

Arquitetura de uma Plataforma para um Ecossistema de Software no Domínio de Ensino de Programação

Rodrigo Ribeiro Silva^{1*}Juliana Costa Fernandes^{1,2}Rodrigo Pereira dos Santos¹

Universidade Federal do Estado do Rio de Janeiro, Depto. de Informática Aplicada & Programa de Pós-Graduação em Informática, Brasil ¹
Instituto Federal do Piauí, Coordenação do Curso de Informática, Brasil ²

RESUMO

O conceito de ecossistemas de software (ECOS) surgiu ao longo dos anos visando melhorias organizacionais. Em sua essência, encoraja que desenvolvedores externos interajam com uma plataforma de software de modo que possam contribuir e evolui-la. Paralelamente, a dificuldade de ensinar programação para iniciantes no ensino superior tornou-se um fator limitante nos cursos de informática. A fim de diminuir a problemática, este trabalho visa caracterizar uma plataforma para o desenvolvimento de um ECOS a partir de um jogo digital para o ensino de programação. Como resultados, foram catalogados os principais requisitos de jogos e tecnologias para o desenvolvimento da plataforma e foi definida a arquitetura, baseando-se no modelo de notação *Software Supply Network* (SSN) e seguindo a Linguagem de Modelagem Unificada (UML) para gerar um diagrama de classe. Além da estruturação, um projeto do uso do jogo digital é apresentado, mostrando uma forma de utilização e um protótipo das principais interfaces.

Palavras-chave: Jogos digitais, gamificação, motores de jogos, elementos de jogos, ecossistemas de software, *software supply network*.

1 INTRODUÇÃO

Segundo dados levantados pela SEMESP [1], um em cada três alunos ingressantes de Sistemas de Informação (SI) terminam o curso, enquanto que, para as faculdades de Ciência da Computação, os números são de um para quatro alunos. Logo, uma alta evasão de ingressantes nos cursos da área de Informática é percebida. Para Deters et al. [2], o desenvolvimento do raciocínio lógico-matemático é de grande dificuldade, ou seja, a programação, que é uma disciplina contida nos cursos da área de TI, pode ser vista como um fator limitante.

Com o intuito de auxiliar o processo ensino-aprendizagem de programação, os métodos para lecionar novos conteúdos vêm se difundindo. Os jogos digitais voltados ao ensino têm se tornado elementos de motivação para os alunos, uma vez que possibilitam o aprendizado de maneira lúdica. Segundo a revisão sistemática realizada por Scaico e Scaico [3], o uso de jogos no ensino introdutório de programação traz resultados positivos para o aprendizado e engajamento.

Junto a essa realidade, a Engenharia de Software (ES) apresenta novas formas de desenvolver funcionalidades em SI, como os ecossistemas de software (ECOS) [4]. Em ECOS, o desenvolvimento é distribuído em torno de uma plataforma e baseado em uma arquitetura comum que agrega à sociedade, de modo que a indústria de software exista para produzir valor e que

a sua percepção seja considerada em todos os aspectos [5][6][7].

Segundo Graciano Neto et al. [4], o conceito de ECOS é inerente ao de SI, fazendo com que novas tecnologias atinjam mais rapidamente seu público alvo, se levadas em consideração a sua portabilidade e o forte suporte para técnicas adequadas de ES [8][9]. Tendo em vista ambos os conceitos (as novas tendências para as funcionalidades de SI que estão sendo apresentadas nos dias de hoje e o uso de jogos na educação), o raciocínio de uni-los pode ser coerente e relevante ao panorama atual da ES. Com esse intuito, o presente trabalho tem como objetivo caracterizar uma plataforma para o desenvolvimento de um ECOS a partir de um jogo digital para o ensino de programação.

O trabalho está organizado da seguinte forma: a Seção 2 discorre sobre a fundamentação teórica; a Seção 3 apresenta os trabalhos relacionados; a Seção 4 apresenta a arquitetura da plataforma; e a Seção 5 traz as conclusões.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 Cenário para Ensino de Programação

Em Santos et al. [10], foi apresentado que a falta de compreensão do raciocínio lógico representa uma das principais razões do alto índice de reprovação nas disciplinas de algoritmos. Portanto, é importante pensar em um cenário que o processo ensino-aprendizado aconteça de forma eficaz a ponto de estimular o aluno a prosseguir.

Neto et al. [11] acreditam que o aluno deve deixar de ser passivo e tornar-se o construtor da própria sabedoria, tendo o professor como um agente facilitador de seu desenvolvimento cognitivo. Nesse sentido, é importante encontrar o principal papel para o professor, a fim de incentivar esse crescimento.

Para Vygotsky [12], existem três níveis de desenvolvimento (Figura 1): (1) Zona de Desenvolvimento Atual (ZDA), em que o aprendiz é capaz de realizar sozinho certas tarefas; (2) Zona de Desenvolvimento Proximal (ZDP), considerada o ideal, na qual o aprendiz consegue realizar determinadas tarefas com a ajuda de terceiros; e (3) Nível de Desenvolvimento Potencial (NDP), em que as tarefas estão além do nível de desenvolvimento cognitivo do aprendiz e este não consegue desempenhar nenhuma tarefa, mesmo com a ajuda de outros. Logo, é importante determinar a zona em que as atividades devem se encontrar, pois, se estiverem na ZDA, os estudantes conseguem executá-las sem avanços significativos. Se estiverem além da zona ZPD (i.e., NDP), o aluno pode ficar desmotivado.

Para conseguir atingir a zona adequada é importante entender e esboçar o cenário ideal para o processo ensino-aprendizagem. A partir dessa análise, Santos et al. [10] identificaram o tutor, o monitor e o aprendiz como os principais atores nesse contexto.

*e-mail: rodrigo.ribeiro@uniriotec.br

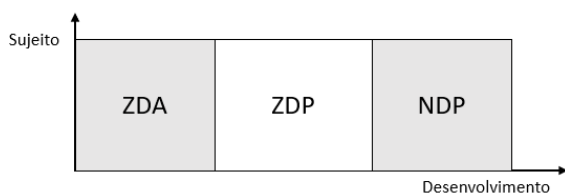


Figura 1: Níveis de Desenvolvimento. Fonte: Santos et al. [10].

A Figura 2 mostra a arquitetura elaborada por Santos et al. [10] para apoiar o cenário proposto.

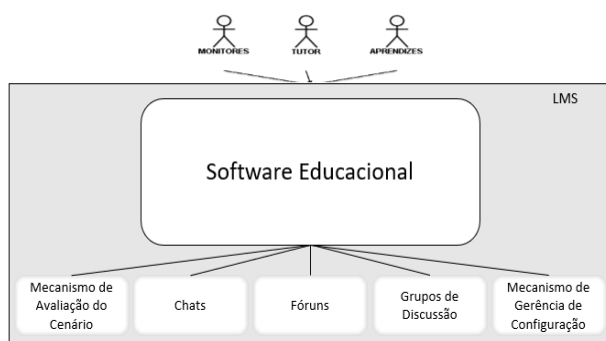


Figura 2: Arquitetura de um Ambiente para Apoiar o Cenário para Ensino de Programação. Adaptado de Santos et al. [10].

