



**CENTRO UNIVERSITÁRIO SENAI BAHIA**

**PROGRAMA DE PÓS-GRADUAÇÃO EM MODELAGEM**

**COMPUTACIONAL E TECNOLOGIA INDUSTRIAL**

**Doutorado em Modelagem Computacional e Tecnologia Industrial**

**Tese de doutorado**

**Um modelo conceitual de Desenvolvimento de Jogos  
Digitais: sugestões e modificações no framework de  
desenvolvimento Scrum**

Apresentada por: Marcelo Vera Cruz Diniz

Orientador: Prof<sup>o</sup>. Dr. Roberto Luiz Souza Monteiro

Co-orientador: Prof<sup>a</sup>. Dr.<sup>a</sup> Tereza Kelly Gomes Carneiro

Julho de 2017



**CENTRO UNIVERSITÁRIO SENAI BAHIA**

**PROGRAMA DE PÓS-GRADUAÇÃO EM MODELAGEM**

**COMPUTACIONAL E TECNOLOGIA INDUSTRIAL**

**Doutorado em Modelagem Computacional e Tecnologia Industrial**

**Tese de doutorado**

**Um modelo conceitual de Desenvolvimento de Jogos  
Digitais: sugestões e modificações no framework de  
desenvolvimento Scrum**

Apresentada por: Marcelo Vera Cruz Diniz

Orientador: Prof<sup>o</sup>. Dr. Roberto Luiz Souza Monteiro

Co-orientadora: Prof<sup>a</sup>. Dr.<sup>a</sup> Tereza Kelly Gomes Carneiro

Julho de 2017

Marcelo Vera Cruz Diniz

# Um modelo conceitual de Desenvolvimento de Jogos Digitais: sugestões e modificações no framework de desenvolvimento Scrum

Tese de doutorado apresentada ao Programa de Pós-graduação em Modelagem Computacional e Tecnologia Industrial, Curso de Doutorado em Modelagem Computacional e Tecnologia Industrial do CENTRO UNIVERSITÁRIO SENAI BAHIA, como requisito parcial para a obtenção do título de **Doutor em Modelagem Computacional e Tecnologia Industrial**.

Área de conhecimento: Interdisciplinar

Orientador: Prof<sup>o</sup>. Dr. Roberto Luiz Souza Monteiro  
*C. U. SENAI Bahia*

Co-orientadora: Prof<sup>a</sup>. Dr.<sup>a</sup> Tereza Kelly Gomes Carneiro  
*UNCISAL*

Salvador  
C. U. SENAI Bahia  
2017

Ficha catalográfica elaborada pela Biblioteca do Centro Universitário SENAI CIMATEC

D585m Diniz, Marcelo Vera Cruz

Um modelo conceitual de desenvolvimento de jogos digitais: sugestões e modificações no framework de desenvolvimento scrum / Marcelo Vera Cruz Diniz. – Salvador, 2018.

103 f. : il. color.

Orientador: Prof. Dr. Roberto Luiz Souza Monteiro.

Tese (Doutorado em Modelagem Computacional e Tecnologia Industrial) – Programa de Pós-Graduação, Centro Universitário SENAI CIMATEC, Salvador, 2018.

Inclui referências.

1. Jogos digitais-design. 2. Metodologia-framework. 3. Modelo-framework. 4. Poker planning. 5. Scrum. I. Centro Universitário SENAI CIMATEC. II. Monteiro, Roberto Luiz Souza. III. Título.

CDD: 620.00113

# CENTRO UNIVERSITÁRIO SENAI BAHIA

Programa de Pós-graduação em Modelagem Computacional e Tecnologia Industrial

Doutorado em Modelagem Computacional e Tecnologia Industrial

A Banca Examinadora, constituída pelos professores abaixo listados, leram e recomendam a aprovação da Tese de doutorado, intitulada "Um modelo conceitual de Desenvolvimento de Jogos Digitais: sugestões e modificações no framework de desenvolvimento Scrum", apresentada no dia 21 de Julho de 2017, como requisito parcial para a obtenção do título de **Doutor em Modelagem Computacional e Tecnologia Industrial**.

