jogador fizer mais longe ele estará. Quanto melhor sua atuação mais rapidamente ele adquirirá pontos.

Controle é a categoria de técnicas que transmitem ao jogador a possibilidade de exercer a criatividade sobre o mundo de jogo e receber de volta um retorno sobre as consequências de suas ações (*feedback*). O controle satisfaz a necessidade de autonomia que é

definida por (Rigby et al, 2011), conforme explicado na seção 2.2.1, como uma das três necessidades básicas do ser humano que os jogos conseguem suprir.

Sociabilidade é a categoria que agrupa técnicas de interação social. As interações podem ser cooperativas, competitivas ou comunicativas. Nem todos os jogos possuem características de sociabilidade.

Incerteza como descrito por Costikyan et al (2013) é a característica primária de todos os jogos. Nesta categoria se encontram técnicas que despertam no jogador a curiosidade pelo desconhecido, o prazer de uma recompensa inesperada e a surpresa de uma reviravolta positiva ou negativa.

Perda é a dimensão que agrupa características que punem o jogador de alguma forma. A punição mais comum é a perda da partida.

Conforme descrito na categoria Incerteza o resultado de um jogo não é previamente conhecido. Por toda a partida existe a chance de perda. Essa dimensão mantém o jogador atento e em suspense. A gravidade e a frequência da perda variam de acordo com o gênero do jogo.

Raridade é a categoria que emprega a inserção de exclusividade em um jogo. Os jogadores se sentem especialmente motivados a se esforçarem mais para conseguir algo que seja único, exclusivo ou raro. A forma mais comum de monetização em um jogo “grátis” é a presença de itens raros e pagos.

A última categoria proposta pelo *framework* Octalysis é Posse. Em economia comportamental existe um fenômeno conhecido como “*endowment effect*” que se caracteriza pelo, geralmente, alto valor que as pessoas atribuem às suas posses (Ariely, 2009). Em ambientes virtuais o fenômeno se repete. Pessoas se dispõem a gastar tempo e dinheiro no cuidado de fazendas, vilas e bichos de estimação que só estão presentes como uma representação gráfica de dados armazenados em servidores cuja localização sequer conhecem. Quanto mais tempo e dinheiro despendido mais dedicação ao objeto virtual.

2.2.4 METADADOS

A reutilização é um conceito chave para Objetos de Aprendizagem (OA). Um OA deve ser granular, isto é, deve ser pequeno ou composto de pequenas partes. Quanto maior a granularidade de um OA, maior será a sua possibilidade de reuso uma vez que um OA pode conter várias partes que podem ser extraídas formando vários outros OA (Burbaite e Stuikys, 2011).

No contexto de desenvolvimento de software para a área de Educação é importante a criação de mecanismos que agilizem o processo de construção de OAs reutilizáveis e que facilitem a sua integração e utilização em repositórios. O uso de abordagens de desenvolvimento de software tais como Linha de Produtos de Software (LPS) pode oferecer um suporte sistemático para a reutilização de OAs. Os benefícios proporcionados são: maior agilidade e menor custo na reutilização e criação (Silva et al., 2012).

Por outro lado, considerando o contexto de LPS, algumas características são mandatórias, para os objetos de aprendizagem, como autoria, título, idioma e versão, mesmo que muitas vezes implícitas. Outras são alternativas, que podem representar as características dos diferentes tipos de OA como exercício e prática, tutorial hiperídia, jogos, por exemplo, e outras características são opcionais. Nesse contexto, uma solução viável para a semântica das aplicações educacionais, utilizada no projeto BROAD é a criação de redes de ontologias que permitam a criação de ontologias necessárias a uma aplicação e domínios específicos, mas também o reuso de ontologias existentes como o padrão OBAA (2012), mapeados para o modelo de features de forma a facilitar a geração de OAs a partir da LPS.

A localização e recuperação de OAs exigem que eles tenham sido identificados/classificados a partir de um modelo robusto de metadados, capaz de descrevê-lo de forma completa e correta, com o objetivo de facilitar seu uso. Ocorre que há muitos padrões disponíveis, mas eles quase sempre são difíceis de serem adotados, além de serem extensos, complexos e muitas vezes negligenciarem os aspectos educacionais envolvidos no OA.

Metadados segundo (Giunchiglia et al, 2013) são dados que definem um nível de abstração maior de dados. Neste trabalho são selecionados três padrões de metadados: LOM, OBAA e BROAD. Estes três padrões são desenvolvidos com o objetivo de descrever Objetos de Aprendizagem (OA). Segundo (Gonçalves, 2014): “Os objetos de aprendizagem são atividades educativas que podem abordar o conteúdo de uma disciplina por meio de jogos eletrônicos nos quais se sobrepõem brincadeiras, interatividade, lógica e criatividade.”. O padrão LOM define nove categorias que descrevem sessenta e sete características dos OA como, por exemplo, a categoria Ciclo de Vida que possui três características: versão, status e contribuições. Cada característica possui uma descrição dos dados que são projetados para comportar subdivisões. A característica status, por exemplo, possui um conjunto de valores fixos: rascunho, final, revisado e indisponível.

O padrão OBAA propõe alguns acréscimos ao LOM com o intuito de permitir a execução de um OA em diferentes dispositivos, incluindo TV Digital, e propriedades de

acessibilidade. Na categoria Técnica são acrescentadas 18 características, na categoria Educacional três características. Duas categorias são acrescentadas: Acessibilidade com seis características e Tabela de Segmentação com 17 características. O padrão BROAD (Souza et al, 2010) é o resultado de uma seleção de metadados essenciais do LOM e OBAA. Apresenta sete categorias e trinta e sete características. A figura 4.4 mostra uma representação gráfica parcial do padrão BROAD.

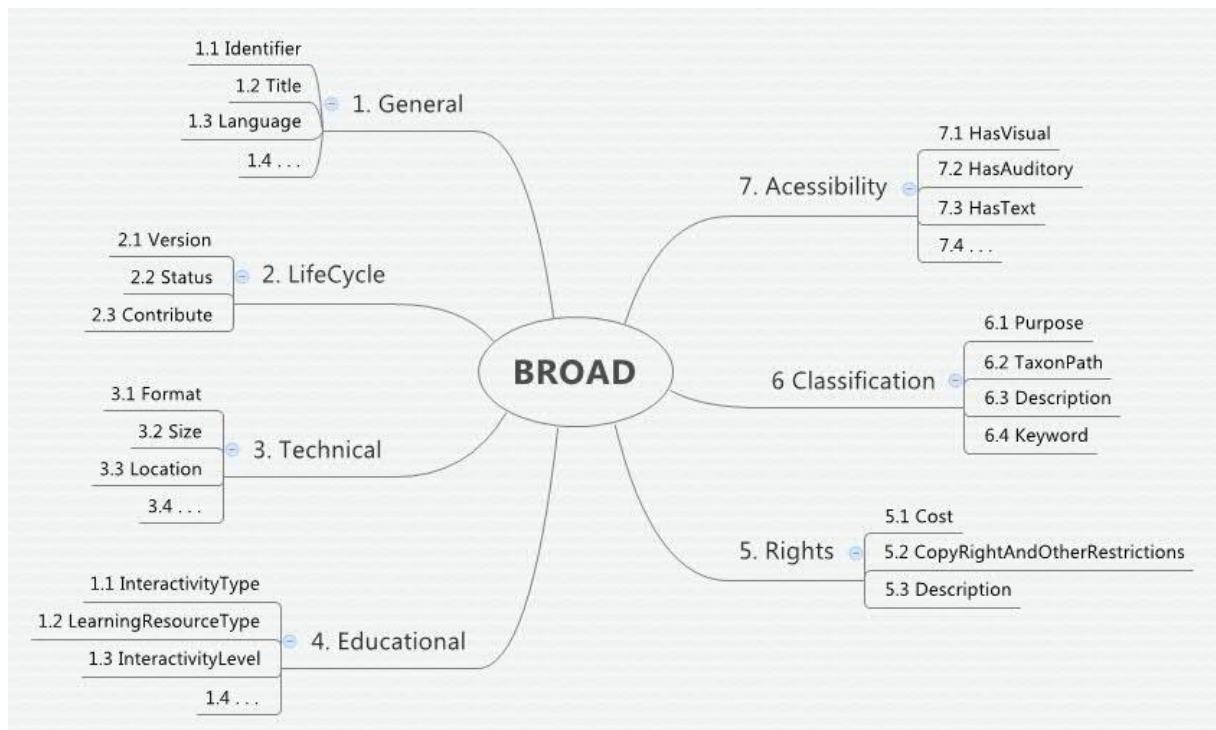


Figura 2.3: Visão parcial dos metadados do BROAD. Adaptado de (Souza et al, 2010).

2.2.5 CONSIDERAÇÕES FINAIS DO CAPÍTULO

Neste capítulo foi realizada a descrição dos principais fundamentos relativos à LPS. Como demonstrado é uma abordagem viável para se conseguir maior produtividade e qualidade nos produtos desenvolvidos. São citados exemplos reais de empresas de grande porte que utilizam LPS para produzir artefatos de software de qualidade. Contudo, a adoção de uma linha não é um processo simples. A realização do primeiro produto é mais custosa variando o custo de acordo com a abordagem de adoção utilizada: proativa, extrativa ou reativa. A etapa de Análise de Domínio prepara artefatos para serem reutilizados e a etapa de Engenharia de Aplicação gera novos produtos através da reutilização promovida pela etapa anterior.

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.
- Permitir a reutilização sistemática de Objetos de Aprendizagem no domínio de Jogos Educacionais que constituem um domínio cujos elementos possuem evidentes características em comum.
- Permitir a criação de arquiteturas de alto nível que comportam implementações para uma grande variedade de ambientes de desenvolvimento e execução.

Não é uma tarefa trivial compreender e dominar todos os conceitos que envolvem LPS. Embora as definições encontradas na literatura sejam sintéticas e similares entre si, existem muitos aspectos a serem explorados. Analisar o domínio, gerenciar as variabilidades, testar os produtos e definir qual a melhor maneira de controlar a evolução da linha são algumas das questões em aberto e que fazem parte dos principais tópicos frequentemente abordados em trabalhos sobre o tema. Neste trabalho o foco principal é propor características comuns e variáveis para jogos, Jogos Educacionais e Gamificação. O objetivo das caracterizações é prover uma base fundamental para a construção de produtos relacionados a esses temas. Uma arquitetura de alto nível é proposta na seção 4.1 para auxiliar na implementação de um processo de Engenharia de Aplicação.

A pesquisa em torno de jogos é cada vez mais relevante porque, como descrito na seção 2.2.1, jogos fazem parte do processo de socialização de praticamente todas as culturas. Em uma sociedade onde o desenvolvimento tecnológico consegue resolver grande parte dos problemas mais básicos como alimentação e segurança normalmente há bastante tempo livre disponível. Independente de idade e classe social pessoas utilizam jogos no seu tempo livre como uma forma de entretenimento, treinamento e relaxamento. O mercado de desenvolvimento e venda de jogos eletrônicos já ocupa uma margem considerável de movimentação financeira e cresce rapidamente a cada ano. Para a Ciência da Computação jogos são importantes por apresentarem desafios tecnológicos como necessidade de ambientes distribuídos, comunicação em tempo real, gráficos em estado da arte e realidade virtual.

A escolha de Jogos Educacionais foi feita com base em dois fatores principais: as melhorias que trazem para o processo de aprendizado, e o ainda presente desafio de se criar jogos de qualidade em larga escala. Os Jogos Educacionais conseguem estimular no aluno, através da interação, resposta imediata e imersão uma prontidão para o ensino fornecido pelo aplicativo. No entanto, trazer para a sala de aula um objeto educacional que consiga capturar a atenção do aluno por um razoável período de tempo ainda é uma realidade distante em muitas

escolas, especialmente as públicas. A abordagem de LPS pode auxiliar nesse processo especialmente em um país das dimensões do Brasil que possui diferentes realidades regionalizadas. Assim como foi demonstrado por Chimalakonda, cujo trabalho é descrito na seção 3.4, a LPS consegue obter bons resultados na produção de aplicativos em massa.

A decisão de acrescentar o processo de gamificação à LPS foi tomada por causa da sua relação sinérgica com jogos e jogos educacionais. Enquanto que 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 que 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 ele dissemine suas atividades e interesses com os amigos e membros dos círculos sociais aos quais pertencem. Enquanto que oferecer um Jogo Educacional para uma determinada disciplina já possa ser considerada um começo de gamificação somente com um projeto bem elaborado e específico de inserção de atributos e qualidades de jogos é que se obtém a experiência completa.

3 TRABALHOS RELACIONADOS

Neste capítulo são apresentados trabalhos relacionados à construção de jogos e artefatos de *software* educacionais utilizando-se da abordagem de Linha de Produtos de *Software*. Os trabalhos descritos abordam o desafio de produzir artefatos de jogos e produtos educacionais de forma sistemática com o objetivo de aumentar a produtividade e a qualidade dos artefatos desenvolvidos. São apresentados quatro trabalhos com características bem distintas. O primeiro, o Arcade Game Maker (SEI, 2014) é um exemplo pedagógico formulado para divulgar a abordagem de LPS tanto para a indústria quanto para o meio acadêmico. O segundo e terceiro trabalhos abordam módulos interativos e tutores inteligentes, respectivamente, com a fase de Engenharia de Aplicação gerando produtos através de um arcabouço e de uma ferramenta de autoria. O quarto trabalho possui uma abordagem mais pragmática e utiliza a sistematização para produzir sistemas educativos em massa. Por fim, o último trabalho descrito utiliza LPS para a criação de jogos de RPG para celulares.

3.1 ARCADE GAME MAKER

A linha de produto Arcade Game Maker (AGM) (SEI, 2014) é um exemplo pedagógico criado pelo SEI (Software Engineering Institute) com a intenção principal de fornecer um exemplo real e abrangente tendo como base a abordagem descrita no livro “*Software Product Lines: Practices and Patterns*” escrito por Paul Clements e Linda Northrop. A AGM é utilizada como material base de cursos de graduação e pós-graduação nos quais o retorno fornecido pela administração das disciplinas contribuiu para a realização de atualizações ao longo dos anos. O escopo da AGM é considerado pelos autores como sendo simples, porém com cobertura necessária e suficiente dos vários tópicos que são atingidos pela implementação de uma linha de produtos. A AGM abrange um conjunto de características que suprem a necessidade de uma companhia fictícia denominada Arcade Game Maker, incluindo a viabilidade comercial, fornecendo em um documento de contexto as características da empresa, seus funcionários e sua situação geral. Esse contexto é importante para se entender as decisões, de projeto e de implementação, que foram tomadas pelos autores da linha de produtos.

O SEI é um centro de pesquisa e desenvolvimento dotado com recursos federais do Departamento de Defesa dos Estados Unidos e administrado pela Universidade Carnegie

Mellon situada na cidade de Pittsburgh, Pennsylvania. O objetivo do SEI é desenvolver a Engenharia de Software de modo a auxiliar no seu objetivo que é entregar produtos livres de defeitos, dentro do orçamento e no tempo pré-determinado. Embora o Instituto atenda principalmente ao governo americano, e suas necessidades, as tecnologias e métodos desenvolvidos são continuamente compartilhados com a comunidade global de pesquisadores e empresas através de cursos, conferências e parcerias.

Desde sua criação o AGM foi utilizado como exemplo e caso de uso para teste de novas tecnologias e abordagens em 72 artigos cujos cinco primeiros somam 179 citações. A seguir são descritos alguns artigos que utilizam o AGM como prova de conceito de novas abordagens em Linha de Produtos de Software.

Em (Lancelot, 2013) é apresentado um analisador de modelos de variabilidade de Linha de Produtos de Software baseados em UML chamado SMartyParser. O AGM é modelado utilizando-se uma ferramenta UML e, então, exportado no formato XMI para ser analisado com o SMartyParser. Em (Junior, 2013) é descrito o método System-PLA (Systematic Evaluation Method for UML-based Software Product Line Architecture). Segundo os autores, o método se destaca por ser capaz de estimar e analisar potenciais produtos, utilizar de métricas básicas baseadas em UML para compor métricas de atributo de qualidade, analisar a viabilidade e o *trade-off* da LPS de acordo com seus atributos de qualidade e tornar a avaliação da linha de produtos mais flexível. O AGM é utilizado como prova de conceito para o método. Em (Marcolino, 2013) o AGM é utilizado em um experimento de validação de métricas baseadas em complexidade. São utilizados estudantes de graduação de três universidades diferentes em sua maioria com pouca ou nenhuma experiência e conhecimento de UML e Linhas de Produtos para utilizarem as métricas no AGM. O resultado do experimento é que mesmo alunos com pouco conhecimento conseguiram com sucesso calcular as métricas.

O AGM é extensivamente detalhado começando com uma descrição da composição da companhia fictícia Arcade Game Maker que conta com uma diretoria composta pelo presidente, chefe executivo, vice-presidente de planejamento de produtos, vice-presidente de desenvolvimento de produtos, vice-presidente para assuntos de negócio e Diretor de Recursos Humanos. A equipe de desenvolvimento responde diretamente ao vice-presidente de desenvolvimento de produtos e é compartilhada entre quatro times funcionais (Interface de Usuário, Motor de Jogos, Arquitetura e Banco de Dados) e quatro times de projeto: um para o desenvolvimento dos ativos do núcleo e um para cada jogo.