2.2 Jogos Digitais

Os jogos digitais na educação têm o poder de estimular o aluno, despertar sua curiosidade e criatividade, desenvolver a capacidade de concentração e raciocínio e proporcionar uma maneira de aprender mais prazerosa. Eles são territórios para a experimentação, onde simulam um ambiente passível de tentativas e erros [13]. A lógica de funcionamento dos jogos digitais apresenta grande capacidade de promover o engajamento de seus participantes, a curiosidade, a colaboração e o aprendizado pelos erros¹. Uma metodologia representativa do uso de jogos na educação é a gamificação, que significa a utilização de jogos eletrônicos em contextos não relacionados a jogos [14]. Neste sentido, Sá et al. [15] apresentam a possibilidade de momentos lúdicos como iterações do processo cognitivo.

Atualmente, os jogos digitais têm sido um recurso utilizado para auxiliar o processo ensino-aprendizado, visto que são uma forma lúdica para abordar um problema qualquer. Em um estudo realizado por Hoefelmann [16], os jogos digitais têm grande potencial para auxiliar a educação, podendo ser um importante instrumento para o processo.

Segundo Uzunca e Jansen [17], os jogos são uma nova forma de aprender, pois eles proporcionam: (1) o aprendizado em uma realidade virtual e/ou com papel simulados. Isso facilita que o jogador não tenha medo de errar, pois sempre pode recomeçar do zero; (2) aprender fazendo, se envolvendo, sendo motivado e se entretendo; (3) a capacidade de encontrar e usar a informação, sem precisar ficar memorizando; e (4) simular ambientes complexos/caros e situações perigosas/críticas. A Tabela 1 mostra uma justificativa para o uso de jogos como forma de aprendizado.

Tabela 1: Justificativa para o Uso de Jogos. Adaptado de Uzunca e Jansen [17].

Justificativa	Descrição
Importância crescente	Design, engenharia e produção de jogos sérios ² ainda estão em sua infância
Alvo em movimento	Negócios emergentes com múltiplos usos nos setores da indústria, educação, saúde e administração pública
Potencial de crescimento	2,35 bilhões de euros no mercado global (estimativa do IDATE 2014 ³)
	Fragmentado e precisa de massa crítica para competir globalmente

2.3 Elementos de Jogos

A gamificação pode ser definida de forma mais consistente como sendo uma estratégia apoiada na aplicação de elementos de jogos para atividades *non-game*⁴ que é utilizada para influenciar e causar mudanças no comportamento de indivíduos e grupos [20].

Segundo Werbach e Hunter [21] e Costa e Marchiori [22], os elementos de jogos são classificados de três tipos: (1) Dinâmicos, que representam o mais alto nível de abstração dos elementos; (2) Mecânicos, que se referem a ações mais específicas; e (3) Componentes, são aplicações específicas visualizadas e utilizadas na interface do jogo. As três categorias são conceituadas nas Tabelas 2. Kapp [14] também comenta sobre outro importante item, a abstração. Nesse sentido, os jogos devem envolver a abstração da realidade, assim como um software. As histórias e signos utilizados no jogo podem carregar tais abstrações, concentrando apenas nos aspectos essenciais do contexto.

Além destes elementos, a movimentação deve ser considerada um elemento de jogo visto que a maneira como o personagem se desloca pelo ambiente proposto é um importante requisito a ser analisado. A representação do jogador no jogo digital deve ser capaz de se locomover através do cenário e, a partir desse movimento, concluir as etapas propostas para atingir o objetivo final.

2.4 Motores de Jogos

Existem diversas etapas entre a idealização e a concepção de um jogo digital. Para facilitar esse processo, existem os motores de jogos. Do inglês *game engines* (ou apenas *engine*), estes motores consistem em programas para computador capazes de simplificar o processo de criação de jogos eletrônicos. Eles servem como um conjunto de bibliotecas que facilitam o desenvolvimento, fazendo com que o programador não precise fazer tudo do zero. Geralmente, dão suporte para animações, sons, colisões, parte gráfica, entre outros.

Na Tabela 3, estão listados alguns motores de jogos levantados por Lima et al. [23] e por Silva et al. [24], o que ajuda a catalogar os principais ambientes de desenvolvimento para, posteriormente, o jogo digital em questão ser desenvolvido. O motor de jogo selecionado será um dos fornecedores para a plataforma, visto que fornecerá produtos/serviços para sua criação.

¹<http://ois.sebrae.com.br/boaspraticas/gamefication-aplicacao-da-logica-de-jogos-na-educacao/>

²Para Protopsaltis et al. [18], são jogos que envolvem alcance de metas baseadas no mundo real.

³http://www.idate.org/en/News/Serious-Games_643.html

⁴Forma de entretenimento que não tem um vencedor ou conclusão [19].

Tabela 2: Elementos de Jogos.

Categoria	Elemento	Descrição
Dinâmico	Emoções	Jogos podem criar diferentes tipos de emoções, especialmente a da diversão (reforço emocional que mantém as pessoas jogando).
Dinâmico	Narrativa	Estrutura que torna o jogo coerente. A narrativa não tem que ser explícita, como uma história em um jogo. Também pode ser implícita, na qual toda a experiência tem um propósito em si.
Dinâmico	Progressão	Ideia de dar aos jogadores a sensação de avançar dentro do jogo.
Dinâmico	Relacionamentos	Refere-se à interação entre os jogadores, seja entre amigos, companheiros ou adversários.
Dinâmico	Restrições	Refere-se à limitação da liberdade dos jogadores dentro do jogo.
Mecânico	Aquisição de recursos	O jogador pode coletar itens que o ajudam a atingir os objetivos.
Mecânico	Feedback	A avaliação permite que os jogadores vejam como estão progredindo no jogo.
Mecânico	Chance	Os resultados de ação do jogador são aleatórios para criar uma sensação de surpresa e incerteza.
Mecânico	Cooperação e competição	Cria-se um sentimento de vitória e derrota.
Mecânico	Desafios	Os objetivos que o jogo define para o jogador.
Mecânico	Recompensas	O benefício que o jogador pode ganhar a partir de uma conquista no jogo.
Mecânico	Transações	Significa compra, venda ou troca de algo com outros jogadores no jogo.
Mecânico	Turnos	Cada jogador no jogo tem seu próprio tempo e oportunidade para jogar.
Mecânico	Vitória	O “estado” que define ganhar o jogo.
Componente	Avatar	Representação visual do personagem do jogador.
Componente	Bens virtuais	Itens dentro do jogo que os jogadores podem coletar e usar de forma virtual e não real, mas que ainda tem valor para o jogador.
Componente	Boss	Um desafio geralmente difícil no final de um nível que tem de ser derrotado, a fim de avançar no jogo.
Componente	Coleções	Formadas por itens acumulados dentro do jogo. Emblemas e medalhas são frequentemente parte de coleções.
Componente	Combate	Disputa que ocorre em que o jogador possa derrotar oponentes em uma luta.
Componente	Conquistas	Recompensa que o jogador recebe por fazer um conjunto de atividades específicas.
Componente	Conteúdos desbloqueáveis	A possibilidade de desbloquear e acessar certos conteúdos no jogo se os pré-requisitos forem preenchidos. O jogador precisa fazer algo específico para ser capaz de desbloquear o conteúdo.
Componente	Medalhas	Representação visual de realizações dentro do jogo.
Componente	Gráfico social	Capacidade de ver amigos que também estão no jogo e ser capaz de interagir com eles. Um gráfico social torna o jogo uma extensão de sua experiência de rede social.
Componente	Missão	Similar a “conquistas”. É uma noção de jogo de que o jogador deve fazer executar algumas atividades que são especificamente definidas dentro da estrutura do jogo.
Componente	Níveis	Representação numérica da evolução do jogador. O nível do jogador aumenta à medida que o jogador se torna melhor no jogo.
Componente	Pontos	Ações no jogo que atribuem pontos. São muitas vezes ligadas a níveis.
Componente	Presentes	A possibilidade distribuir ao jogador coisas como itens ou moeda virtual para outros jogadores.
Componente	Ranking	Lista jogadores que apresentam as maiores pontuações/conquistas/itens em um jogo.
Componente	Times	Possibilidade de jogar com outras pessoas com mesmo objetivo.

Tabela 3: Comparação entre as Principais Tecnologias para Desenvolver Jogos.

Nome	Programação	Prós	Contras	Licença
Adobe Flash Professional CS5	ActionScript	Multiplataforma, formato vectorial	Linguagem própria	Adobe
Alice	Visual	3D, criação de métodos	Não possui muitos recursos	Livre
Construct 2/ Construct 3	Java e C++	<i>Drag-n-drop</i> , detecta colisão e noções de eventos	Só pode rodar em HTML no <i>Free</i>	Scirra
GameMaker	Delphi	Tutoriais, versão "lite", linguagem simples	Linguagem própria	Yoyo
Scratch	Visual	Fácil "programação"	Comandos limitados	Livre
Stencyl	Haxe, C++ e Java	<i>Drag-n-drop</i> , multiplataforma, possibilita extensões	-	Stencyl, LLC
Unity 3D	C++ e C#	3D, multiplataforma, fácil de usar, melhor condição de licenciamento	Ferramentas limitadas	Unity Technologies

2.5 Ecossistema de Software

A comunidade de ES tem investigado aspectos técnicos, econômicos e sociais proporcionando o desenvolvimento de plataformas globalizadas que têm sido tratadas como o núcleo de ecossistemas de software (ECOS) [25]. ECOS é um conjunto de atores funcionando como uma unidade e interagindo em um mercado compartilhado de software e serviços, centrado em uma plataforma comum [26]. Neste contexto, fornecedores de software trabalham de forma cooperativa e competitiva para apoiar e desenvolver novos produtos e serviços.

Os ECOS se transformaram em uma alternativa para empresas, onde uma plataforma de software possibilita a interoperabilidade de componentes de diferentes desenvolvedores, com o intuito de tecnologias se comunicarem, compartilharem e usarem informações [4]. Alguns autores sumarizaram grandes temas para as teorias e conceitos identificados de ECOS, conforme são exibidos na Tabela 4 [5]. O grande desafio é quanto à qualidade de um ECOS, que pode ser medido por meio de sua estrutura, saúde e prosperidade [25].

Tabela 4: Fontes da Teoria de ECOS.
Adaptado de Hanssen e Dybå [27].

Categoria	Teoria e Conceito
Funções e características	Abertura/transparência
Estrutura e forma	Papéis e relacionamentos
Descrição e modelagem	Modelagem e taxonomia

Um dos benefícios para uma empresa (ou desenvolvedor) se tornar um membro de um ECOS é a oportunidade de explorar a inovação aberta, para que atores colaborem a fim de alcançar benefícios regionais e globais [5]. É importante nos dias atuais pensar em desenvolver softwares considerando a abertura de plataforma para possibilitar a troca de informação em rede, bem

como redução de custos tendo em vista a possibilidade de desenvolvedores externos à organização colaborarem neste processo. O desafio, portanto, consiste em entender o dinamismo destas trocas de informação entre plataformas abertas e propor, construir e compreender o impacto dos sistemas de informação que apoiam este cenário.

As teorias e conceitos sumarizados na Tabela 4 são importantes para descrever as principais características do jogo digital proposto neste trabalho, visto que o jogo servirá como uma plataforma de software comum para a formação de um possível ECOS. A primeira teoria, de abertura e transparência, é essencial para o conceito “sem fronteiras” da plataforma, pois é necessário entender o ecossistema como um sistema aberto que interage com o meio em que está inserido. Os papéis e relacionamentos também são importantes para o processo, pois o jogo digital está inserido em um cenário para ensino de programação e conhecer os principais atores torna-se fundamental para apoiar o uso da plataforma como método de ensino. Por último, a modelagem e taxonomia, conceitos importantes para descrever o ECOS por meio de seus componentes. Santos [5] sumariza alguns desafios e fatores de ECOS em relação aos jogos digitais. A Tabela 5 exhibe os desafios e a classificação por nível organizacional, rede de produção de software e ecossistema apontados em pesquisas no assunto. As teorias nas quais a arquitetura da plataforma proposta neste estudo se baseia são apresentadas como desafios para ECOS no domínio de jogos digitais, reforçando a sua justificativa.

2.6 Software Supply Network

O modelo de rede de fornecimento de software, do inglês *Software Supply Network* (SSN), serve para representar os atores principais que interagem em um ECOS. Boucharas et al. [28] propõem uma modelagem que divide ECOS em três níveis: (1) Nível 1, os objetos de estudo são os próprios atores e suas relações no contexto da organização pertencem a um ECOS; (2) Nível 2, os principais atores e suas relações diferentes são os objetos de estudo (o foco está nas conexões entre software, hardware e serviços organizacionais); e (3) Nível 3, os objetos de estudo são os próprios ECOS, incluindo as relações entre eles.

Tabela 5: Desafios de ECOS para Jogos Digitais.
Adaptado de Santos [5].

Nível	Desafio
Organizacional	Arquitetura extensível portátil e com variabilidade na plataforma.
Rede de Produção de Software	Estabelecimento das relações entre atores.
Ecosistema	Caracterização e modelagem de ECOS.

A Figura 3 exibe os níveis propostos. Para Boucharas et al. [28], é importante entender que os ECOS devem ter seus limites bem definidos, e.g. uma tecnologia, uma plataforma, um mercado ou uma empresa.