A documentação do AGM recomenda que a construção da LPS seja feita através da

abordagem incremental. O método de desenvolvimento incremental de uma Linha de Produtos de Software 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 começando com zero e chegando próximo a 100%. O principal custo de manutenção recai, a cada interação, sobre o primeiro produto pois o esforço de manutenção é iniciado a partir da criação do ativo.

Os requisitos levantados pela documentação da AGM são descritos pelos seguintes casos de uso, representados na figura 3.1: *Play selected game*, *Exit game*, *Save game*, *Save score*, *Check previous best score*, *Play Brickles*, *Play Pong*, *Play Bowling*, *Initialization*, *Animation loop*, *Install game* e *Uninstall game*.

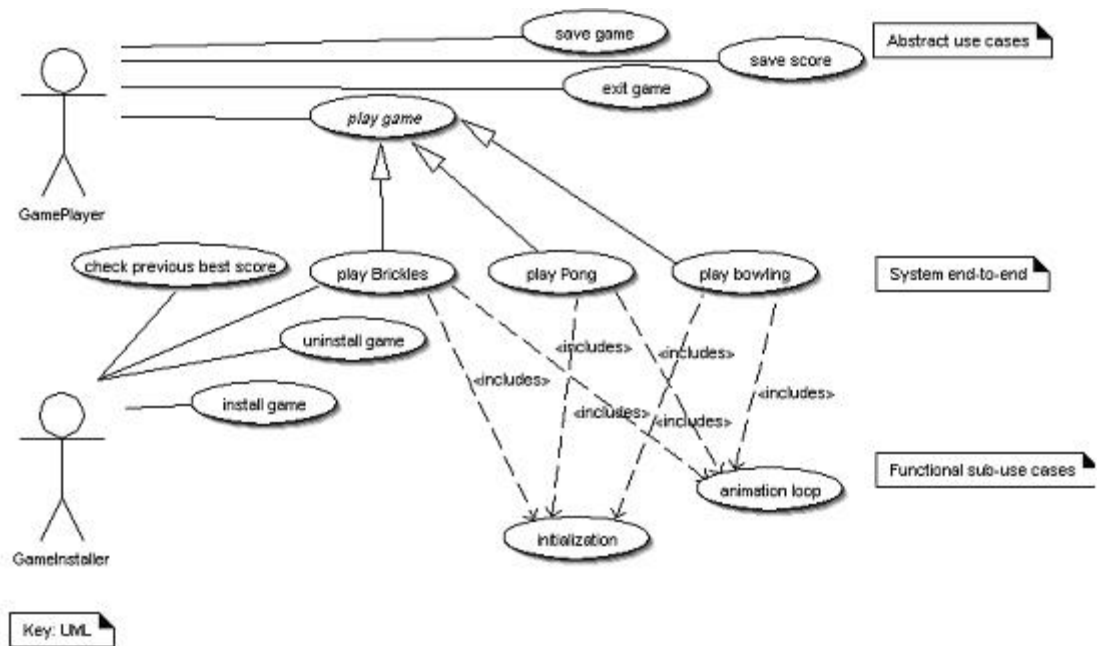


Figura 3.1: Diagrama de Casos de Uso da Arcade Game Maler

O caso de uso *Animation Loop* se faz referência a uma característica muito marcante dos jogos eletrônicos que é o laço principal do programa dentro do qual ocorrem as atualizações dos elementos gráficos e lógicos. Esse laço principal é executado diversas vezes por segundo sendo o elemento principal que define a fluidez da jogabilidade. Uma taxa considerada aceitável para um jogo de arcade é de 60 repetições do laço por segundo.

A documentação do AGM representa a variabilidade da linha através de um diagrama de *features*. O diagrama de *features* da AGM é representado na figura 3.2.

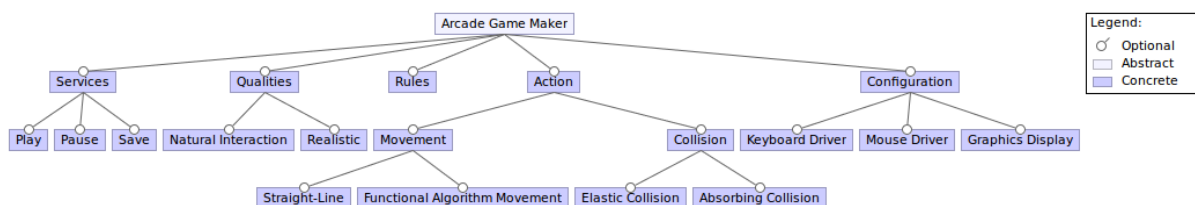


Figura 3.2: Diagrama de Features da Arcade Game Maker. Adaptado de SEI (2009).

3.2 LINHA DE PRODUTOS DE SOFTWARE PARA MÓDULOS DE APRENDIZAGEM INTERATIVA

Em (Dalmon, 2012) é proposta uma linha de produtos para Módulos de Aprendizagem Interativos (iMA). Os iMA são aplicativos educacionais de atividades interativas. Desenvolvidos por pesquisadores da Universidade de São Paulo, seu objetivo é auxiliar o professor acrescentando ao seu ferramental didático atividades interativas. São, originalmente, concebidos para serem construídos através de ferramentas de autoria, e utilizados junto com Sistemas Gerenciadores de Curso (SGC) com os quais se comunicam através de um protocolo específico. Possuem a particularidade de oferecerem as funcionalidades de avaliação automática e retroatividade (*feedback*).

A motivação que alavanca o trabalho encontra base na constatação, feita através de revisão da literatura, de que fazer software educacional não é uma tarefa fácil. Além dos problemas comumente apontados por equipes de software como falta de metodologia, comunicação e experiência, o desenvolvimento de aplicações educativas ultrapassam fronteiras de múltiplas áreas de conhecimento por causa da sua natureza eminentemente multidisciplinar. A realização de uma pesquisa com outros grupos de desenvolvedores acadêmicos de software educacional concluiu que 57% dos projetos possuem 1 ou dois desenvolvedores, 55% tem três anos ou menos de experiência com programação e 59% dos projetos são desenvolvidos sem metodologia (Dalmon, 2012). A metodologia de LPS, é então, a resposta dos autores para facilitar o desenvolvimento dos iMAs.

O início da fase de Engenharia de Domínio capturou cinco requisitos base:

- i) Ser acessível através da web da forma de um *applet* (miniaplicativo).
- ii) Ser capaz de se comunicar com o SGC.
- iii) Possuir uma ferramenta de autoria.
- iv) possuir interatividade
- v) Ser capaz de corrigir automaticamente as respostas dos alunos.

Após o levantamento dos requisitos é construída uma lista de funcionalidades que são visíveis ao usuário como, por exemplo: executar *script*, importar imagem e mostrar gabarito do exercício. O levantamento foi feito com base nas funcionalidades dos iMAs já existentes. Com os requisitos e as funcionalidades elegidas é realizado o mapeamento das características para um modelo de *features*, como descrito na seção 2.1. A figura 3.3 mostra o diagrama de *features* dos iMAs.

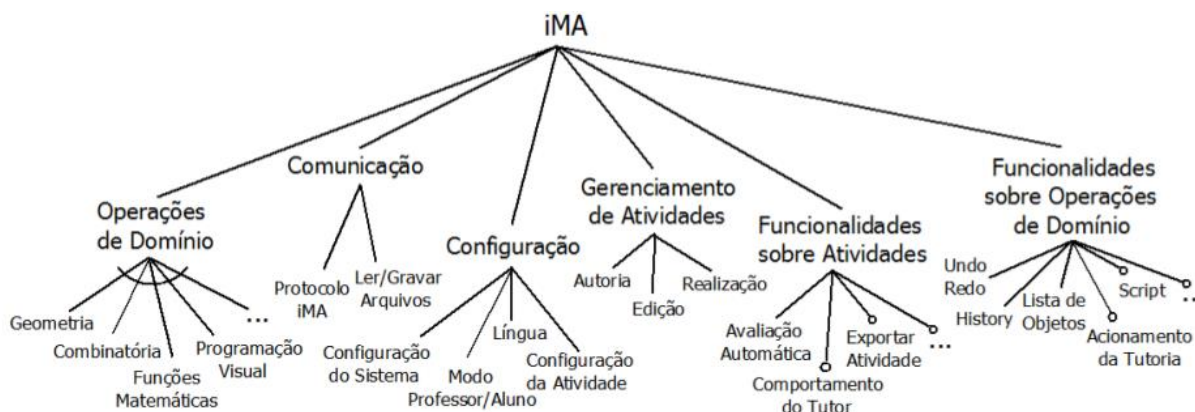


Figura 3.3: Diagrama de *Features* de iMA (Dalmon, 2012).

A fase de Engenharia de Aplicação da LPS de iMA começa com a instanciação do arcabouço. Ao iniciar a construção de um novo produto uma nova instância do arcabouço é gerada e as especificidades do novo produto são acrescentadas. O arcabouço forma a peça central da materialização do modelo da linha e, segundo o autor: “... *no cotidiano do andamento deste projeto, muitas vezes a LPS se confundia com ele*”. O arcabouço foi desenvolvido baseado em uma arquitetura de componentes ilustrada na figura 3.4. As especificidades do novo produto são implementadas através da extensão dos componentes *DomainModel* e *DomainUserInterface*.

A avaliação da proposta da LPS foi realizada através de uma prova de conceitos. O nome do novo produto da linha é iTangram, um iMA, que simula a manipulação do antigo quebra-cabeça chinês que consiste de cinco peças geométricas que são reorganizadas para formarem figuras. O desenvolvimento foi realizado com três programadores estudantes pós-graduandos em Ciência da Computação. Todos os desenvolvedores possuíam experiência prévia em programação e foram instruídos na utilização do LPS de iMAs através de uma vídeo-aula e um manual. Durante o desenvolvimento o autor estava disponível para sanar dúvidas e realizar correções no arcabouço. A conclusão da prova de conceitos, segundo o autor, é satisfatória. O produto iTangram foi criado com 13 novas classes e, aproximadamente 700 linhas de código, sendo que um produto gerado para testes produz 500. As principais

contribuições da LPS de IMA, segundo os desenvolvedores, foi a organização do código e funcionalidades pré-existentes independentes de domínio. Uma das falhas apontadas pelos desenvolvedores foi a ausência do protocolo de comunicação com o SGC. A figura 3.5 mostra a execução do iTangram.

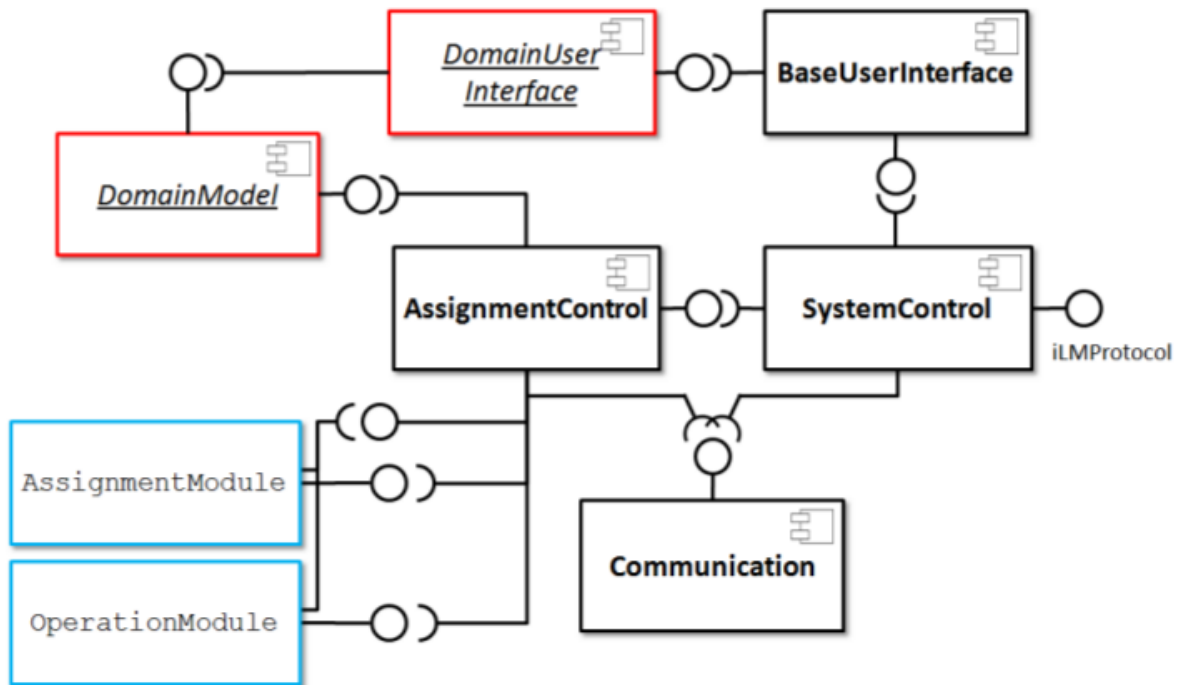


Figura 3.4: Diagrama de componentes do arcabouço da iMA (Dalmon, 2012).

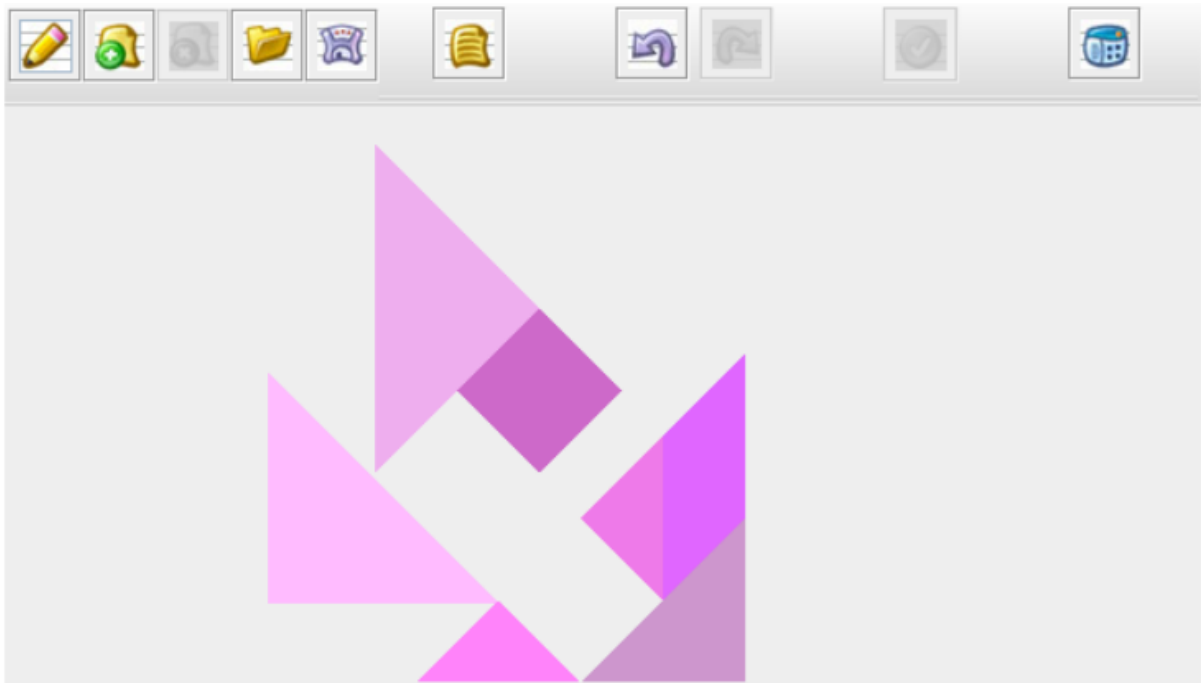


Figura 3.5: Tela de execução do iTangram

3.3 LINHA DE PRODUTOS DE SOFTWARE BASEADA NA WEB SEMÂNTICA PARA TUTORES INTELIGENTES

Em (Silva et al, 2012; Silva, 2011) é proposta uma Linha de Produtos de Software para Sistemas de Tutores Inteligentes (STI). Um STI é um sistema de informação que, através da utilização de técnicas de Inteligência Artificial (IA), reproduz o conhecimento e o comportamento de um professor (tutor) dentro de um ambiente de ensino. A construção de um STI é uma prática multidisciplinar que envolve as áreas de IA, psicologia e educação. Um STI típico tem, então, que modelar três aspectos básicos: Domínio, Aprendiz e Estratégias Pedagógicas. A modelagem do domínio, dentro do contexto do STI e não da LPS, representa a aptidão do tutor e seu escopo. A modelagem do domínio representa formalmente os fatos referentes a um campo específico do conhecimento. A modelagem do aprendiz representa a presença do estudante e suas interações em relação ao STI. Todas as respostas, avaliações e conhecimentos adquiridos devem ser armazenados e previstos na modelagem. A modelagem das Estratégias Pedagógicas considera quais estratégias e táticas serão utilizadas no ensino. As estratégias pedagógicas podem ser baseadas em teorias consagradas ou baseadas no empiricismo através da experiência de cada organização que atua na criação de um STI.