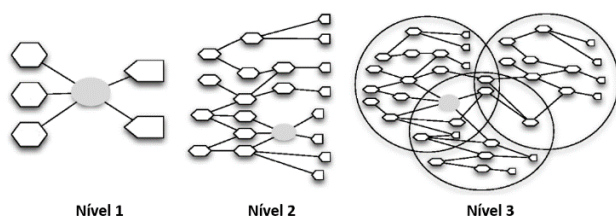


Figura 3: Níveis de um ECOS.
Adaptado de Jansen et al. [26].

O modelo SSN foi escolhido para representar a plataforma sugerida neste trabalho e a notação utilizada nesta proposta está baseada no modelo recomendado por Costa et al. [29], no qual o próprio software é o elemento principal para a formação do ecossistema. Os componentes do modelo SSN são apresentados na Figura 4.

3 TRABALHOS RELACIONADOS

Pettersson et al. [30] discutiram sobre a essência do conceito de ECOS e elaborou, através de uma análise de um ecossistema para *mobile learning (m-Learning)*, uma série de aspectos particulares ao domínio. Os autores contribuíram formando um modelo de referência para *m-Learning* e um esboço para um método de análise para ECOS de domínio específicos.

Este trabalho tem como objetivo caracterizar como um jogo digital pode servir como plataforma comum em um ECOS, criando um modelo de referência próprio para o domínio. Logo, é uma abordagem de pesquisa em um cenário diferente do proposto por Pettersson et al. [30], que utilizou um contexto referente a um ECOS para *m-Learning*.

4 ARQUITETURA DA PLATAFORMA

Esta seção apresenta uma proposta de arquitetura de uma plataforma para um ECOS baseado em um jogo digital educacional. Além disso, é apresentada uma possível modelagem, implementação e projeto para o ensino de programação.

Segundo Silva et al. [24], o ensino de programação faz parte da formação acadêmica básica em Computação e áreas afins, neste sentido, diversas técnicas e ferramentas são concebidas para apoiar o ensino-aprendizagem de disciplinas de programação, como a utilização de jogos digitais. Tendo em vista os requisitos levantados por Silva et al. [24], o uso de jogos digitais no ensino é embasado pelo pensamento de serem eficazes ferramentas para o envolvimento dos alunos no processo ensino-aprendizagem. Com

base nisto, este trabalho propõe caracterizar um estudo a partir de um jogo digital no domínio de educação.

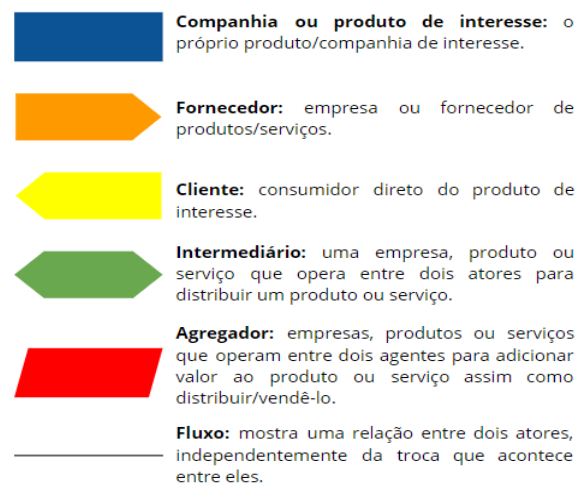


Figura 4: Componentes do Modelo SSN. Fonte: Costa et al. [29].

4.1 Modelando a Plataforma

Como o intuito desta plataforma de software é servir como base para auxiliar o processo ensino-aprendizagem de programação, a caracterização do cenário de ensino é importante.

Seguindo a linha de raciocínio de Santos et al. [10], este trabalho também será baseado na utilização de três personagens fundamentais para o quadro proposto, sendo eles: (1) o educador; (2) o monitor; e (3) o aprendiz. Todos têm papéis distintos no processo e possuem características próprias.

- O educador - possui a função de mediar e facilitar o processo ensino-aprendizagem, e.g. professor;
- O monitor - atua como colaborador do processo, complementando o educador, e.g. alunos mais experientes;
- O aprendiz - um conjunto de pessoas que têm em comum a assimilação do conteúdo como objetivo principal, e.g. alunos da disciplina em questão.

A arquitetura do ambiente é apresentada na Figura 5 e, para compreender o cenário, as relações entre os elementos são fundamentais e estão descritas a seguir:

- Jogo Digital / Aprendiz - a linha de comunicação do jogo digital com o aprendiz é unidirecional, onde a ferramenta em questão apresenta o conteúdo de forma lúdica ao aluno. O canal se dá através de alguma plataforma, e.g. internet determinada pelo próprio software;
- Aprendiz / Canal de Comunicação / Educador e Monitor - essa relação é extremamente importante para o cenário, pois através dela o aprendiz consegue expor seus conhecimentos e dúvidas ao educador/monitor. Por se tratar de um caminho bidirecional, também possibilita que o aluno receba um *feedback* dos tutores. A comunicação se dá a partir de um canal, que pode ser um sistema de gerenciamento, e.g. Moodle ou por maneiras não digitais;

- Educador e Monitor / Jogo Digital - esse trecho fica restrito à escolha do educador de como o jogo digital será utilizado para auxiliar o processo ensino-aprendizagem.

Como um ECOS é um conjunto de atores funcionando como uma unidade e interagindo em um mercado compartilhado de software e serviços centrado em uma plataforma comum, segundo Jansen et al. [26], faz-se necessária a abertura das fronteiras que limitam o Jogo Digital. Isto possibilita a integração com pessoas de fora que agreguem valor à plataforma e a evoluam com novas funcionalidades.

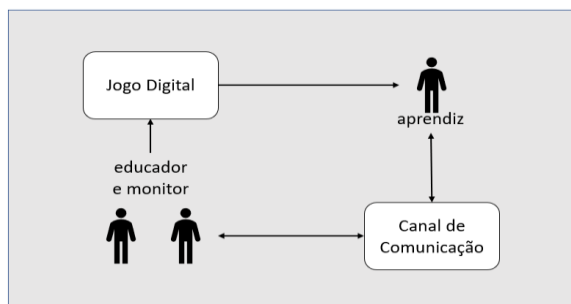


Figura 5: Arquitetura do Ambiente.