A motivação dos autores para criar uma LPS de STI vem da realização de que a web fornece um ambiente propício ao ensino à distância. Para fornecer essa modalidade de ensino através da rede, no entanto, são necessárias diversos recursos e tecnologias como fórum, *chat*, e-mail, *sites* e objetos digitais de áudio e vídeo. Esses recursos não atuam por si só na realização do ensino-aprendizagem. A necessidade de um professor/tutor é indispensável. Por se caracterizar, primariamente, como um meio de comunicação assíncrono, ou seja, a resposta não tem um tempo pré-determinado para chegar, o professor pode ser requisitado a qualquer momento independente do horário. Além de tarefas como preparação de conteúdos, avaliações e publicação do conteúdo educacional que se tornam uma carga adicional para o docente. Os STI visam suprir, em parte, a necessidade da participação do tutor. Além de desonerar o professor o STI conta com a vantagem de estar disponível em qualquer horário para o aluno. Por se tratar de uma atividade complexa os autores propõem que a construção de STI é beneficiada pelo uso de LPS.

A fase da Engenharia de Domínio da LPS de STI, segundo relatada pelos autores, tem início com a construção de diagramas de caso de uso e de descrição de caso de uso. Os diagramas representam dois atores: estudante e sistema. Em seguida, foi criado o diagrama de

atividades. Depois do diagrama de atividades é desenvolvido o modelo de *features*. A figura 3.6 mostra a representação do modelo de *features* através do diagrama de *features*. A arquitetura da linha é baseada no processo de desenvolvimento de software baseado em componentes UML *Components*. A arquitetura proposta por este processo constitui-se de quatro camadas, duas do lado cliente e duas do lado servidor: Interface de Usuário, Sessão do Usuário, Serviços de Sistema e Serviços de Negócio. Na camada de Serviços de Sistema fica o componente Tutoring e, na camada Serviços de Negócio, estão os seguintes componentes: Modelo de Domínio, Modelo de Estudante e Modelo Pedagógico. Por fim foram criadas descrições semânticas das entidades Domínio, Estratégia Pedagógica, Estudante, LPS e de Decisões (Configurações da LPS) e definido o conector semântico, que realiza a conexão do kernel com os componentes satélites.

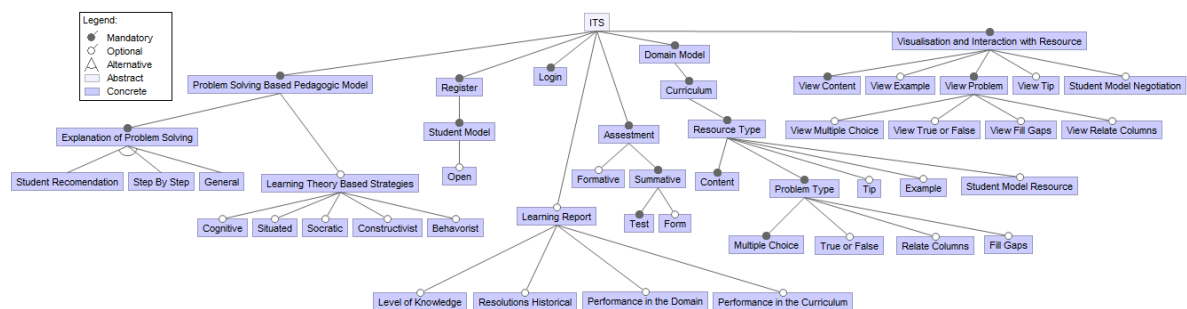


Figura 3.6: Diagrama de Features da LPS de STI. Adaptado de (Silva et al ,2012).

Para a fase de Engenharia de Aplicação o trabalho propõe uma ferramenta de autoria baseada em quatro componentes: *Repository*, *User Registration*, *SPL Registration* e *SPL Instantiation*. O componente *Repository* provê um acesso de alto nível aos dados armazenados. É projetado para trabalhar com bancos de dados relacionais e bancos de dados para ontologias. O componente *User Registration* permite o gerenciamento dos usuários que possuem acesso à ferramenta de autoria. O componente *SPL Registration* permite o cadastro das LPS utilizadas para a geração dos produtos. O componente *SPL Instantiation* é responsável por gerar o produto e é composto por seis componentes: *Login*, *SPL Selection*, *Product Customization*, *Feature Validation*, *Product Generation* e *Product Deployment*. Esses componentes atuam sequencialmente para selecionar a linha, realizar a validação, armazenar a linha na forma de uma ontologia e gerar um produto de software “.war” para finalmente ser implantado (*deployed*) no servidor web.

Com o objetivo de avaliar a viabilidade da proposta é realizada a construção de um tutor inteligente para o domínio de programação. O currículo do tutor abrange a introdução à

programação com conteúdos como entrada e saída de dados, estrutura de programas em C, comandos de atribuição, constantes, expressões aritméticas e expressões lógicas. Cada conteúdo possui sua descrição e um problema atribuído. É atribuído também um modelo pedagógico composto de planos instrucionais. Um diagrama de *features* mais específico, e mais simples foi criado assim como uma nova arquitetura. Uma ferramenta de autoria foi criada e implementa o componente *SPL Instantiation* e seus seis sub-componentes. O produto gerado foi avaliado por uma população de 10 alunos de graduação dos quais 63% classificaram como tendo uma boa ou ótima usabilidade, 64% que a ferramenta permite executar tarefas de maneira rápida e fácil, e 92% classificaram o nível de adequação para usuários experientes e não experientes como ótimo.

3.4 LINHA DE PRODUTOS DE E-LEARNING

Em uma sequência de trabalhos (Chimalakonda et al, 2013; Chimalakonda et al, 2012; Chimalakonda, 2011; Chimalakonda, 2010) os autores descrevem a utilização de técnicas de LPS para a criação massiva e customizada de aplicativos de ensino. Os aplicativos especificamente atendem ao problema da alfabetização de adultos na Índia. Segundo os autores um relatório da UNESCO, de 2012, estima que a quantidade de adultos iliterados na Índia seja cerca de 287 milhões. Além do grande número de pessoas a serem alcançadas subsiste a problemática de que na Índia existem 22 línguas oficiais, e cada uma contém em média 3 dialetos. Ao ser publicado um novo material, o mesmo tem de ser traduzido para 22 línguas diferentes e seu conteúdo multimídia atualizado. Além da língua o conteúdo deve ser regionalizado, respeitando as diferenças culturais de cada localidade. As aplicações desenvolvidas para a alfabetização de adultos na Índia devem seguir os padrões definidos pela *National Literacy Mission* (NLM). A solução proposta pelos autores tem o nome de *Adult Literacy Programme* (ALP).

A iniciativa de adoção da LPS pelos autores foi a extrativa. Na abordagem de adoção extrativa é realizada uma engenharia reversa em produtos existentes que são transformados em uma LPS. Os autores têm como base inicial um pequeno conjunto de softwares dos quais são extraídos ativos e características. As características em comum são utilizadas para formar um arcabouço semelhante ao demonstrado em (Dalmon, 2012). Os ativos são mapeados em uma taxonomia para posterior reaproveitamento. A maioria dos ativos consiste em imagens e sons. Cada aplicação possui milhares de arquivos multimídia. O processo de desenvolvimento das aplicações também foi mapeado. Com o auxílio de uma ferramenta de captura de tela os

processos são capturados, modelados e otimizados. Cada processo é representado através de uma máquina de estados finitos. A figura 3.7 mostra um exemplo da modelagem de um processo de produção. Uma vez mapeados, os processos são divididos em etapas menores e então automatizados através de um *script*. Os *scripts* passam então a fazer parte dos *assets* da linha, e são de grande importância durante o processo de engenharia de aplicação.

A fase de Engenharia de Aplicação de modo semelhante a (Silva et al, 2012) é baseada em uma ferramenta de autoria. Segundo os autores foi alcançada a independência com relação a desenvolvedores e especialistas na construção de uma nova ALP.

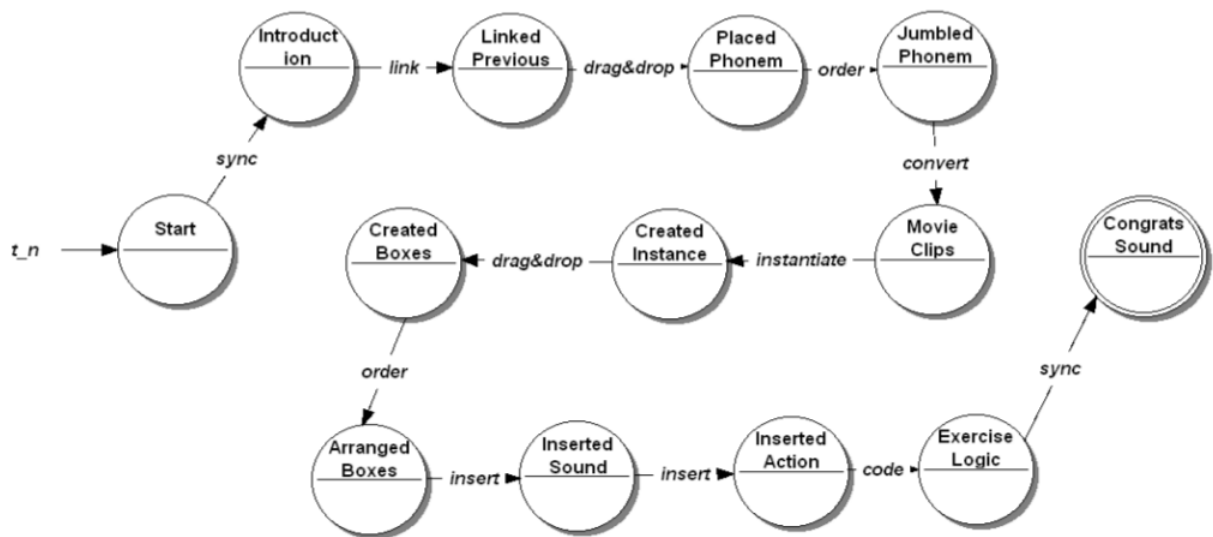


Figura 3.7: Mapeamento de um processo de produção (Chimalakonda, 2010)

Ao utilizar os processos de LPS para a confecção de ALPs os autores clamam ter conseguido uma melhora na performance de 5 pessoas-ano para 5 pessoas-mês. O repositório de ativos criado contém aproximadamente 25 mil arquivos de multimídias armazenados de modo a se obter fácil recuperação. O arcabouço consegue economizar cerca de 40% do esforço e produz uma estrutura organizada e fácil de se reutilizar. Apesar das qualidades do processo adotado existe uma falha no que tange ao mapeamento das variabilidades. Nenhuma metodologia é selecionada para auxiliar de modo formal a seleção de características.

3.5 LINHA DE PRODUTOS PARA JOGOS MÓVEIS

No capítulo 2 é citado o caso bem sucedido de LPS para a construção de sistemas operacionais para aparelhos móveis. Com o aumento da capacidade dos celulares e, em especial, com o aparecimento da *Java Micro Edition (JME)*, a demanda por jogos aumentou. (Zhang et al, 2005) utiliza a abordagem de adoção de LPS extrativa em cima de uma base de quatro jogos do gênero *Role Playing Game (RPG)*. Através da experiência dos especialistas de negócio os pontos em comum e variabilidades foram mapeados e representados em um diagrama de *features*. A figura 3.8 mostra a representação do diagrama de *features*. Com o mapeamento das características 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. São removidas classes desnecessárias, métodos obsoletos, relações de herança redundantes e demais artefatos de código que possuem pouca ou nenhuma funcionalidade. Na fase de Engenharia de Aplicação as meta-variáveis são substituídas com o código das *features* selecionadas. Como resultado final, os autores clamam haver reduzido o tempo de desenvolvimento de um novo jogo de 88 homens-dia para 28 homens-dia.

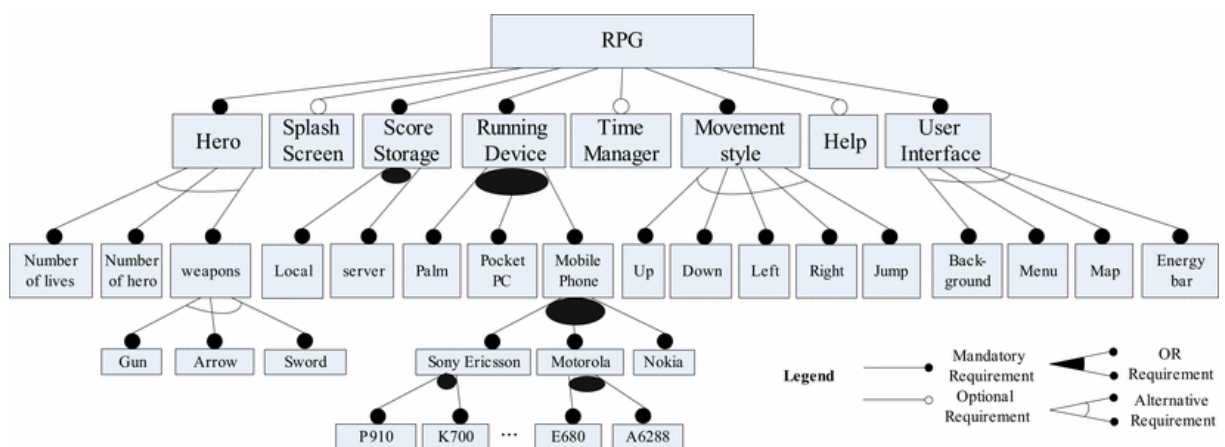


Figura 3.8: Diagrama de Features de domínio de jogos móveis do gênero RPG.

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 conta com restrições de interface e de capacidade de memória e poder de processamento. Embora atualmente os dispositivos móveis estejam próximos de alcançar os computadores pessoais em termos de performance os modelos mais econômicos ainda enfrentam as mesmas restrições de anos atrás. Cada aparelho tem um tamanho de tela, intensidade de brilho e esquema de cores. Diferentes dispositivos possuem sistemas operacionais diferentes e

importantes componentes de hardware. Em um cenário como o proposto acima a LPS forma uma oportuna e satisfatória solução para atender as diversas características de demanda. Os autores, no entanto, apontam que a solução de LPS não tem sido utilizada extensivamente no desenvolvimento móvel por causa do rápido ciclo de desenvolvimento que desencoraja uma adoção apoiada por metodologias e processos e a aversão de pequenas e médias empresas a aceitar riscos. O que os autores sugerem para aumentar a segurança para adoção de LPS para plataformas móveis é a criação de padrões de arquitetura.

Assim como os conhecidos padrões de projeto os padrões de arquitetura são independentes de plataforma e tecnologia. O *template* para o padrão de arquitetura tem os seguintes campos: Nome, Contexto, Solução, Variantes e Padrões Relacionados. Como exemplo os autores descrevem o padrão de arquitetura “*Contents-adaptable*”. Este padrão é utilizado quando uma aplicação *mobile standalone* necessita ter seu conteúdo alterado. A solução é desenvolver um componente de gerência de conteúdo que, através da leitura de pacotes que contém os novos artefatos, cuja aquisição é realizada automaticamente através de *download*, realiza as alterações. Essa solução é amplamente utilizada pela indústria de jogos atualmente, e é popularmente conhecida pelos usuários como DLC (*Downloadable Content*). Os DLCs constituem uma importante parcela na economia da indústria de jogos. As variações são a utilização do carregamento automático de classes. Os padrões relacionados são *Factory* e *MVC*. Os autores clamam terem sido bem sucedidos no desenvolvimento de produtos comerciais reais utilizando os padrões de arquitetura descritos. O diagrama de *features* da LPS utilizado é mostrado na figura 3.9. Segundo os autores, o mapeamento dos padrões de arquitetura para as *features* foi “fácil e intuitivo” possibilitando a geração de um código base (*skeleton code*).

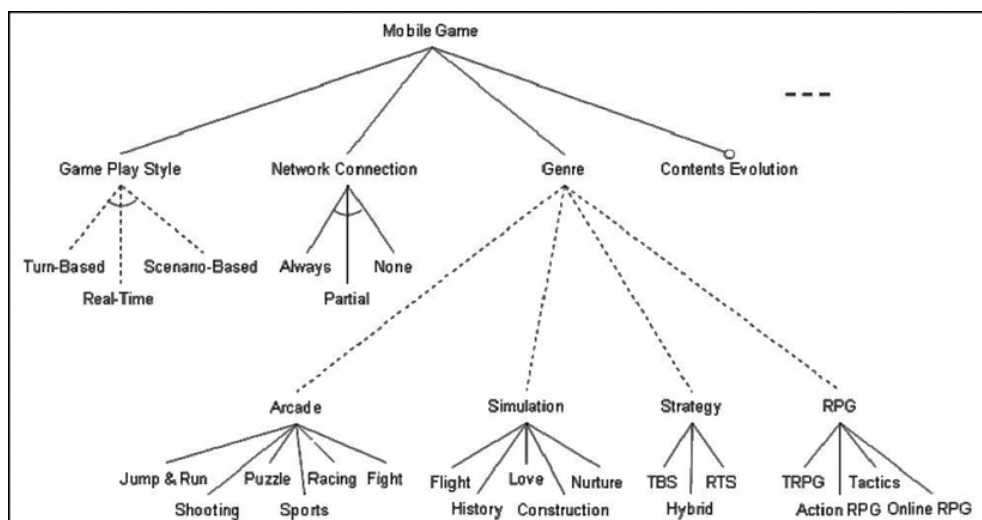


Figura 3.9: Diagrama parcial de *features* do domínio de jogos móveis.

3.6 CONSIDERAÇÕES FINAIS DO CAPÍTULO

Neste capítulo foram apresentados três trabalhos, uma tese de doutorado e duas dissertações de mestrado, que propõem a utilização de LPS para a produção de software educacional, e um último trabalho propondo uma LPS voltada para a produção de jogos *mobile*. O primeiro trabalho propôs uma linha para Módulo de Aprendizagem Interativos. A Engenharia de Domínio entregou requisitos não funcionais e um diagrama de *features* contendo pelo menos 28 entidades. A Engenharia de Aplicação é realizada através da instanciação de um arcabouço e as especificidades da aplicação são adicionadas manualmente. O segundo trabalho propôs uma linha para Tutores Inteligentes. A fase de Engenharia de Domínio produziu diagramas de casos de uso, diagramas de atividade, uma arquitetura de quatro camadas e um diagrama de *features* com 48 entidades. A fase de Engenharia de Aplicação consiste de uma ferramenta de autoria com quatro componentes. O terceiro trabalho propôs uma LPS de Programa de Alfabetização de Adultos (ALP). A Engenharia de Domínio consistiu de uma padronização dos ativos e do mapeamento, na forma de diagramas de estado, e automatização de partes do processo de desenvolvimento através de *scripts*. A Engenharia de Aplicação é realizada, assim como no primeiro trabalho, através de uma ferramenta de autoria.

O que o BROAD-PLG propõe, e que se destaca dos demais, é apresentado a seguir:

- A Análise de Domínio culminou na separação de assuntos em três interesses distintos: jogos, jogos educacionais e gamificação. Cada interesse tem seu próprio modelo de *features*. O modelo de jogos possui trinta e quatro entidades, o modelo educacional vinte entidades e o modelo de gamificação trinta entidades. A divisão em interesses ajuda a simplificar o processo de escolha de *features*. Existe também uma relação sinérgica entre eles. A modelagem das características de jogos fornecem mecânicas básicas que ajudam a promover um resultado quantificável em uma disputa. A modelagem de características educacionais insere um conteúdo para ser aprendido e opcionalmente mecanismos de avaliação, exercício e metadados. A modelagem de características de gamificação possibilita a inserção de elementos que promovem a motivação e subsequentemente o engajamento.
- A LPS permite a criação de até oito tipos de aplicação: um *stub* genérico, um jogo, um objeto de aprendizagem, um sistema genérico de gamificação, um jogo educacional, um jogo com elementos de engajamento, um objeto educacional com elementos de

motivação e um jogo educacional com elementos de engajamento. Os modelos são independentes, e a modificação de um não interfere com o outro. De modo semelhante, o acréscimo de um novo modelo não deve interferir com os demais. A mudança dos modelos, no entanto, se propaga para a fase de Engenharia de Aplicação e modifica o componente *Factory Connector*.

- Na fase de Engenharia de Aplicação o componente *Factory Connector* insere dentro do arcabouço anotações para permitir a realização de testes de unidade e de comportamento.
- Diferente dos trabalhos citados a arquitetura da LPS proposta não especifica nenhuma tecnologia específica ou restrições não funcionais. A arquitetura comporta cinco camadas com responsabilidades distintas que não possuem restrições para implementação. Por causa da ampla variedade de produtos possíveis é interessante manter o alto nível para dar mais liberdade aos desenvolvedores que atuam nos componentes da linha.
- A última capacidade da linha é a possibilidade de aproveitar motores de jogos produzidos por terceiros. Como descrito na seção 4.1 motores de jogos são artefatos importantes que realizam diversas funcionalidades comuns. O reaproveitamento dos motores de jogo requer uma integração com o *Factory Connector*.

A tabela 3.1 realiza uma comparação entre os trabalhos relacionados descritos acima e o BROAD-PLG.

Tabela 3.1 – Comparação de características entre os trabalhos relacionados

	RPG	AGM	ALP	ITS	iMA	BROAD-PLG
Ano	2005	2009	2010	2011	2012	2014
Desktop		✓	✓		✓	✓
Web				✓	✓	✓
Mobile	✓					✓
Componentes				✓	✓	✓
Autoria			✓	✓		
Arcabouço					✓	✓
Educacional			✓	✓	✓	✓
Jogos	✓	✓			✓	✓
Gamificação						✓

4 BROAD-PLG: MODELO COMPUTACIONAL PARA CONSTRUÇÃO DE JOGOS EDUCACIONAIS

4.1 INTRODUÇÃO

O projeto BROAD (Rezende et al., 2013) busca a adoção de tecnologias como ontologias, serviços web semânticos, agentes e workflow para a construção de uma arquitetura para composição e sequenciamento de OA. O projeto avança em questões relativas a trabalhos anteriores como em arquiteturas como CelOWS (Matos, et al., 2009), SASAgent (Mendes et al., 2011), ComposerScience (Silva et al., 2011), BROAD (Braga et al., 2011) (Campos et al., 2012), em repositórios semânticos como (Santos et al., 2008) e objetos de aprendizagem (Campos et al., 2011) (Souza et al., 2010).

Considerando que a composição e sequenciamento de OA é uma atividade chave para o compartilhamento e reuso de conteúdos educacionais (Damaševičius and Štuikys, 2009), essa proposta utiliza conceitos de LPS para viabilizar o reuso sistemático de OAs do tipo jogos educacionais. Entretanto, o projeto BROAD-PL (BROAD Product Line) (Castro et al., 2012), permite a personalização dos objetos de aprendizagem, através de uma arquitetura para LPS apoiada semanticamente em uma rede de ontologias, através do mapeamento dos elementos ontológicos (termos, restrições e relacionamentos) para as features especificadas na LPS.

A infraestrutura, denominada BROAD-PL (Figura 4.1), foi definida através da abordagem de Linha de Produtos de Software e da utilização de uma arquitetura orientada a serviços. Considerando-se os mecanismos que apoiam a construção de OA e que facilitam a integração com Ambientes Virtuais de Aprendizagem, a abordagem de LPS pode oferecer um suporte sistemático para a reutilização a partir das características e preferências do aluno.

O módulo Gerente de Processos e Composição é responsável pela criação de sequências/composição de objetos de aprendizagem para atender às necessidades específicas dos alunos. Neste contexto, diferentes processos poderão ser criados e deverão ser gerenciados.

O módulo Gerente de Variabilidades trata da análise das características que diferenciam os objetos de aprendizagem em contextos específicos. Esta análise auxilia o

projeto de OAs identificando “quando” e “como” as características são associadas ao objeto para atender às necessidades dos alunos (Lee e Kang, 2004). Como resultado, esses objetos poderão compor os processos para a infraestrutura BROAD-PL. Este gerente é responsável pela associação da rede de ontologias com o modelo de características. Esta associação é de suma importância, uma vez que a semântica de conteúdos específicos e os diferentes contextos semânticos do OAs somente estão detalhados na rede de ontologias. O modelo de características possui uma semântica específica que é apresentar as variabilidades possíveis no domínio. Unindo as duas visões, ou seja, a variabilidade possível no domínio, dada pelo modelo de características, e a semântica dos conteúdos e contextos dos OAs, dada pela rede de ontologias, o desenvolvimento e/ou o reuso dos OAs poderão ser facilitados.

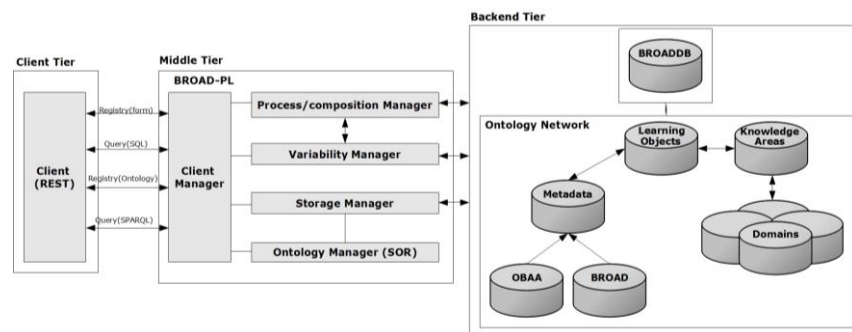


Figura 4.1: Visão Geral Arquitetura BROAD-PL (Campos et al., 2012).

O núcleo de artefatos, representado pela camada *backend*, contém não apenas os objetos de aprendizagem e os seus metadados, mas também fornece as direções para identificá-los e derivar outros artefatos a partir das variabilidades, das redes de ontologias e das necessidades específicas dos usuários. O módulo Gerente de Armazenamento é o responsável pela gerência de versões dos objetos, bem como das configurações a eles associadas. Adicionalmente, a rede de ontologias oferece o suporte para as atividades de busca e recuperação de objetos no núcleo de artefatos. Cabe ao módulo Gerente de Ontologias realizar as buscas na rede de ontologias disponível. Estas atividades contam com o auxílio do módulo Gerente de Variabilidades o qual associa as features de cada OA às informações a eles associadas, por exemplo, “como” e “quando” elas poderão ser aplicadas ao contexto que está sendo trabalhado. Este módulo interage com o Gerente de Armazenamento com o objetivo de recuperar as informações associadas, por exemplo, ao contexto (tecnológico) de utilização do objeto.

A camada Cliente é responsável pela interação com os usuários e implementa o padrão Facade. Seu propósito é fornecer uma interface única com os clientes do sistema, de

forma que eles não tenham acesso à estrutura interna do BROAD-PL.

A partir de um modelo de processo (educacional), do modelo de domínio e dos atributos de qualidade associados a este processo, objetos serão selecionados ou compostos para atender às demandas específicas dos usuários. Portanto, existirão relações entre os processos, os modelos e os objetos que deverão ser persistidas no repositório da linha de produto.

A ferramenta Easy Tutorial E-learning (EasyT) foi um primeiro passo para a construção de objetos de aprendizagem no contexto do BROAD-PL (Castro et al., 2012). É um template que visa facilitar a construção e integração de OAs com repositórios de OA. Trata-se de um Tutorial Hipermedia - documento hipertexto dinâmico que permite estabelecer ligações entre páginas e embutir recursos de páginas HTML. O EasyT é formado por uma ou mais páginas distintas. Cada página, além do texto formatado, pode conter qualquer tipo de mídia externa - imagem, áudio, vídeo ou animação – e também pode conter ligações para outros OAs. A ferramenta permite a geração de OAs específicos e a integração automática com repositórios de OA. Os metadados do EasyT são carregados uma única vez pelo autor e ficam armazenados no tutorial. Quando se deseja integrar o tutorial a um repositório, os metadados serão cadastrados automaticamente através do mecanismo presente na ferramenta.

Através de um índice ordenado alfabeticamente por título das páginas, o usuário poderá acessar uma página e visualizá-la. Portanto, o índice contém as ligações para qualquer página do tutorial. Cada página possui uma identificação e um conjunto de metadados de OA que descrevem informações, tais como, título, autor, idioma, versão etc. Esses metadados são apresentados no momento da visualização de cada página. Além dos metadados de cada página, o tutorial também possui seus próprios metadados. O tutorial possui um padrão de metadados que é compatível com os mecanismos de busca do projeto BROAD.

O mecanismo de interoperabilidade presente na ferramenta permite que repositórios e outras aplicações recuperem os metadados do OA de forma programável. Dessa forma, o esquema do tutorial permite que sistemas com diferentes plataformas consigam integrá-lo. Quando o EasyT é cadastrado no repositório do BROAD, o mecanismo permite que todas as páginas sejam automaticamente cadastradas como se fossem novos OAs. Dessa maneira, temos como recuperar qualquer parte do tutorial no repositório. A ferramenta é empacotada por um web service que disponibiliza os metadados do tutorial e suas páginas.

A seguir é descrita a infraestrutura BROAD-PLG, uma extensão da arquitetura BROAD-PL, onde um modelo computacional é apresentado para a construção de objetos de aprendizagem do tipo jogos educacionais.

4.2 BROAD-PLG – BROAD PRODUCT LINE FOR EDUCATIONAL GAMES

4.2.1 FEATURES

Nesta seção são apresentadas as características que capturam a variabilidade e os pontos comuns entre os produtos da linha. As características foram divididas em três modelos distintos. Cada modelo mapeia um aspecto específico da linha. As divisões são: características específicas de jogos, características educacionais e características de gamificação. A vantagem de se dividir o modelo de características em três subdomínios é a possibilidade de se obter diferentes tipos de produtos de forma independente, altamente coesa e de baixo acoplamento. A tabela 4.1 mostra as possíveis categorias de produtos que se podem obter através da LPS. Outra vantagem é a de que os modelos de características, atuando de forma autônoma, podem ser modificados sem impactar, ao menos ao nível de modelo, as características de outros subdomínios.

Tabela 4.1 – Possíveis combinações e seus produtos na seleção de features.

Jogos	Educacional	Gamification	Produto
			O produto gerado é apenas um arcabouço vazio 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 que contém elementos especialmente projetados para se obter engajamento.

É importante destacar que é ausente a pretensão, neste trabalho, de 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 em características foi baseado no estudo da literatura acadêmica e de

trabalhos de profissionais da indústria de reputação reconhecida e especialistas neste campo de conhecimento. O levantamento realizado pode e deve servir como referência para autores e profissionais que desejarem realizar trabalhos nessa área, mas deve-se ter em mente a liberdade de adaptação para as especificidades dos projetos a serem realizados.

4.2.1.1 Características de Jogos

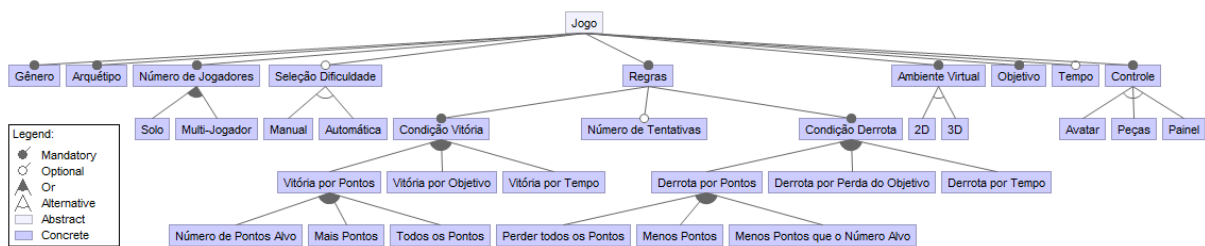


Figura 4.2: Diagrama de Características de Jogos

A figura 4.2 mostra o diagrama de *features* de jogos. O mapeamento das características de jogos tem como princípio a busca por mecânicas de alto nível e independência de hardware, plataforma e protocolos. O detalhamento das características é tornado explícito a seguir:

- **Gênero:** O gênero de um jogo delimita elementos gerais que servem de guia para o projeto do jogo. É uma das primeiras características a serem pensadas na criação do jogo. Exemplos de gêneros de jogos: Ação, Estratégia e RPG (Role Playing Game).
- **Arquétipo:** O arquétipo de um jogo é um conjunto específico de regras reutilizável, utilizado como modelo, e que é retirado de algum outro jogo. No original grego arquétipo significa o primeiro de um tipo. Quando um jogo apresenta uma inovação muito grande e faz um grande sucesso ele se torna base de um arquétipo. Exemplos de arquétipo: *Tower Defense*, *Plataform* e FPS (*First Person Shooter*).
- **Número de Jogadores:** Esta característica define a quantidade de jogadores suportada. A característica Solo se refere ao suporte de apenas um jogador. A característica Multi-Jogador permite que vários jogadores participem de uma mesma partida simultaneamente de modo presencial ou online.
- **Seleção de Dificuldade:** Não existe um consenso, entre os jogadores, sobre qual é a

dificuldade ideal de um jogo. Enquanto alguns preferem ficar horas para vencer determinado desafio outros preferem superá-lo na primeira tentativa. Por esse motivo é desejável haver uma seleção de dificuldade para poder agradar ao maior número possível de jogadores. A seleção da dificuldade pode ser manual, onde o jogador escolhe a dificuldade, ou automática onde o jogo seleciona a dificuldade baseando-se no desempenho do jogador (Sylvester, 2013).

- **Regras:** As regras formam o conjunto de características que permitem que o jogo tenha um resultado quantificável.
- **Condição de Vitória:** O conjunto de características que definem certas condições para que o jogador ou jogadores sejam vitoriosos. A contraparte Condição de Derrota define situações que, caso ocorram, implicam na derrota do jogador ou jogadores.
- **Vitória por Pontos:** Pontos representam abstrações de valores que são utilizados para recompensar o jogador que conseguir realizar uma determinada tarefa. Por exemplo, no futebol, 1 ponto é fornecido ao time do jogador que conseguir colocar a bola dentro da rede do time adversário. Existem três variações de regras nas quais é possível vencer utilizando-se pontos: alcançar um determinado número de pontos alvo, fazer mais pontos que o adversário e conquistar todos os pontos possíveis em uma partida. Do mesmo modo, é possível perder através de um sistema de pontos de três maneiras: não ser capaz de alcançar uma determinada quantidade de pontos, não conseguir fazer mais pontos que o adversário e não estar de posse de nenhum ponto. Utilizando-se novamente o futebol como exemplo para vencer é necessário fazer mais pontos que o adversário e, para perder, basta fazer menos pontos que o adversário.
- **Número de tentativas:** Sendo um mecanismo comum a muitos jogos a característica Número de Tentativas define a política de chances que o jogador terá para alcançar determinado objetivo. Se o jogador não conseguir satisfazer o objetivo e esgotar o número de tentativas ele terá de recomeçar o jogo, pois sua atuação indica que seu nível de proficiência não é o suficiente.
- **Ambiente virtual:** O ambiente virtual define a fronteira na qual o jogador irá exercer

sua influência. O ambiente virtual pode ser representado em duas ou três dimensões.