A Figura 6 expande o elemento Jogo Digital (Figura 5) e demonstra os possíveis atores principais por meio de um diagrama criado para o modelo SSN do Jogo Digital, conforme Boucharas et al. [28] e Costa et al. [29]. O diagrama (Figura 6) então contempla as seguintes agentes:

- Fornecedores - são os desenvolvedores que fornecem o produto de interesse. A idealização e concepção do serviço também parte deles. Nele, também está contido toda a parte de desenvolvimento (e.g. o *keystone*⁵, os desenvolvedores, os motores de jogos, os softwares utilizados no processo de criação);
- Produto de interesse - o produto de interesse do sistema é o próprio jogo digital em questão;
- Cliente - são os usuários do produto, no contexto do jogo digital, os educadores/aprendizes. Conectam-se com a plataforma através de intermediários;
- Intermediário - são um canal de integração da plataforma com os clientes determinado pelo próprio software (e.g. internet, aplicativo móvel);
- Agregadores - ator proposto por Costa et al. [27], em que um grupo de desenvolvedores externos aos fornecedores adicionam algum valor ao produto. O usuário tem a opção de utilizar o jogo simples ou de baixar as extensões disponibilizadas pelos intermediários. Nesse ponto, o jogo digital deixa de ser um software e passa a ser um ECOS.

A Figura 7 apresenta um fluxograma da utilização dos principais atores baseados no Jogo Digital com os usuários finais sendo o cenário proposto de ensino. No fluxograma, as relações e comunicações entre os principais atores do ECOS no domínio do ensino de programação são propostas. Os desenvolvedores externos também são acrescentados ao cenário, pois eles

produzem conteúdo para a comunidade, agregando valor à plataforma e tem funções similares aos dos agregadores.

Com o mapeamento dos principais atores que interagem com o Jogo Digital, apresentado como produto principal do ecossistema, torna-se importante caracterizar a plataforma em si, delimitando o escopo de cada ator. Delimitar o escopo permite representar como um ator interage com o núcleo da plataforma, sejam eles desenvolvedores internos ou externos. Para delimitar a plataforma, os elementos de jogos são importantes porque caracterizam o nível de abstração do Jogo Digital, permitindo assim entender como cada ator poderá interagir. A Figura 8 mostra a arquitetura da plataforma. O Núcleo e as Extensões são os dois principais segmentos encontrados dentro do limite do Jogo Digital. Os segmentos caracterizam como os atores poderão fornecer e/ou agregar valor à plataforma. A proposta deste trabalho é permitir que os elementos de jogos dinâmicos delimitem o escopo do núcleo e sejam providos pelos fornecedores e, como são definições de mais alto nível, e.g. a narrativa e as restrições, tais elementos devem ser encapsulados aos desenvolvedores externos.

Os elementos de jogos mecânicos devem ser delimitados pelos fornecedores, porém, estão dispostos nas extensões, visível aos agentes externos aos fornecedores. Os componentes serão os elementos de jogos que os desenvolvedores externos usarão para adaptar o contexto do jogo ao seu uso próprio. Por meio dos componentes será possível disponibilizar APIs, i.e. interface de programação de aplicativos e extensões para casos específicos do jogo digital.

4.2 Implementando a Plataforma

A Figura 9 exhibe um modelo de referência refinado e baseado em Pettersson et al. [30] que descreve os objetos do sistema e seus relacionamentos. A integração entre os componentes principais do cenário é representada por meio da Linguagem de Modelagem Unificada (UML). O jogo digital é composto pelo núcleo (N) e pelas extensões (E), no qual N é delimitado pelo fornecedor e é composto pelos elementos de jogos dinâmicos. Os fornecedores definem certos conceitos das extensões, compostas por elementos de jogos do tipo componente e mecânico, e possuem uma relação com os desenvolvedores externos (uma vez que são chave para tornar o sistema um ecossistema). Por último, os agregadores são uma especialização dos desenvolvedores externos e, por isso, interagem com o jogo da mesma forma que eles.

4.3 Projeto do Ecossistema

Após a idealização da plataforma, é importante definir um cenário para o seu desenvolvimento. Uma vez definidos os elementos de jogos, o fornecedor, através dos motores de jogos e outros softwares que possibilitam a construção da plataforma, o núcleo com seus limites e restrições deve ser desenvolvido.

Os fornecedores também vão delimitar as interfaces para as APIs, i.e. as extensões que os desenvolvedores externos terão acesso para alterar componentes e construir fases exclusivas para seu domínio. No núcleo estarão dispostos os elementos relacionados ao escopo do jogo (restrições e narrativas) permitindo que a essência do jogo não seja alterada. Nas extensões, estarão presentes os elementos de jogos visíveis aos desenvolvedores externos. Por meio delas será possível alterar elementos de jogos mais físicos, como aparência e contexto do jogo. Assim, o cenário do desenvolvedor externo mantém-se adaptável. Como exemplo, pode-se imaginar um jogo digital desenvolvido para ensinar comandos de lógica em uma determinada linguagem de programação (LP). Os usuários da plataforma podem utilizar o jogo dessa forma, estando sujeitos a

⁵ O termo *keystone* é usado para rotular os atores centrais em um ECOS [31].

aceitar a LP e os comandos impostos pela aplicação. Porém, por se tratar de um software aberto à interação e agregação de valores de terceiros, torna-se viável que usuários possam desenvolver uma API para o cenário proposto por eles, podendo assim: (1) alterar a LP utilizada; (2) alterar o comando de programação que está

sendo ensinado; e (3) alterar ambos, ou seja, um novo comando em uma nova LP. Sendo assim, o usuário também pode consumir o jogo em uma versão modificada para suprir as suas particularidades.

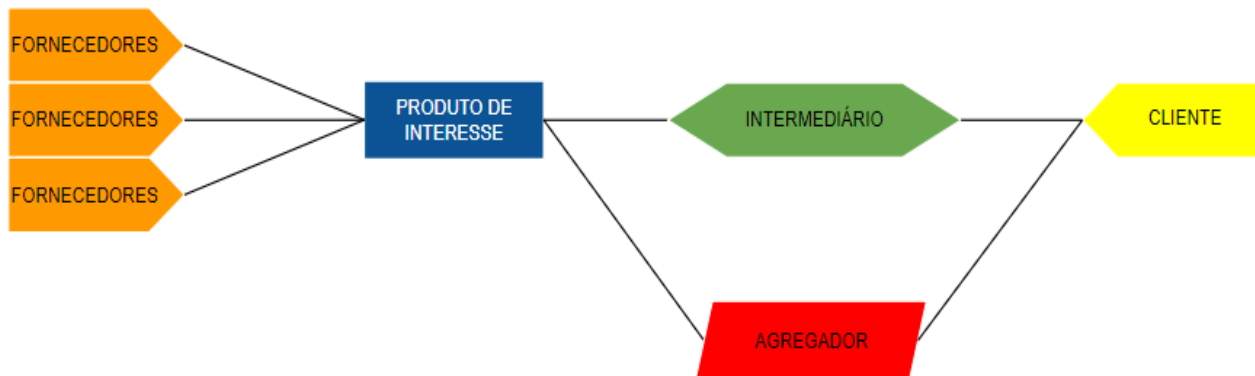


Figura 6: Modelo SSN do Jogo Digital.

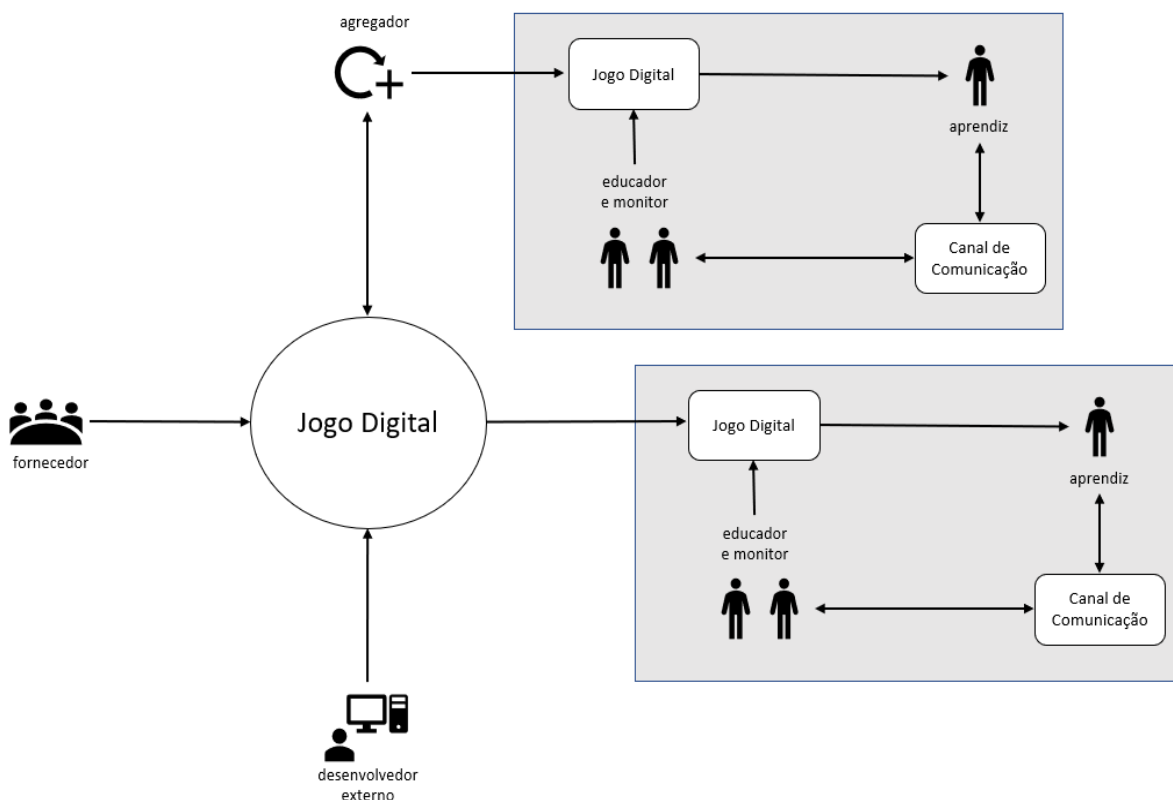


Figura 7: Ecossistema Baseado no Jogo Digital como Plataforma Comum.

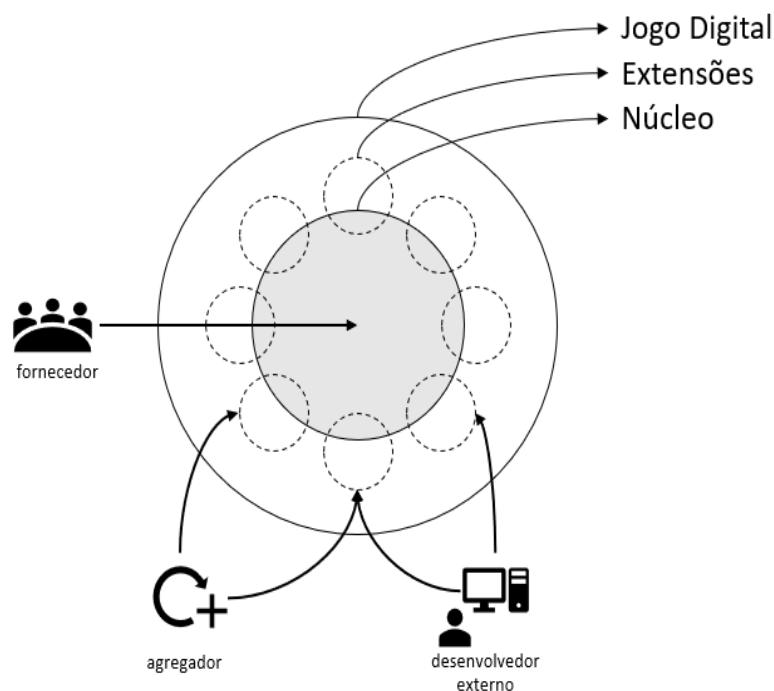


Figura 8: Arquitetura da Plataforma.

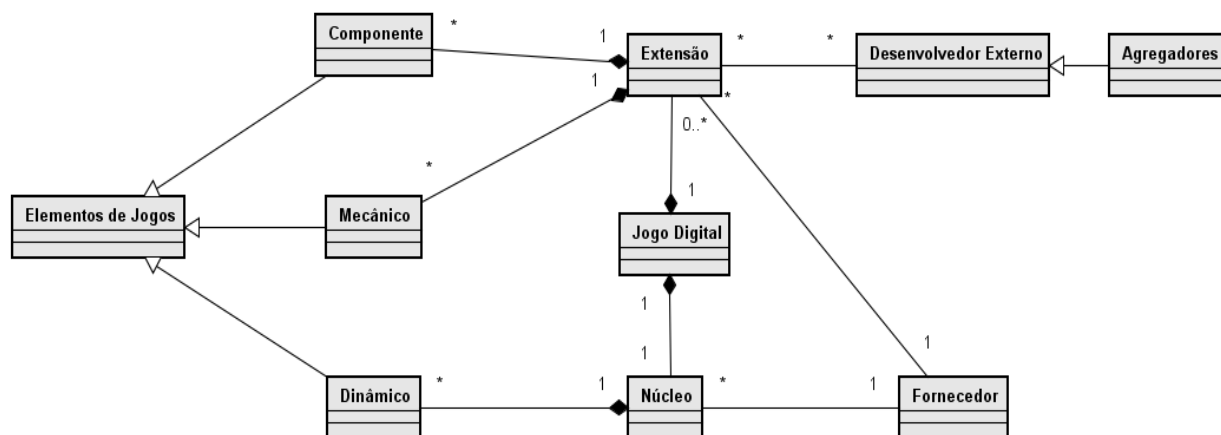


Figura 9: Diagrama de Classes da Plataforma.

4.3.1 Protótipo das Interfaces

Esta seção apresenta um protótipo que simula uma versão da tela do jogo digital proposto neste trabalho. Para a criação do protótipo, foi utilizado o software de design Balsamiq⁶.