- **Objetivo:** Metaforicamente o objetivo pode ser comparado à linha de chegada em uma corrida. Ao alcançar o objetivo o jogador é declarado vencedor. Um objetivo pode ser composto por objetivos menores.
- **Tempo:** O tempo é uma característica opcional, mas importante para ajudar a conseguir um resultado mensurável. Afinal, se a partida não tiver fim o jogo não terá um resultado definido.
- **Controle:** A característica controle define o modo pelo qual o jogador irá interagir com o ambiente virtual. Ao utilizar a característica Avatar o jogador assume controle de um personagem que através de um conjunto de ações interage com o ambiente. Exemplos que jogos que usam avatares: Super Mário Bros, *Street Fighter* e *Need for Speed*. A característica Peças permite que o jogador controle múltiplas representações de personagens. As peças normalmente possuem menos detalhamento e elementos que os personagens controlados com a característica Avatar. A característica peças é comumente utilizada em jogos de estratégia e tabuleiros. Exemplo de jogos que utilizam a característica peças: *Starcraft*, Xadrez e *Final Fantasy Tactics*. A última característica dentro de controle é Painel. Ao utilizar a característica o jogador assume controle sobre um painel administrativo com múltiplos parâmetros onde suas ações determinam o resultado do jogo. Exemplos de jogos que utilizam a característica Painel como método de controle: *Soccer Manager*, *Civilization* e *Cut The Rope*.

4.2.1.2 Características Educacionais

A figura 4.3 mostra o diagrama de *features* das características educacionais. A característica mais proeminente deste diagrama é Metadados. Metadados segundo (Giunchiglia et al, 2013) são dados que definem um nível maior de dados. Na LPS proposta por este trabalho são selecionados três padrões de metadados: LOM, OBAA e BROAD. Para fins da primeira versão não foram instanciados os padrões para jogos educacionais, e sim, utilizados os conjuntos de metadados como características dos objetos de aprendizagem.

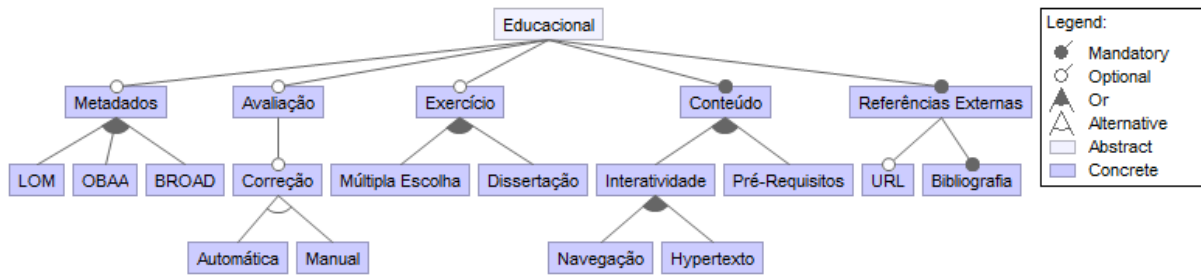


Figura 4.3: Diagrama de Características Educacionais

As outras características presentes no modelo de *features* educacional são descritas abaixo:

- **Avaliação:** Essa característica determina a possibilidade de que o sistema possa avaliar o conhecimento do aluno. Se selecionada o sistema obrigatoriamente terá que permitir a correção da avaliação. A correção pode ser alternativamente manual ou automática.
- **Exercício:** Quando selecionada, essa característica confere ao sistema a propriedade de consolidar o conhecimento do aluno através de exercícios. Os exercícios podem ser tanto de múltipla escolha quanto dissertativos.
- **Conteúdo:** O conteúdo é o núcleo das características educacionais. É através do conteúdo que o aluno conseguirá absorver uma nova unidade de informação. Para acessar o conteúdo pode ser necessário apresentar conteúdos que formem uma base para absorvê-lo. Essa necessidade é definida pela característica Pré-Requisitos. O Conteúdo também pode oferecer algum grau de interatividade, opcionalmente, tanto através de navegação, quando em formato de vídeo e texto, quanto através da apresentação de *hyperlinks* para navegação interna.
- **Referências Externas:** Embora o conteúdo possa ser auto-contido é importante mostrar de onde foi retirado. Com esse objetivo a característica fornece a possibilidade de se apresentar a bibliografia e *hyperlinks* para o acesso a conteúdos correlacionados externos.

4.2.1.3 Características de Gamificação

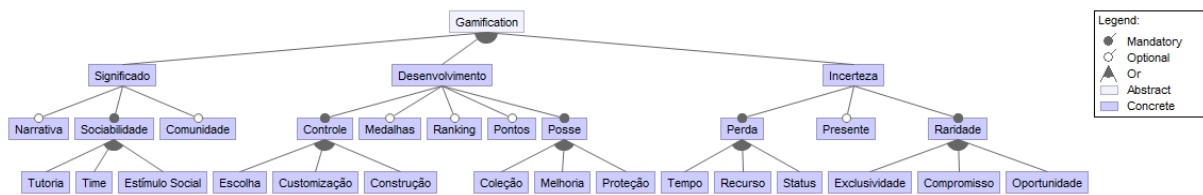


Figura 4.4: Diagrama de Características de Gamificação.

A figura 4.4 mostra o diagrama de características de gamificação. As características de gamificação tem como objetivo comum a fomentação da motivação e a promoção do engajamento através de mecanismos práticos (Kapp, 2012).

- Significado:** A característica 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, expressa geralmente em texto, 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. Um mecanismo que envolva a característica comunidade pode, por exemplo, consistir de uma doação a uma entidade carente a cada 100 atividades concluídas.
- Sociabilidade:** A característica 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, enquanto está utilizando o sistema, 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*) deve ser mais escarça do que a distribuição de pontos, e serve para guiar o jogador em direção à realização de tarefas que o projetista da medalha deseja que o usuário realize. Algumas plataformas de ensino à distância implementam o mecanismo de *badges* como, por exemplo, o Moodle. Distribuindo-se pontos e medalhas é possível organizar os usuários em *rankings*. O objetivo de formar um *ranking* é estimular a competição, porém jogadores que estiverem muito abaixo no *ranking* podem se sentir desmotivados a continuar persistindo na conquista de posições.

- **Controle:** 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:** A característica que consegue despertar no usuário o sentimento de preocupação com relação ao *status* de elementos que estejam presentes apenas no sistema é denominada Posse. 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. Um exemplo de implementação da característica de proteção é através do uso de um bicho de estimação virtual que deve ser cuidado de tempos em tempos e, caso negligenciado, ele pode ficar doente e até morrer.
- **Incerteza:** A 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.

- **Perda:** Aversão à Perda é um bem documentado 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:** O último grupo de características 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.

4.2.2 ARQUITETURA BROAD-PLG

Nessa seção é apresentada a visão geral da arquitetura da LPS para Jogos Educacionais. 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 4.5 mostra a visão geral da arquitetura. As camadas da arquitetura são descritas a seguir:

- **Client Tier:** a camada 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:** essa camada representa os serviços do sistema e de negócio. É composta de Domain Tier e Application Tier.
 - **Domain Tier:** a camada Domain Tier é responsável por providenciar um meio do usuário selecionar a configuração do produto através das *features* disponíveis na LPS. 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.

- **Application Tier:** 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 feito sob medida que consegue se comunicar com os dois tipos de motores da LPS, 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:** A camada Backend Tier armazena os ativos da LPS 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.

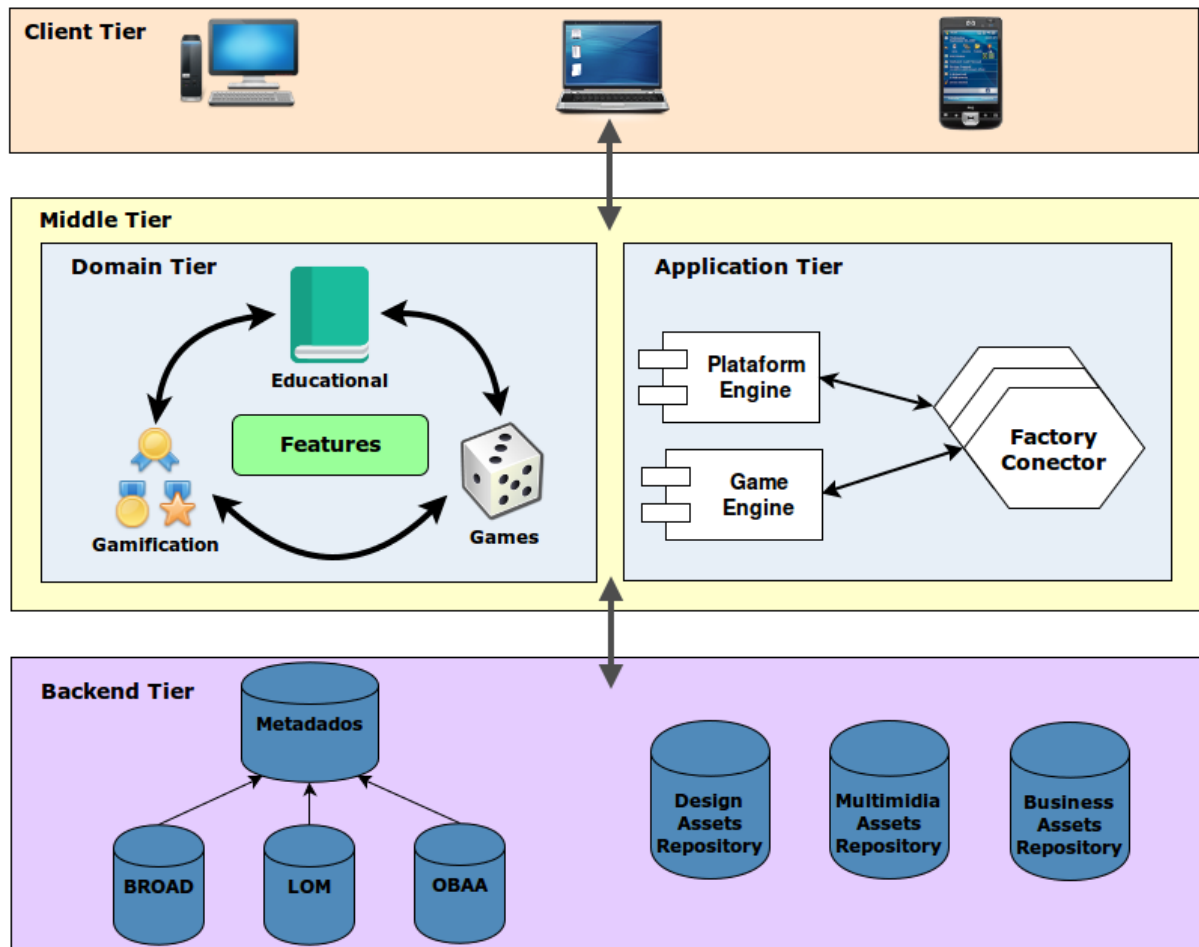


Figura 4.5: Arquitetura do BROAD-PLG.

4.2.3 MODELO COMPUTACIONAL

Para avaliar a proposta é implementada a arquitetura sugerida 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. Na avaliação foi feita um motor de plataforma para HTML5 que está dentro da aplicação que instancia o arcabouço. São criados três produtos na avaliação: *Goal Keeper*, *Monty Cup* e *Game Work*. Os três produtos ressaltam, respectivamente, características de jogos, educacionais e de gamificação sem pertencer exclusivamente a uma determinada classe, demonstrando a versatilidade da LPS e a sinergia entre os três domínios.

Nas seções seguintes são detalhadas as camadas cliente, a camada de meio e a engenharia de aplicação dos produtos.

4.2.3.1 Implementação das camadas cliente e de domínio

A camada cliente foi implementada como um *front-end* web HTML 5 com a parte dinâmica *client-side* programada com javascript utilizando-se as bibliotecas jQuery e jQuery-ui. A biblioteca jQuery fornece uma grande quantidade de métodos para a manipulação do DOM (*Data Object Model*) e chamadas assíncronas de conteúdo. A biblioteca jQuery-ui é uma extensão da jQuery e fornece métodos para a geração de temas, efeitos e transformações dos elementos HTML. O ambiente de seleção de *features* foi implementado utilizando-se a linguagem Processing. A linguagem Processing possui um IDE próprio com suporte tanto para a programação desktop quanto para a programação web. No ambiente desktop o código é compilado e executado utilizando-se a JVM (*Java Virtual Machine*). No ambiente para web o código é executado através biblioteca ProcessingJs, uma biblioteca que emula o comportamento do ambiente desktop. A figura 4.6 mostra a execução da IDE do Processing.

A seleção das características respeita as restrições definidas no trabalho original de (Kang et al, 1999). As características são associadas a quatro tipos: mandatórios, OU, opcionais e alternativas. A divisão em tipos faz referência às restrições de seleção de cada característica. Características mandatórias são sempre selecionadas e não podem ser desselecionadas. Características do tipo OU podem ser selecionadas ou não juntamente com seus “irmãos”. Caso o pai de uma característica OU seja selecionado pelo menos um dos filhos também deve ser selecionado. Caso uma característica OU seja selecionada então o pai obrigatoriamente deve ser selecionado caso já não esteja. A regra da seleção do pai vale para todos os tipos de características. Caso uma característica alternativa seja selecionada, se houver outro irmão selecionado, o outro irmão previamente selecionado é desselecionado. Caso a característica seja opcional a seleção ou desseleção não afeta irmãos ou pai. Se o pai que possui filhos com características do tipo OU, ou alternativa, é desselecionado, então os filhos são automaticamente desselecionados.

Na *interface* da camada cliente, após a seleção das características de uma árvore o usuário pode passar para as próximas árvores selecionando os ícones nos campos superiores esquerdo e direito conforme pode ser visto nas figuras 4.7, 4.8 e 4.9. 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

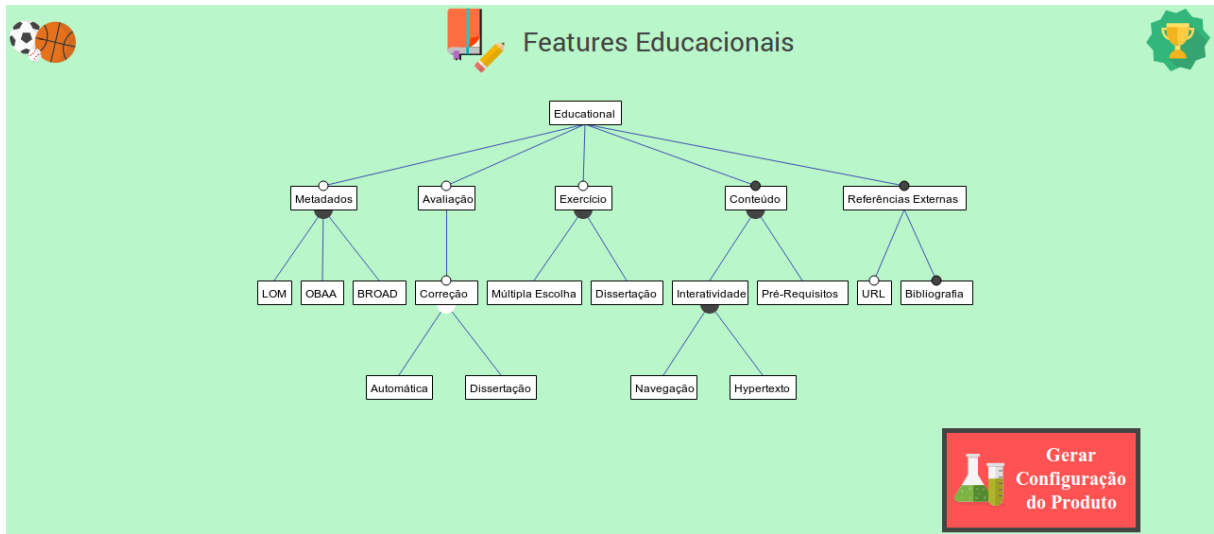


Figura 4.8: Interface *web* para seleção de características educacionais.