A Figura 10 (A) representa a interface principal do jogo digital. Ela é composta por quatro componentes principais: (1) o menu; (2) a tela; (3) o nome da fase; e (4) as instruções. O menu, no canto superior esquerdo, apresenta algumas opções ao jogador (e.g. recomeçar a fase). Na imagem, são as ações padrão do Balsamiq para um menu (*File*, *Edit*, *View* e *Help*), não as opções que serão usadas no jogo em si. A tela será onde o jogo se

desenvolve e, sendo representada pelo quadro colorido, apresenta de maneira visual o contexto da fase. O nome da fase aparece no canto superior direito da interface e serve para indicar ao usuário em que estágio ele se encontra no jogo. As instruções são um conjunto de textos que esclarecem ao jogador a missão de determinada fase, ou seja, o que o usuário precisa cumprir para alcançar o objetivo. Ele também apresenta, de forma lúdica, o conteúdo a ser aprendido no determinado momento.

A proposta do jogo digital é permitir que haja interação entre desenvolvedores externos e a aplicação. Portanto, é necessária uma interface para a API que viabilize esta comunicação. A Figura 10 (B) representa a interface da API e por ela será possível desenvolver as extensões do jogo. Nela estão listados os principais

⁶ <https://balsamiq.com>

elementos de jogos disponíveis para o usuário modificar e adaptar o cenário ao seu contexto específico.

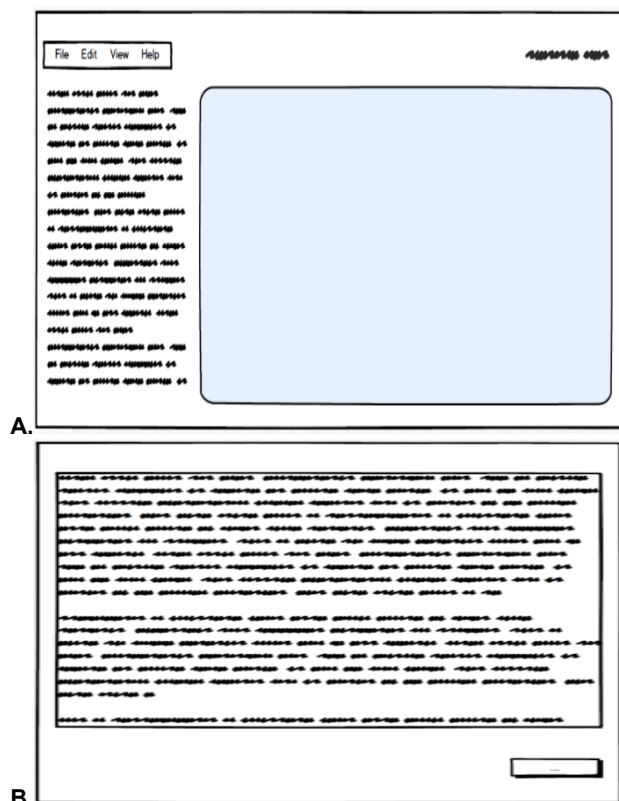


Figura10: Protótipo das Principais Interfaces do Jogo Digital.

5 CONCLUSÃO

Este trabalho teve como objetivo caracterizar uma plataforma para o desenvolvimento de um ECOS a partir de um jogo digital para o ensino de programação. Acredita-se que um jogo possui grande potencial no processo ensino-aprendizagem. Pretende-se, com o jogo, aumentar o engajamento dos alunos nas matérias de programação e reduzir a alta taxa de evasão apresentada por SEMESP [1]. A motivação para este trabalho foi criar um modelo de referência para: (1) cenários para o ensino de programação; (2) motores de jogos; (3) elementos de jogos; e (4) o processo de evolução de um jogo digital educacional para se transformar em plataforma de um ECOS.

O trabalho propôs uma progressão, utilizando notações de alto nível, o modelo SSN e um diagrama de classes, para, de um cenário de ensino de programação proposto, evoluir o esquema para um ECOS baseado no jogo digital como plataforma comum. Observou-se também que o estabelecimento da evolução do jogo para um ECOS se dá a partir do momento em que muitos atores diferentes começam a interagir e evoluir tal plataforma.

Considerando os requisitos e modelos elaborados, pretende-se escolher um motor de jogo que possibilite a construção do jogo digital, decidir os principais elementos de jogos que serão abordados no jogo e, além de evoluir os protótipos exibidos, desenvolver o cenário de ecossistema proposto.

REFERÊNCIAS

- [1] Sindicato das Entidades Mantenedoras de Estabelecimentos de Ensino Superior no Estado de São Paulo - SEMESP. Índice de Evasão de Alunos é maior na Área de Tecnologia da Informação. <http://www.semesp.org.br/imprensa/indice-de-evasao-de-alunos-e-maior-na-area-de-tecnologia-da-informacao-2/>, acessado em Jul/2018, Outubro de 2013.
- [2] J. I. Deters, J. Marques, J. M. C. Silva, E. M. Maschio, A. Maria e A. M. R. Fernandes. O Desafio de Trabalhar com Alunos Repetentes na Disciplina de Algoritmos e Programação, 2008.
- [3] A. Scaico e P. D. Scaico. Uso de Jogos em Cursos Introdutórios de Programação no Ensino Superior na Área de Computação: Uma Revisão Sistemática. *Anais do XXVII Simpósio Brasileiro de Informática na Educação* (Uberlândia, Minas Gerais, Outubro 24-27, 2016), SOCIEDADE BRASILEIRA DE COMPUTAÇÃO, pp. 549-558, Outubro de 2016.
- [4] V. V. Graciano Neto, R. Santos e R. Araújo. Sistemas de Sistemas de Informação e Ecossistemas de Software: Conceitos e Aplicações. In *Tópicos em Sistemas de Informação: Minicursos SBSI 2017*, SOCIEDADE BRASILEIRA DE COMPUTAÇÃO, pp. 22-41, Junho de 2017.
- [5] R. Santos. Ecossistemas de Software no Projeto e Desenvolvimento de Plataformas para Jogos e Entretenimento Digital. *Anais do XVI Simpósio Brasileiro de Jogos e Entretenimento Digital* (Curitiba, Paraná, Novembro 2-4, 2017), SOCIEDADE BRASILEIRA DE COMPUTAÇÃO, pp. 1327-1337, Novembro de 2017.
- [6] B. Boehm. A View of 20th and 21st Century Software Engineering. *Proceedings of the 28th International Conference on Software Engineering* (Shanghai, China, Maio 20-28, 2006) ACM/IEEE, pages 12-29, Maio de 2006.
- [7] B. Boehm e A. Jain. Developing a Process Framework using Principles of Value-Based Software Engineering: Research Sections. In *Software Process: Improvement and Practice v.12*, n.5, pages 377-385, Maio de 2007.
- [8] H. M. Deitel, P. J. Deitel. Java Como Programar. BOOKMAN COMPANHIA ED, 2003.
- [9] R. P. Santos, H. A. X. Costa. Análise de Metodologias e Ambientes de Ensino para Algoritmos, Estruturas de Dados e Programação aos Iniciantes em Computação e Informática. INFOCOMP, [S.l.], v. 5, n. 1, pp. 41-50, Março de 2006.
- [10] R. P. Santos, A. S. Vivacqua, J. M. Souza e H. A. X. Costa. Uma Proposta de Cenário para Ensino de Algoritmos e Programação com Contribuições de Cooperação, Colaboração e Coordenação. *Anais do XVI Workshop sobre Educação em Computação* (Belém, Pará, Julho 12-18, 2008), SOCIEDADE BRASILEIRA DE COMPUTAÇÃO, pp. 218-227, Julho de 2008.
- [11] G. F. Neto, B. L. E. Silva, R. R. Pereira e L. O. A. Júnior. Utilização do kit Lego Mindstorm NXT no Ensino de Controle de Processos. *Anais do XL Congresso Brasileiro de Educação em Engenharia* (Belém, Pará, Setembro 03-06, 2012), Setembro de 2012.
- [12] L. S. Vygotsky. Mind in Society: The Development of Higher Psychological Processes. Cambridge: Harvard University Press, pp. 52-91, Outubro de 1980.
- [13] S. P. Klapztein. Jogos e Gamificação como Ferramentas de Auxílio à Educação. *Anais do 10º Seminário Nacional de Educação a Distância* (São Paulo, São Paulo, Abril 24-26, 2014), ABED, Abril de 2014.
- [14] K. M. Kapp. The Gamification of learning and instruction: Game-based methods and strategies for training and education. PFEIFFER, Julho de 2012.
- [15] E. J. V. Sá, J. S. F. Teixeira e C. T. Fernandes. Design de atividades de aprendizagem que usam Jogos como princípio para Cooperação. *Anais do XVIII Simpósio Brasileiro de Informática na Educação* (Novembro 28-30, 2007), SOCIEDADE BRASILEIRA DE COMPUTAÇÃO, pp. 539-549, Novembro de 2007.
- [16] C. Hoefelmann. O uso dos jogos digitais educacionais no processo de ensino e aprendizagem, Universidade Federal de Santa Catarina, Florianópolis, Brasil, Outubro de 2016.
- [17] B. Uzunca e S. Jansen. How do Ecosystem Dynamics work in Serious Gaming Ecosystems? Challenges and Opportunities.