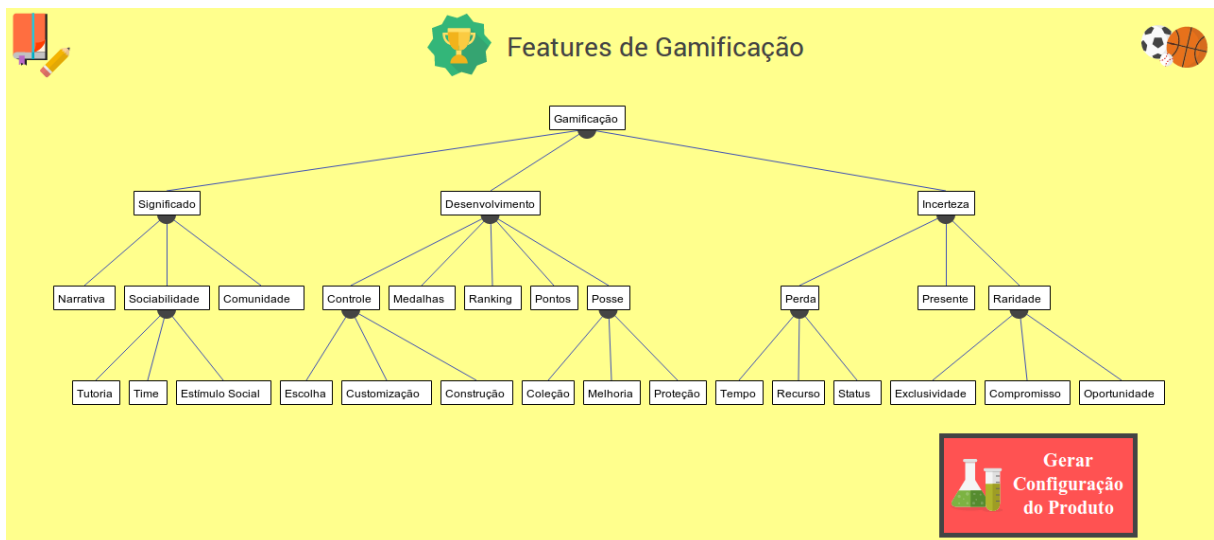


Figura 4.9: Interface *web* para seleção de características de gamificação.

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```

<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<featureModel chosenLayoutAlgorithm="1">
  <struct>
    <and abstract="true" mandatory="true" name="Jogo">
      <or mandatory="true" name="Número de Jogadores">
        <feature mandatory="true" name="Solo"/>
        <feature mandatory="true" name="Multi-Jogador"/>
      </or>
      <alt name="Seleção Dificuldade">
        <feature mandatory="true" name="Manual"/>
        <feature mandatory="true" name="Automática"/>
      </alt>
      <feature mandatory="true" name="Gênero"/>
      <and mandatory="true" name="Regras">
        <or mandatory="true" name="Condição Vitória">
          <or mandatory="true" name="Vitória por Pontos">
            <feature mandatory="true" name="Número de Pontos Alvo"/>
            <feature mandatory="true" name="Mais Pontos"/>
            <feature mandatory="true" name="Todos os Pontos"/>
          </or>
          <feature mandatory="true" name="Vitória por Objetivo"/>
          <feature mandatory="true" name="Vitória por Tempo"/>
        </or>
        <feature name="Número de Tentativas"/>
        <or mandatory="true" name="Condição Derrota">
          <or mandatory="true" name="Derrota por Pontos">
            <feature mandatory="true" name="Perder todos os Pontos"/>
            <feature mandatory="true" name="Menos Pontos"/>
          </or>
          <feature mandatory="true" name="Menos Pontos que o Número Alvo"/>
        </or>
        <feature mandatory="true" name="Derrota por Perda do Objetivo"/>
        <feature mandatory="true" name="Derrota por Tempo"/>
      </and>
      <feature mandatory="true" name="Objetivo"/>
      <feature name="Tempo"/>
    </and>
    <alt mandatory="true" name="Ambiente Virtual">
      <feature mandatory="true" name="2D"/>
      <feature mandatory="true" name="3D"/>
    </alt>
  </struct>
</featureModel>

```

Figura 4.10: XML gerado com a configuração do produto.

4.2.3.2 Implementação da Camada de Aplicação

A camada de aplicação é 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. A figura 4.11 mostra a aplicação gerando o arcabouço de um produto com o motor de plataforma para HTML 5.