- Proceedings of 36th Annual Conference* (Berlim, Setembro 17-20, 2016), STRATEGIC MANAGEMENT SOCIETY, Setembro de 2016.
- [18] A. Protopsaltis, S. Hetzner, D. Pappa e L. Pannese. Serious Game for Formal and Informal Learning. *E-Learning Papers*, ISSN: 1887-1542, pp. 1-8, Julho de 2011.
 - [19] S. Iwata. GDC 2005: Iwata Keynote Transcript - IGN. <http://www.ign.com/articles/2005/03/11/gdc-2005-iwata-keynote-transcript?page=3>, acessado em Jul/2018, Março de 2005.
 - [20] BUNCHBALL INC. Gamification 101: an introduction to the use of game dynamics to influence behavior, 2010.
 - [21] K. Werbach e D. Hunter. For the win: how game thinking can revolutionize your business. WHARTON DIGITAL PRESS, Janeiro de 2012.
 - [22] A. C. S. Costa e P. Z. Marchiori. Gamificação, elementos de jogos e estratégia: uma matriz de referência. *Revista de Ciência da Informação e Documentação – InCID*, v. 6, n. 2, pp. 44-65, 2015.
 - [23] T. Lima, B. Campos, R. Santos e C. Werner. UbiRE: A game for teaching requirements in the context of ubiquitous systems. *Proceedings of XXXVIII Conferencia Latinoamericana En Informatica* (Medellin, Colômbia, Outubro 1-5, 2012), Outubro de 2012.
 - [24] R. R. Silva, J. C. Fernandes e R. P. Santos. Panorama da Utilização de Jogos Digitais no Ensino de Programação no Nível Superior na Última Década: Uma Revisão Sistemática da Literatura. *Anais do XXIX Simpósio Brasileiro de Informática na Educação* (Fortaleza, Ceará, Outubro 29-31, 2018) SOCIEDADE BRASILEIRA DE COMPUTAÇÃO, Outubro de 2018.
 - [25] R. P. Santos, G. Valença, D. Viana, B. Estácio, A. Fontão, S. Marczak, C. Werner, C. Alves, T. Conte e R. Prikladnick. Qualidade em Ecossistemas de Software: Desafios e Oportunidades de Pesquisa. *Anais do VIII Workshop de Desenvolvimento Distribuído de Software, Ecossistemas de Software e Sistemas de Sistemas* (Maceió, Alagoas, Setembro 28, 2014), Congresso Brasileiro de Software: Teoria e Prática, pp. 41-44, Setembro de 2014.
 - [26] S. Jansen, S. Brinkkemper e A. Finkelstein. Business Network Management as a Survival Strategy: A Tale of Two Software Ecosystems. *Proceedings of the First International Workshop on Software Ecosystems* (Falls Church, Virginia, Setembro 27, 2009), 11th INTERNATIONAL CONFERENCE ON SOFTWARE REUSE, pp. 34-48, Setembro de 2009.
 - [27] G. K. Hanssen e T. Dybå. Theoretical Foundations of Software Ecosystems. *Proceedings of the 4th International Workshop on Software Ecosystems* (Boston, EUA, Junho 18, 2012), CEUR-WS, pp. 6-17, Junho de 2012.
 - [28] V. Boucharas, S. Jansen e S. Brinkkemper. Formalizing Software Ecosystem Modeling. *Proceedings of the 1st International Workshop on Open Component Ecosystems* (Amsterdã, Holanda, Agosto 24-28, 2009), ACM SIGSOFT Symposium on the Foundations of Software Engineering, pp. 41-50, Agosto de 2009.
 - [29] G. Costa, F. Silva, R. Santos, C. Werner e T. Oliveira. From Applications to a Software Ecosystem Platform: An Exploratory Study. *Proceedings of the 5th International Conference on Management of Emergent Digital EcoSystems* (Luxemburgo, Luxemburgo, Outubro 28-31, 2013), pp. 9-16, Outubro de 2013.
 - [30] O. Pettersson, M. Svensson, D. Gil, J. Andersson e M. Milrad. On the Role of Software Process Modeling in Software Ecosystem Design. *Proceedings of IV European Conference on Software Architecture: Companion Volume* (Copenhagen, Denmark, August 23-26, 2010), pp. 103-110, Agosto de 2010.
 - [31] R. Ferreira, M. Miranda, C. R. B. Souza, F. F. Filho, C. Treude e L. Singer. Os Aspectos Sociais dos Ecossistemas de Software. *Anais do XII Simpósio Brasileiro de Sistemas Colaborativos* (Salvador, Bahia, Novembro 3-6, 2015), SOCIEDADE BRASILEIRA DE COMPUTAÇÃO, Novembro de 2015.