```
geva@geva-notebook:~$ /usr/lib/jvm/java-8-oracle/jre/bin/java -jar factory.jar -MontyCup
Iniciando a leitura do arquivo de configuração do produto... Pronto!
Validando a configuração do produto... Pronto!
Carregando a plataforma de jogo... Plataforma HTML5 carregada.
Construindo arcabouço... .
|----MontyCup
| |----css
| | |----img
| | |----MontyCup.css
| | |----img
| | |----js
| | |----libs
| | | |----angular
| | | | |----angular-animate.js
| | | | |----angular-cookies.js
| | | | |----angular-csp
| | | | |----angular-loader.js
| | | | |----angular-mocks.js
| | | | |----angular-resource.js
| | | | |----angular-route.js
| | | | |----angular-sanitize.js
| | | | |----angular-scenario.js
| | | | |----angular-touch.js
| | | | |----angular.js
| | | |----jquery
| | | |----jquery.js
| | |----MontyCup.js
| |----index.html
Inserindo arquivos de teste... Pronto!
Testando o o deploy... Pronto!
Arcabouço instanciado com sucesso!
```

Figura 4.11: Criação do arcabouço utilizando a camada do meio.

4.2.4 AVALIAÇÃO DA PROPOSTA

A avaliação da proposta da arquitetura BROAD-PLG se deu por meio da geração de três produtos: Goal Keeper, Monty Cup e Game Work. Após a construção da primeira versão do BROAD-PLG, foram planejadas as provas de conceitos seguindo modelos que contemplassem a avaliação da viabilidade da proposta. Os três produtos ressaltam, respectivamente, características de jogos, educacionais e de gamificação sem pertencer exclusivamente a uma determinada classe, demonstrando a versatilidade da linha e a sinergia entre os três domínios.

4.2.4.1 Goal Keeper

O primeiro produto tem como objetivo testar a viabilidade da Engenharia de Aplicação da LPS. Goal Keeper é uma releitura do clássico jogo de arcade Pong. Dois adversários controlam barreiras e tentam evitar a derrota rebatendo a bola e ao mesmo tempo tentando induzir o oponente a um erro deixando a bola invadir seu campo além da barreira. A figura 4.12 mostra a seleção de características na interface web. O arcabouço forneceu a estrutura básica necessária, e o esforço restante é apenas o de adicionar os arquivos de mídia e as regras específicas de negócio. As figuras 4.13 e 4.14 mostram o jogo em execução.

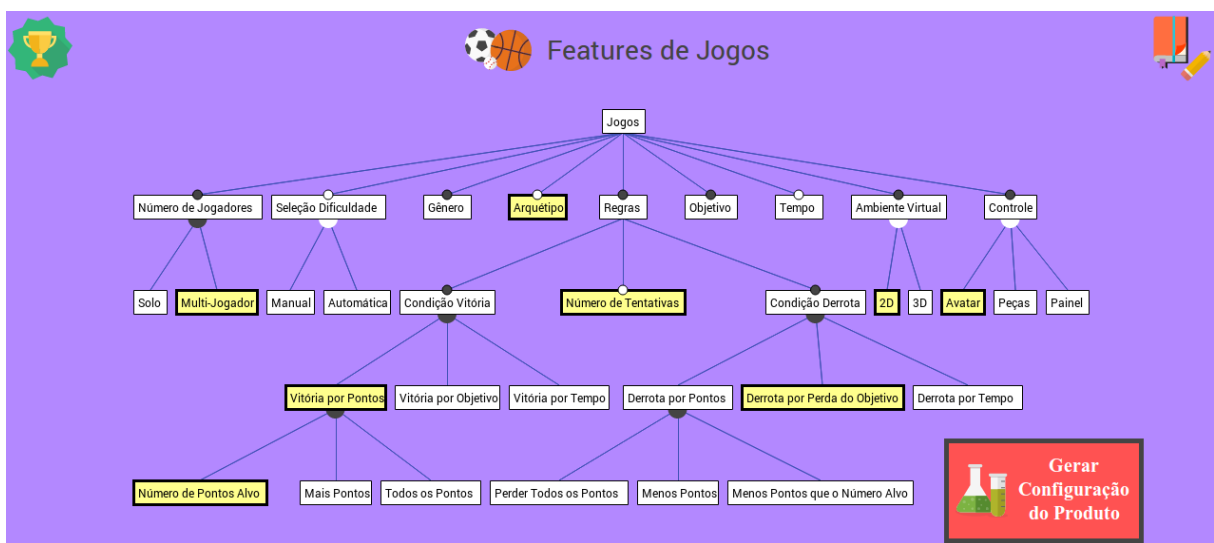


Figura 4.12: Seleção de características para o jogo *Goal Keeper*.



Figura 4.13: O jogo Goal Keeper em execução.



Figura 4.14: Término do jogo Goal Keeper.

4.2.4.2 Monty Cup

O segundo produto explora a junção de características de jogos e produtos educacionais. O conteúdo a ser ensinado é uma introdução lúdica ao quão contraintuitivo pode ser o cálculo de probabilidades.

O problema de Monty Hall, nomeado em referência a um famoso apresentador estadunidense, foi publicado pela primeira vez em 1975 na revista *American Statistician*. Originalmente o problema simula um *Game Show* na qual o jogador deve escolher uma dentre três portas. Duas portas contém uma cabra e uma porta contém um carro novo. O objetivo é tentar adivinhar qual porta contém o carro. Ao escolher uma porta ela não é aberta de imediato. O apresentador, que sabe o conteúdo de todas as portas, abre uma porta que não contém o carro. Em seguida, ele pergunta ao jogador se ele quer trocar de porta. O problema, no entanto, consiste de responder a pergunta: "É mais vantajoso trocar de porta, permanecer com a porta escolhida ou não faz diferença?".

Intuitivamente é tentador dizer que não há diferença já que as portas têm probabilidades iguais de conter o prêmio: $1/3$. Escolhendo entre as duas portas restantes também obteríamos probabilidades iguais para cada porta: $1/2$. Entretanto, é mais vantajoso sempre trocar de porta com uma chance de conseguir o carro de duas chances em três. A explicação é a seguinte: a chance de errar o grande prêmio é de $2/3$. Supondo que a porta escolhida seja a errada, o apresentador abrirá a outra porta contendo a cabra restando apenas a porta com o carro. Para vencer basta trocar de porta. Portanto, usando a estratégia de trocar de portas, sempre que se escolhe no início uma porta errada o jogo é ganho. E a chance de errar no início é de $2/3$.

O problema de Monty Hall apesar de simples mostra como o cálculo correto das probabilidades podem ser contraintuitivas. Mesmo o gênio matemático Paul Erdős só se convenceu da vantagem de se trocar as portas após ver a simulação estocástica do problema ser executada no computador (Rosenhouse, 2009).

O produto desenvolvido possui um tema diferente do problema original. Em vez de um *Game Show* é simulada uma cobrança de pênalti. Em vez de portas são mostradas três possíveis posições onde o goleiro pode estar para pegar a bola. O jogador seleciona uma posição e o sistema revela uma posição do goleiro e pergunta se o jogador quer trocar de porta. O jogador ganha se conseguir selecionar a posição aonde está a bola, e perde se selecionar a posição onde está um goleiro. As figuras 4.15 e 4.16 mostram a seleção das características de jogos e a seleção das características educacionais respectivamente. As

figuras 4.17 e 4.18 mostram as telas de execução do jogo.

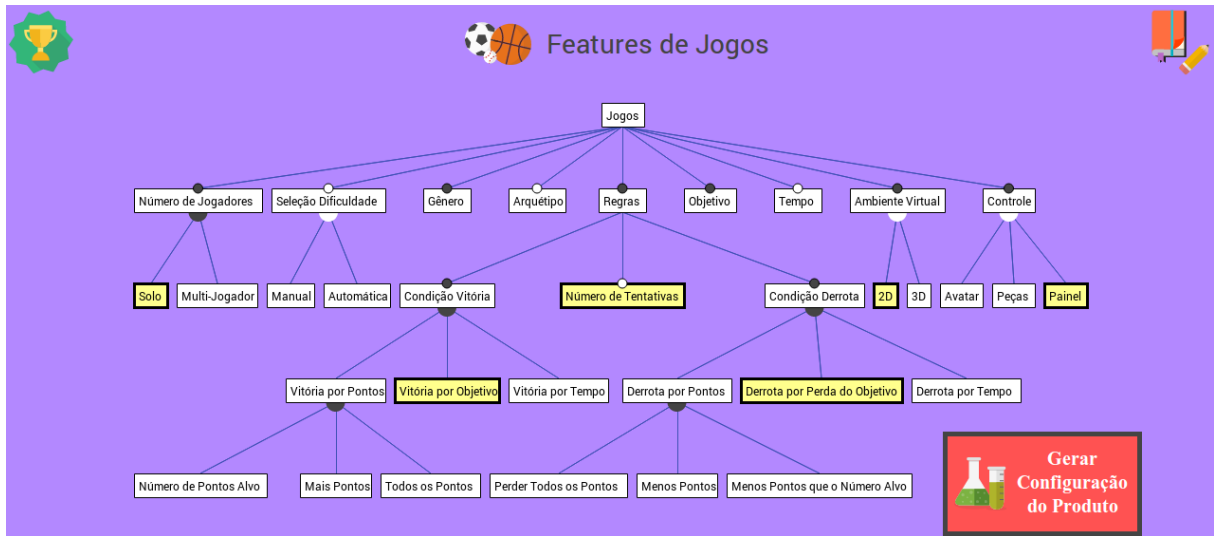


Figura 4.15: Seleção de características de jogos do produto Monty Cup.

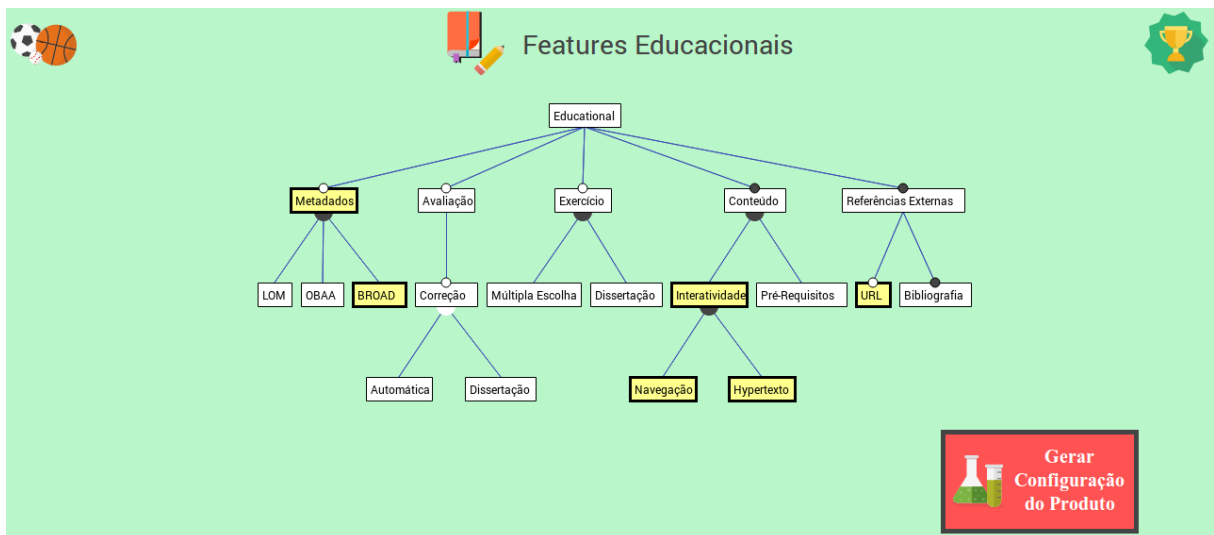


Figura 4.16: Seleção de características educacionais do produto Monty Cup.



Figura 4.17: Configuração inicial do produto Monty Cup



Figura 4.18: O jogador consegue marcar um gol no produto Monty Cup.

4.2.4.3 Game Work

A construção do produto *Game Work* tem como intuito demonstrar a viabilidade de se construir um sistema que aplique as características de gamificação. O produto desenvolvido é um sistema de controle de atividades gamificado de uma empresa de fabricação de software. A figura 4.19 mostra a seleção de característica de gamificação do produto.

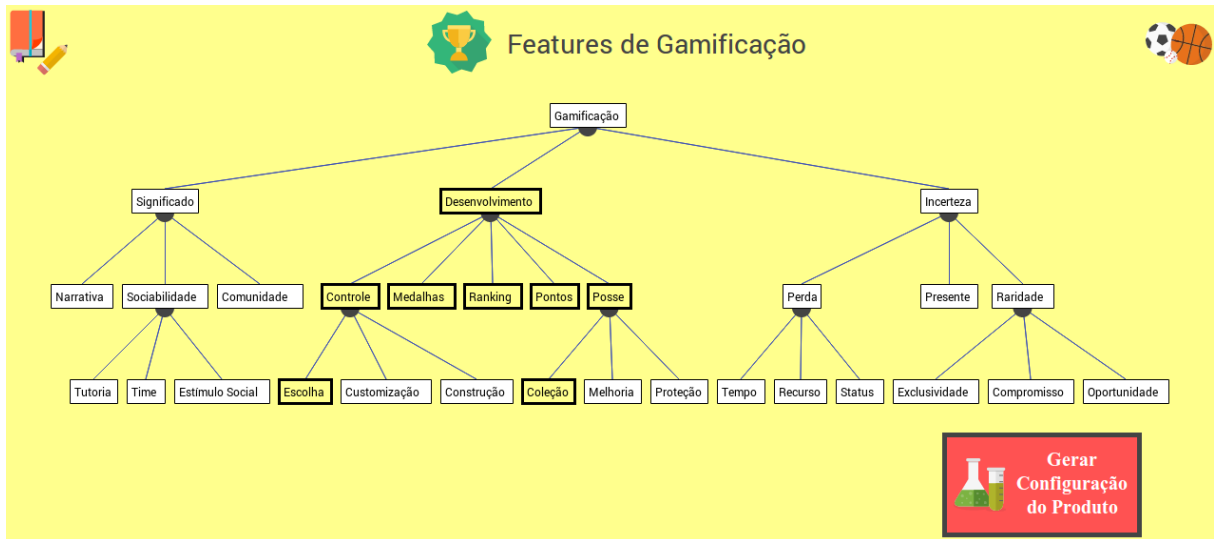


Figura 4.19: Seleção das características de gamificação para o produto *Game Work*.

A aplicação se concentra em características de desenvolvimento. A cada tarefa desempenhada o funcionário recebe pontos pela conclusão dependendo da dificuldade de cada tarefa. Se a tarefa é concluída dentro do prazo o funcionário recebe pontos integrais e se a tarefa for entregue fora do prazo o funcionário recebe menos pontos. Pontos extras também são concedidos caso as tarefas sejam entregues subsequentemente dentro do prazo sem que o funcionário quebre a corrente de entregas dentro do prazo. Como exemplo, foi gerada uma simulação de uma empresa com 18 funcionários. A figura 4.20 mostra o quadro com a classificação dos funcionários.

CLASSIFICAÇÃO			
Anual Mensal Semanal Diário			
Ordem Ascendente	Vanessa Bezerra Silva 9.885 pontos última tarefa concluída há 15 horas	Alexandre Braz Batista 9.734 pontos última tarefa concluída há 22 horas	Rithiane Aparecida Santos 9.691 pontos última tarefa concluída há 20 horas
Ordem Descendente	Alexandre Mattar Domelas 9.560 pontos última tarefa concluída há 11 horas	Thais Richelle Carvalho 9.395 pontos última tarefa concluída há 5 horas	Diackes Diarli Fernandes 9.225 pontos última tarefa concluída há 17 horas
CONQUISTAS	Sarah Vendrame Ferreira 9.061 pontos última tarefa concluída há 6 horas	Diogo Batista dos Santos 8.940 pontos última tarefa concluída há 3 horas	Ricardo Martins Vasconcelos 8.815 pontos última tarefa concluída há 1 dia
Medalhas	Carolina Cardoso Vicente 8.637 pontos última tarefa concluída há 23 horas	Rebeca Hattori da Costa 8.443 pontos última tarefa concluída há 2 horas	Cinthia Cristina Pereira 8.368 pontos última tarefa concluída há 1 dia
OBJETIVOS	Natália Cristina Deliberali 8.305 pontos última tarefa concluída há 12 horas	Isabele Vitória Cunha 8.194 pontos última tarefa concluída há 18 horas	Welington de Lima Campos 8.010 pontos última tarefa concluída há 16 horas

Figura 4.20: Classificação geral dos funcionários.

A concessão de medalhas é outra característica selecionada para o produto. A concessão de medalhas fica a cargo do gerente de cada equipe. Na implementação foram definidas nove tipos de medalhas:

- **Funcionário do Mês:** Concedido ao funcionário que mais se destacou no mês.
- **Excelência:** Concedido a quem demonstrou maestria na execução de suas funções.
- **Crescimento:** Concedido a quem conseguiu obter uma qualificação formal relativa à sua área de atuação.
- **Iniciativa:** Concedido a quem executa suas atividades sem a necessidade de microgerenciamento.
- **Inovação:** Concedido a quem trouxe novas idéias e/ou implementou novas tecnologias que melhoraram o trabalho e a empresa de maneira geral.
- **Sempre Alerta:** Concedido a quem demonstrou capacidade de reagir à pequenas crises tomando as atitudes corretas.
- **Liderança:** Concedido a quem conseguiu com sucesso unir os colegas para a realização de uma tarefa de dificuldade considerável.
- **Além do Dever:** Concedido a quem demonstrou dedicação à empresa além do simples vínculo empregatício.
- **Trabalho em Equipe:** Concedido a quem auxilia os colegas na realização de tarefas e mantém um bom relacionamento com todos.

A figura 4.21 mostra um quadro individual de medalhas de um funcionário que conseguiu coletar todas as medalhas possíveis.

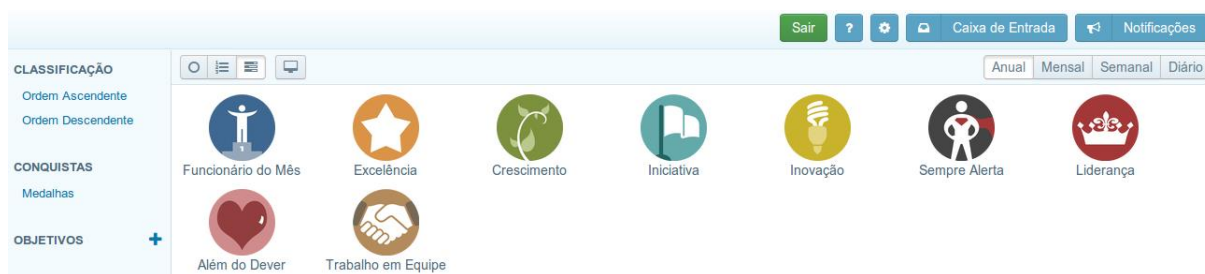


Figura 4.21: Quadro de medalhas de um funcionário que possui todas as medalhas.

Outra característica selecionada e implementada na aplicação *Game Work*, é a capacidade de escolha. A implementação toma forma na possibilidade do funcionário poder escolher quais tarefas deseja executar. A figura 4.22 mostra a tela de seleção de tarefas.



Figura 4.22: Tela de seleção de tarefas na visão do funcionário.

4.2.5 CONCLUSÕES E LIÇÕES APRENDIDAS

O interesse maior é observar a viabilidade de utilização da infraestrutura e conjunto de *features* para a geração automática ou semiautomática de jogos educacionais. Investigações futuras quanto à capacidade de evolução e adaptação às outras tecnologias que não a web poderão validar a confiança na viabilidade de uso da BROAD-PLG.

A experiência em desenvolver os produtos 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 que é menos dependente que mídias foi o que mais 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.

A engenharia de domínio, muito mais do que a engenharia de aplicação, foi o fator mais importante para auxiliar na reutilização e confecção dos produtos. Selecionar as características dos três tipos de interesse (educação, jogos e gamificação) permite, mesmo em alto nível, determinar as partes centrais das funcionalidades do novo produto e permite clarear as fronteiras do escopo.

Embora esteja em sua fase inicial, o projeto BROAD-PLG conseguiu contribuir de maneira satisfatória para a instrumentalização dos exemplos. É sugerido que possíveis avanços sejam direcionados para a engenharia de aplicação possibilitando maior automatização ao custo do estreitamento do escopo.

A avaliação do modelo computacional necessita ser expandida através da criação de modelos e exemplos que contemplem os três modelos e testes que permitam a usuários validarem esses exemplos em características que incluam usabilidade e adequabilidade dos mesmos.

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 alcançados 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. A linha de produtos de software pode ser uma solução viável, apesar do escopo amplo da área de jogos e jogos educacionais e gamificação. a dificuldade em gerenciar as variabilidades de jogos, jogos educacionais e gamificação. Gerenciar as variabilidades desse domínio é um desafio, portanto, ressaltamos a contribuição desse trabalho para os estudos iniciais da LPS.

BROAD-PLG é um projeto em andamento, mas alguns passos importantes já foram finalizados como a especificação de características para jogos, jogos educacionais e gamificação, a definição de uma arquitetura que poderá compor a linha de produtos 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 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 capacidade de adaptação e potencial de geração de produtos de forma semiautomática.

5.1 LIMITAÇÕES E DIFICULDADES

Como descrito anteriormente criar um jogo educacional está longe de ser, ou se tornar em um futuro próximo, uma tarefa trivial. À medida que novas tecnologias surgem e a quantidade de dispositivos de execução de mídia proliferam aumentam os pré-requisitos para se iniciar um projeto de construção de jogos. Nesta seção são listadas limitações e dificuldades a serem superadas tanto pelo BROAD-PLG quanto por abordagens que se propõem a auxiliar no

desenvolvimento de jogos educacionais.

Os alunos nas suas horas de lazer consomem jogos profissionais, de alto orçamento, de alta qualidade gráfica, com efeitos especiais de última geração, com enredos e tramas escritos por profissionais do cinema, com sons engenhosamente construídos para o ambiente, trilha sonora sob medida, dubladores famosos, e outros atributos que fazem com que a experiência de jogar se torne uma atividade que perdure na memória afetiva do jogador por anos a fio. A experiência que o jogador tem em casa, ou em um aparelho móvel, determina um nível alto de expectativa em relação à qualidade que os jogos devem ter que é muito difícil de ser superada por jogos educacionais. O problema pode ser expresso, então, com a seguinte pergunta: “Como criar um jogo educacional que supere, ou ao menos iguale, a expectativa que os alunos, intrinsecamente, têm em relação ao que um jogo deve ser?”.

A resposta pode estar em sucessos recentes, e tendendo a crescer, de uma parte da indústria de jogos denominada *indie*. A palavra *indie* é um apelido para “Independente”. Jogos *indie*, então, são jogos desenvolvidos por independentes, denominação recebida por uma ou poucas pessoas que não estão ligadas a uma grande companhia ou distribuidora. O segredo do sucesso dos jogos *indies* não está na tecnologia ou na qualidade técnica, mas, na inovação. Pela razão de não estarem envolvidos com grandes orçamentos as equipes *indie* podem arriscar mais pois não têm o compromisso de retornar uma grande soma monetária aos investidores. Para mostrar apenas um, de dezenas, de casos de uso de jogos *indie* é apresentado o caso do jogo Flappy Bird. No primeiro trimestre de 2013 um desenvolvedor de jogos vietnamita chamado Dong Nguyen criou no seu tempo livre, no espaço de três dias, um jogo para celulares da Apple no qual o objetivo era guiar, através de toques na tela, um pássaro cartunesco através de um labirinto de canos. No começo de 2014 o jogo, gratuito, rendia ao desenvolvedor \$50.000 dólares, por dia, apenas em anúncios e contava com mais de 50 milhões de downloads (Folha, 2014). O jogo desprezioso virou um fenômeno cultural. A inovação do jogo se caracteriza por uma quebra de um paradigma da indústria de jogos que é fazer jogos simples de aprender e difíceis de dominar. A dificuldade de “aprendizado” ou seja de domínio necessário para começar a se desenvolver no Flappy Bird era extremamente alto embora fosse fácil de se dominar. Embora essa quebra de paradigma não explique por si só o sucesso do jogo, ela inequivocamente teve sua parcela de contribuição.

Uma das limitações, portanto, do BROAD-PLG é a dificuldade que uma ferramenta técnica possui de medir inovação. Essa limitação, no entanto, pode ser mitigada em trabalhos futuros através da adição de um novo domínio que caracterize situações de projeto (design) que fomentem a criatividade, a mistura de elementos (*plot twist*) e a ruptura de *clichés*. Esse

novo domínio possibilita que a camada de aplicação instancie junto com o arcabouço a implementação de mecanismos e ativos de projetos com requisitos a serem satisfeitos.

Uma dificuldade encontrada no desenvolvimento de jogos educativos é a quantidade de tecnologias a serem aprendidas e absorvidas. O desenvolvimento de jogos, apesar de fomentar uma indústria que possui um porte significativo, ainda é considerado um nicho entre os desenvolvedores. Perto do desenvolvimento de aplicações mais comuns como aplicativos financeiros, aplicativos voltados para o comércio, para a indústria, e até para o meio acadêmico a criação de jogos ainda pode ser considerada uma segmentação especializada. Poucos profissionais e desenvolvedores, se comparados com os outros segmentos, dominam as tecnologias e a *expertise* necessárias para se desenvolver jogos de qualidade. O trabalho proposto consegue acelerar parte do desenvolvimento, mas, ainda precisa ser utilizado por pessoas com conhecimentos e habilidades muito específicos. Esses profissionais são escassos e o treinamento pode levar de meses a anos para que um recém-chegado consiga se tornar pleno no desenvolvimento de jogos.

Uma possível solução é a adição de um componente de tutoria na camada de aplicação que, junto com a instanciação do arcabouço, acrescentaria mídias de tutoria, interativas ou não, para auxiliar o desenvolvedor novato por onde e de que maneira deve-se começar a trabalhar a aplicação.

5.2 TRABALHOS FUTUROS E PERSPECTIVAS

Nesta seção são sugeridas algumas direções e perspectivas para trabalhos que visem estender o BROAD-PLG.

Um dos assuntos mais visitados na literatura que aborda LPS é como fazer testes. Atualmente o BROAD-PLG carece de uma especificação para a realização de testes dos produtos. Os testes relativos à validade da configuração são realizados pelo componente Factory Connector. Outros tipos de testes, entretanto, são necessários. Testes de integração, adaptação, unitários, validação, deploy, execução, dinâmico, regressão e aceitação são alguns dos testes sugeridos pela literatura. Cada autor coloca ênfase em um tipo de teste mas todos, de maneira geral, colocam os testes como parte integrante do ciclo de vida da linha de produto. Em trabalhos futuros é interessante abordar os testes, mesmo que somente na fase de engenharia de aplicação.

Outro problema em aberto é como tratar a evolução do produto e do modelo. Inevitavelmente durante o ciclo de vida de qualquer artefato de *software* surgem requisitos

que a arquitetura original não foi planejada para atender. Essa evolução não intencional pode ter como origem vários motivos, como por exemplo: demandas dos *stakeholders* por novas funcionalidades ou alterações em funcionalidades já existentes, características podem ter se tornado obsoletas sendo substituídas por uma ou mais novas características e componentes de terceiros que sofreram alterações e fizeram com que a LPS ficasse incompatível com os novos tipos de entrada e saída. Essas alterações podem ser complexas e necessitam de um acompanhamento tanto metodológico quanto documental. Na literatura existe uma quantidade considerável de sugestões sobre como abordar a evolução de uma LPS, no entanto não há consenso e, certamente, nem todas as soluções são adequadas para o BROAD-PLG. Encontrar qual a melhor solução para o acompanhamento da evolução da LPS fornece um desafio para trabalhos futuros.

REFERÊNCIAS

- ADKINS, S. Sam. **The 2012-2017 Worldwide Game-based Learning and Simulation-based Markets**. Disponível em: http://www.ambientinsight.com/Resources/Documents/AmbientInsight_SeriousPlay2013_WW_GameBasedLearning_Market.pdf. Acesso em 17 de junho de 2014.
- ALVES, C.; Camara, T. & Alves, V. **Experiences with Mobile Games Product Line Development at Meantime Software Product Line Conference**, 2008. SPLC '08. 12th International, 2008, pp. 287-296.
- ANTONIOU, G., Harmelen, F. van. (2008) “**A Semantic Web Primer**”, Cambridge, MA, USA: MIT Press.
- APEL, S.; Batory, D.; Kästner, C. & Saake, G. Springer (Ed.) **Feature-Oriented Software Product Lines**. Springer, 2013.
- ARNAB, S.; Lim, T.; Carvalho, M. B.; Bellotti, F.; de Freitas, S.; Louchart, S.; Suttie, N.; Berta, R. & De Gloria, A. **Mapping learning and game mechanics for serious games analysis**. British Journal of Educational Technology, 2014, 45, pp. 125-146.
- ARIELY, D. **Predictably Irrational, Revised and Expanded Edition: The Hidden Forces That Shape Our Decisions**. HarperCollins, 2009.
- ARIELY, D. **The Upside of Irrationality: The Unexpected Benefits of Defying Logic at Work and at Home**. HarperCollins, 2010.
- BEATSON, O. **Challenge vs Skill**. Disponível em <http://en.wikipedia.org/wiki/File:Challenge_vs_skill.svg>. Acessado em 26 de junho de 2014.
- BELLOTTI, F.; Kapralos, B.; Lee, K.; Moreno-Ger, P. & Berta, R. **Assessment in and of Serious Games: an overview Advances in Human-Computer Interaction**, Hindawi Publishing Corp., 2013, 11 páginas.
- BISADI, M. & Chua, A. Y. **DGIST: a digital game for international students' adjustment Interactive Technology and Smart Education**, Emerald Group Publishing Limited, 2014, 11, 15-31.
- BOLETSIS, C. & McCallum, S. **The Table Mystery: An Augmented Reality Collaborative Game for Chemistry Education Serious Games Development and Applications**, Springer, 2013, pp. 86-95.
- BOSCH, J. Education, P. (Ed.) **Design and use of software architectures: adopting and evolving a product-line approach**. Pearson Education, 2000.
- BOUVIER, P.; Lavoué, E.; Sehaba, K.; George, S. & others. **Identifying learner's engagement in learning games: a qualitative approach based on learner's traces of interaction**. 5th International Conference on Computer Supported Education (CSEDU 2013), Aachen, Germany, 2013, pp. 6-8.
- BOYLE, E. A.; Connolly, T. M.; Hainey, T. & Boyle, J. M. **Engagement in digital**

entertainment games: A systematic review *Computers in Human Behavior*, 2012, 28, 771 – 780.

BRAGA, R. et al., (2011) “**EFACTAGENT: Agent architecture for management, search and retrieval of e-learning artifacts**”, Proceedings of the IADIS International Conference on WWW/Internet 2011, Rio de Janeiro, Brazil, pp.468-472.

BURBAITE, R., and Stukys, V., (2011) “**Analysis of Learning Object Research Using Feature-based Models**”. In *Information Technologies’ 2011*, Proceedings of the 17th International Conference on Information and Software Technologies, Lithuania, pp. 201-208.

CAGE, David. **No one should be allowed to define what a game is**. Disponível em: <<http://www.gamespot.com/articles/david-cage-no-one-should-be-allowed-to-define-what-a-game-is/1100-6415457/>>. Acesso em 10 de junho de 2014.

CAMPOS, F. et alli. (2012) “**BROAD Project: Semantic Search and Application of Learning Objects**”, *IEEE Technology and Engineering Education (ITEE)*, Dez., vol.7, n.3.

CAMPOS, F., Braga, R., Santos, N., Souza, A. C., Mattos, E. and Nery, T. (2011) “**Applying Semantics for the Retrieval of Learning Objects on the Web**”. In: 4th Brazilian Workshop on Semantic Web and Education in SBIE 2011, Aracaju, pp. 2374-2383.

CANTELE, R. C. (2009) “**Construindo ontologias a partir de recursos existentes: uma prova de conceito no domínio da educação**”, Tese de Doutorado.

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**. Eighth International AAAI Conference on Weblogs and Social Media, 2014, v. 10, pp. 1405-1429.

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.

CHIMALAKONDA, S. & Nori, K. **Accelerating educational technologies using software product lines**. *Technology Enhanced Education (ICTEE)*, 2012 IEEE International Conference on, 2012, 1 -4.

CHIMALAKONDA, S. **GAMBLE: Towards Ensuring Quality of Education using Goal Driven Model Based Learning**. *Environments 11th IEEE International Conference on Advanced Learning Technologies*, 2011.

CHIMALAKONDA, S. **Towards Automating the Development of a Family of eLearning Systems**. MS Thesis, IIIT Technical Report, 2010.

CHO, H. & Yang, J.-S. **Architecture Patterns for Mobile Games Product Lines Advanced Communication Technology**, 2008. *ICACT 2008*. 10th International Conference on, 2008, 1, pp. 118-122.

CHOU, Y. **Octalysis: Complete Gamification Framework**. Disponível em: <<http://www.yukaichou.com/octalysis-tool/>>. Acesso em 30 de junho de 2014.

CHUA, A. Y.; Balkunje, R. S. & Chang, K. **Rendezvous-a social web-based application for knowledge sharing and entertainment**. Proceedings of the International MultiConference of Engineers and Computer Scientists, 2012, 1.

CLEMENTS, P. & Northrop, L. Addison-Wesley (Ed.) **Software product lines Addison-Wesley**, 2002.

COSTIKYAN, G. **Uncertainty in Games**. MIT Press, 2013.

CSIKSZENTMIHALYI, M. & Row, H. (Ed.) **Flow: The Psychology of Optimal Experience**. Harper & Row, 1990.

DALMON, D. L. **Uma Linha de Produto de Software para Módulos de Aprendizagem Interativa**. Universidade de São Paulo, 2012.

DAMAŠEVIČIUS, R. and Štuikys, V., (2009) “**Specification and Generation of Learning Object Sequences for E-learning Using Sequence Feature Diagrams and Metaprogramming Techniques**” In: *Advanced Learning Technologies, ICALT 2009*. Ninth IEEE International Conference, July, pp. 572-576.

DIAZ, P.; Paredes, P.; Alvarado, D. & Giaccardi, E. **Co-designing social games with children to support non formal learning** *Advanced Learning Technologies. (ICALT)*, 2012 IEEE 12th International Conference on, 2012, pp. 682-683.

D'AQUIN, M., Gangemi, A., Haase, P. (2006) “**Definition of Ontology Networks**”, *NeOn Book – NeOn Methodology in a Nutshell*. [S.l.: s.n.].

EEDAR. 2013. **Deconstructing Mobile & Tablet Gaming**. Disponível em <<http://www.eedar.com/images/uploads/EEDAR-Deconstructing-Mobile-and-Tablet%20Gaming-2013.pdf>>. Acesso em 11 junho de 2014.

ESA. **Essential Facts About The Computer And Video Game Industry**. Disponível em: <http://www.theesa.com/facts/pdfs/esa_ef_2013.pdf>. Acesso em 11 de junho de 2014.

FERGUSON, C. & Olson, C. **Video Game Violence Use Among “Vulnerable” Populations: The Impact of Violent Games on Delinquency and Bullying Among Children with Clinically Elevated Depression or Attention Deficit Symptoms**. *Journal of Youth and Adolescence*, Springer US, 2014, 43, pp. 127-136.

FOLHA. **'Flappy Bird' chega a 50 milhões de downloads e rende US\$ 50 mil por dia**. Disponível em <<http://www1.folha.uol.com.br/tec/2014/02/1409240-flappy-bird-chega-a-50-milhoes-de-downloads-e-rende-us-50-mil-por-dia.shtml>>. Acesso em 20 de agosto de 2014.

FORBES. **The World's Biggest Public Companies**. Disponível em: <<http://www.forbes.com/global2000/list/>>. Acesso em 2 de junho de 2014.

FORBES. **Thanks To Media, Sports Industry Growth Set To Accelerate**. Disponível em: <[Thanks To Media, Sports Industry Growth Set To Accelerate](#)>. Acesso em 11 de junho de 2014.

GARTNER. **Gartner Says Worldwide Video Game Market to Total \$93 Billion in 2013.** Disponível em: <<http://www.gartner.com/newsroom/id/2614915>>. Acesso em 11 de junho de 2014.

GIANNAKOS, M. N. **Enjoy and learn with educational games: Examining factors affecting learning performance** *Computers & Education*. Elsevier, 2013, 68, 429-439.

GILLAIN, J.; Faulkner, S.; Heymans, P.; Jureta, I. & Snoeck, M. **Product portfolio scope optimization based on features and goals.** Proceedings of the 16th International Software Product Line Conference-Volume 1, 2012, pp. 161-170.

GIUNCHIGLIA, F.; Khandaker, T. H.; Sheikh Shaugat, A. & Rezwana, 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.

GOMAA, H. WESLEY, A. (Ed.) **Designing Software Product Lines With Uml: From Use Cases to Pattern-Based Software Architectures.** ADDISON WESLEY Publishing Company Incorporated, 2005.

GONÇALVES, H. D. A. **Objetos de Aprendizagem (AO): Contribuições ao Ensino de Metodologia Científica Online.** Proceedings of International Conference on Engineering and Technology Education, 2014, 13.

HAMARI, J.; Koivisto, J. & Sarsa, H. **Does Gamification Work?--A Literature Review of Empirical Studies on Gamification System Sciences.** (HICSS), 2014 47th Hawaii International Conference on, 2014, pp. 3025-3034.

HEINZEN, T. **Gamification of a Statistics Course.** Farmingdale State College Teaching of Psychology Conference 2013: Conference Proceedings, 2013, 59.

HORACHEK, D. Jadhav, N. (Ed.) **Creating E-Learning Games with Unity.** Packt Publishing, 2014.

KANG, K.; Cohen, S.; Hess, J.; Novak, W. & Peterson, A. **Feature-oriented domain analysis (FODA) feasibility study.** DTIC Document, 1990.

KAPP, K. Taff, R. (Ed.) **The Gamification of Learning and Instruction: Game-based Methods and Strategies for Training and Education.** Wiley, 2012.

KHATIB, F.; DiMaio, F.; Cooper, S.; Kazmierczyk, M.; Gilski, M.; Krzywda, S.; Zabranska, H.; Pichova, I.; Thompson, J.; Popović, Z. & others. **Crystal structure of a monomeric retroviral protease solved by protein folding game players.** Nature structural & molecular biology, Nature Publishing Group, 2011, 18, pp. 1175-1177.

KIM, P. W.; Kim, S. Y.; Shim, M.; Im, C.-H. & Shon, Y.-M. **The influence of an educational course on language expression and treatment of gaming addiction for massive multiplayer online role-playing game (MMORPG).** *Players Computers & Education*, Elsevier, 2013, 63, 208-217.

KNIGHT, C., Gašević, D., Richards, G. (2006) "An Ontology-Based Framework for

Bridging Learning Design and Learning Content”, IEEE Educational Technology & Society, Vol. 9, No. 1, pp. 23-37.

KNOWLEDGE, Wharton. **Nike FuelBand: Did the Brand Score a Goal?** Disponível em <https://knowledge.wharton.upenn.edu/article/nike-fuelband-band-brand-strike-goal/>. Acesso em 30 de junho de 2014.

KOHN, A. Harcourt, H. M. (Ed.) **Punished by Rewards: The Trouble with Gold Stars, Incentive Plans, A's, Praise, and Other Bribes**. Houghton Mifflin Harcourt, 1999.

LEE, J., Muthig, D., Naab, M. (2008) “**An approach for developing service-oriented product lines.**” In: SPLC’08: 12th International Software Product Line Conference. IEEE Computer Society, pp. 275–284.

LEE, J., Kang, K. C. (2004) “**Feature Binding Analysis for Product Line Component Development.**”, Software Product Family Engineering, Lecture Notes in Computer Science, pp. 266-276, Springer-Verlag.

LEMANN, Fundação. **A visita de Khan ao Brasil e o que vem por aí**. Disponível em <<http://fundacaolemann.org.br/novidades/khan-no-brasil>>. Acesso em 30 de junho de 2014.

MARKETS, and Markets. **Gamification Market**. Disponível em <<http://www.marketsandmarkets.com/Market-Reports/gamification-market-991.html>>. Acesso em 30 de junho de 2014.

MATOS, E. E. et al. (2009) “**CelOWS: an ontology based framework for the provision of semantic web services related to biological models.**”, Journal of Biomedical Informatics, v. 43, pp. 125-136.

MEDEIROS, F. M., de Almeida, E. S., de Lemos Meira, S. R. (2010) “**Designing a set of Service-Oriented Systems as a Software Product Line.**”, 2010 Fourth Brazilian Symposium on Software Components, Architectures and Reuse (SBCARS), pp. 70-79.

MENDES, L. F. et al. (2011). **SASAgent: An agent based architecture for search, retrieval and composition of scientific models**, Computers in Biology and Medicine, v. 1, pp. 1-14.

MOHALIK, S.; Ramesh, S.; Millo, J.-V.; Krishna, S. N. & Narwane, G. K. **Tracing spls precisely and efficiently**. Proceedings of the 16th International Software Product Line Conference-Volume 1, 2012, pp. 186-195.

MPAA. **Theatrical Market Statistics**. Disponível em: <http://www.mpaa.org/wpcontent/uploads/2014/03/MPAA-Theatrical-Market-Statistics-2013_032514-v2.pdf>. Acesso em 11 de junho de 2014.

NAH, F. F.-H.; Eschenbrenner, B.; Zeng, Q.; Telaprolu, V. R. & Sepehr, S. **Flow in gaming: literature synthesis and framework development**. International Journal of Information Systems and Management, Inderscience, 2014, 1, pp. 83-124.

NEUMANN, N. & Böckenholt, U. **A Meta-analysis of Loss Aversion in Product Choice**. Journal of Retailing , 2014, 90, pp. 182 – 197.

OBAA. **Projeto de Objetos de Aprendizagem Baseado em Agentes** (2012) Disponível em: <http://www.portalobaa.org/>. Acesso em 18/08/2014.

O'ROURKE, E.; Haimovitz, K.; Ballwebber, C.; Dweck, C. S. & Popovic, Z. **Brain Points: A Growth Mindset Incentive Structure Boosts Persistence in an Educational Game**. CHI - Conference on Human Factors in Computing Systems, 2014.

PALAZZI, D.; Matos, E. E. S.; Campos, F.; Braga, R. (2010) "**Development Approach for Modeling Biological Ontologies**." In: Joint 5th International Workshop on Vocabularies, Ontologies and Rules for the Enterprise (VORTE), Vitória.

PARLETT, D. Press, O. U. (Ed.) **The Oxford History of Board Games Oxford**. University Press, 1999.

PERNAS, A. M., Diaz, A., Motz, R. (2011) "**Situations and ontology networks to define adaptive actions in e-learning systems**", Proceedings of the IADIS International Conference on WWW/Internet 2011, Rio de Janeiro, Brazil, pp. 237-244.

PINK, D. **Drive: The Surprising Truth About What Motivates Us**. Penguin Group US, 2011.

POHL, K.; Bockle, G. & Van Der Linden, F. Springer (Ed.) **Software product line engineering**. Springer, 2005, 10.

REIS, V. R. **Implementing Software Product Line Adoption Strategies**. Universidade Federal de Pernambuco, 2007.

Rezende, P.; Campos, F.; David, J. M. N. & Braga, R. **BROAD-RS: Uma Arquitetura para Recomendação de Objetos de Aprendizagem Sensível ao Contexto Usando Agentes**. ESUD 2013 – X Congresso Brasileiro de Ensino Superior a Distância, 2013.

RIBEIRO, M. M. **Emergent Feature Modularization**. Universidade Federal de Pernambuco, 2012.

RICHARD M. Sherman and Robert B. Sherman, "**A Spoonful of Sugar**", Mary Poppins, The Walt Disney Company, 1964.

RIGBY, S. & Ryan, R. Andersen, R. (Ed.) **Glued to Games: How Video Games Draw Us in and Hold Us**. Spellbound Praeger, 2011.

ROSENHOUSE, J. **The Monty Hall Problem : The Remarkable Story of Math's Most Contentious Brain Teaser**. Oxford University Press, USA, 2009.

ROYER, J. & Arboleda, H. Wiley (Ed.) **Model-Driven and Software Product Line Engineering**. Wiley, 2013.

SALEN, K. & Zimmerman, E. Salen, K. (Ed.) **Rules of Play: Game Design Fundamentals**. MIT Press, 2004.

SANTOS, J., David, J. M. N., Campos, F., Braga, R. (2012) "**EasyT: Apoiando a Construção de Objetos de Aprendizagem para uma Linha de Produtos de Software**", VII Conferência Latinoamericana de Objetos e Tecnologias de Aprendizagem, Guayaquil, pp.

1-11.

SANTOS, N. et al. (2008) “**Digital Libraries and Ontology**”. In: Yin-Leng Theng; Schubert Foo; Dion Hoe-Lian Goh; Jin-Cheon Na.. (Org.). *Handbook of Research on Digital Libraries: Design, Development, and Impact*. Hershey, Pensilvania: Idea Group Inc, 2008, v. 1, pp. 206-214.

SEI. **Software Engineering Institute**. Disponível em: https://www.sei.cmu.edu/product_lines/ppl/index.html. Acesso em 15 de agosto de 2014.

SCHEIDT, R.; Schmidt, K.; Pessoa, G.; Viera, M. & Dantas, M. **A software product line approach to enhance a meta-scheduler middleware**. *Journal of Physics: Conference Series*, 2012, 341, 012030.

SCHELL, J. Kaufmann, M. (Ed.) **The Art of Game Design: A book of lenses**. Taylor & Francis, 2008.

SHIN, N.; Sutherland, L. M.; Norris, C. A. & Soloway, E. **Effects of game technology on elementary student learning in mathematics**. *British Journal of Educational Technology*, Wiley Online Library, 2012, 43, 540-560.

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, v. 20, pp. 87.

SILVA, A. P; **Uma Linha de Produto de Software baseada na Web Semântica para Sistemas de Tutores Inteligentes**. Universidade Federal de Campina Grande, 2011.

SILVA, L. et al. (2011) “**Composer-Science: A semantic service based framework for workflow composition in e-Science projects**”. *Information Sciences*, p. 186-208.

SOUSA, L. O. S. **ReqSys-MDD: Uma Ferramenta para Mapeamento entre Modelos de Features e Requisitos em Linhas de Produto de Software**. Universidade Federal do Rio Grande do Norte, 2012.

SOUZA, A. C.; Campos, F.; Braga, R.; Rabello, C. & Santos, N. **Metadados Educacionais para Objetos de Aprendizagem: Projeto BROAD**. *Congresso Iberoamericano de Informática na Educação*, 2010, 3, 1.

SOUZA, A. C., et al., (2010) “**Metadados Educacionais para Objetos de Aprendizagem: Projeto BROAD**”. In: *Congreso Iberoamericano de Informática Educativa*, Santiago. *Anais do RIBIE 2010*, v. 1, pp. 653-659.

SYLVESTER, T. Roumeliotis, R. (Ed.) **Designing Games: A Guide to Engineering Experiences**. O'Reilly Media, Incorporated, 2013.

THOMAS, C. & Berkling, K. **Redesign of a gamified Software Engineering course Interactive Collaborative Learning (ICL)**, 2013 International Conference on, 2013, 778-786.

WEISS, D. M. & Lai, C. T. R. Huber, A. R. (Ed.). **Software Product-line Engineering: A Family-based Software Development Process**. Addison-Wesley Longman Publishing Co.,

Inc., 1999.

WIKIPEDIA. **Development of Grand Theft Auto V**. Disponível em <http://en.wikipedia.org/wiki/Development_of_Grand_Theft_Auto_V>. Acesso em 18 de junho de 2014.

WOHLIN, C., et alli. (2000) “**Experimentation in Software Engineering: An Introduction**”. Springer.

ZHANG, W. & Jarzabek, S. **Reuse without compromising performance: industrial experience from RPG software product line for mobile devices**. Software Product Lines, Springer, 2005, pp. 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.