Orientador:

---

Prof<sup>o</sup>. Dr. Roberto Luiz Souza Monteiro  
C. U. SENAI Bahia

Co-orientadora:

---

Prof<sup>a</sup>. Dr.<sup>a</sup> Tereza Kelly Gomes Carneiro  
UNCISAL

Membro interno da Banca:

---

Prof<sup>o</sup>. Dr. Hernane Borges de Barros Pereira  
C. U. SENAI Bahia

Membro externo da Banca:

---

Prof<sup>a</sup>. Dr.<sup>a</sup> Ingrid Winkler  
C. U. SENAI Bahia

Membro externo da Banca:

---

Prof<sup>o</sup>. Dr. Eduardo Manuel de Freitas Jorge  
UNEB

Membro externo da Banca:

---

Prof<sup>o</sup>. Dr. José Roberto de Araújo Fontoura  
UNEB

Enzo, Davi e Silvana. Obrigado pela oportunidade de passar uma encarnação ao lado de vocês.

---

## Agradecimentos

---

A minha amada esposa Silvana e aos meus filhos Enzo e Davi, pela paciência e apoio em todos os momentos que passei nos últimos 4 anos. Não consigo imaginar como seria a minha vida sem vocês, eu amo vocês. Aos meus pais Luciano e Isabela e meu irmão Rodrigo, pelo grande exemplo de vida que vocês me dão. A Daiane Morbeck, por construir uma família com meu irmão e trazer à Terra meu sobrinho Antônio. Aos meus padrinhos Itamar e Dinha, por tudo que vocês me ensinaram nos últimos anos. Ao meu cunhado e primo Anderson e a sua linda família: Camila, Nicole e Jullie. Adoro vocês. Agradeço por ter todos vocês em minha vida.

Aos meus orientadores Prof<sup>o</sup>. Dr. Roberto Luiz Souza Monteiro e Prof<sup>a</sup>. Dr.<sup>a</sup> Tereza Kelly Gomes Carneiro ; eu não tenho como expressar todo sentimento de gratidão que tenho por vocês. Aceitar orientar um aluno com cronograma de execução da tese atrasado e com restrições de tempo para dedicação ao curso foi um ato de coragem e de respeito ao próximo. Obrigado por acreditarem em mim e na minha proposta de trabalho.

Aos membros da minha banca Prof<sup>o</sup>. Dr. Hernane Borges de Barros Pereira, Prof<sup>a</sup>. Dr.<sup>a</sup> Ingrid Winkler, Prof<sup>o</sup>. Dr. Eduardo Manuel de Freitas Jorge e Prof<sup>o</sup>. Dr. José Roberto de Araújo Fontoura, por contribuírem para o aperfeiçoamento da minha tese. Espero um dia trabalhar com vocês em outros projetos.

A dois grandes amigos que ganhei durante os últimos anos, Prof. Dr. Dielson Pereira Hohenfeld, Prof. Dr. Jancarlos Meneses Lapa e a minha madrinha acadêmica Prof<sup>a</sup>. Ms.<sup>a</sup> Lucília Santa Rosa. Vocês resignificaram a relação que eu tenho com o IFBA.

Ao meu guia espiritual, por nunca desistir de mim e sempre em mostrar os caminhos da luz e da verdade. Obrigado meu irmão, espero um dia poder retribuir tudo que você fez por mim. Que a luz do GADU faça os nossos caminhos se cruzarem por muitas encarnações.

E, especialmente, gostaria de agradecer ao meu filho Enzo. O ano de 2016 foi cheio de trabalho, pesquisa e dedicação ao seu irmão por causa dos cuidados que um bebê necessita. Devido a isso, eu tive que dizer não muitas vezes para você, meu filho. Desculpe-me, espero que o Pai me dê vida e saúde para passar muito tempo com você, seu irmão e sua mãe.

Salvador, Brasil  
21 de Julho de 2017

Marcelo Vera Cruz Diniz

---

Enzo: Papai, conta uma história?

Eu: Conto papai. Qual você quer?

Enzo: A do rato jiu-jiteiro.

Eu: Certo ... Aiiiiiii ... Boa noite, meu filho

---

## Resumo

---

Este trabalho propõe, com base em um modelo que define o processo de desenvolvimento de Jogos Digitais, duas práticas que objetivam eliminar problemas causados pela individualização do trabalho, aumentar a cooperação entre os membros da equipe de desenvolvimento durante as reuniões de planejamento, e diminuir problemas causados por falhas nas estimativas das *user stories*. O modelo proposto contempla as diferentes perspectivas dos jogadores e desenvolvedores de Jogos Digitais, e apresenta uma estrutura que favorece a concepção de jogos. A partir do modelo proposto, apresentamos um protocolo para descrição de Jogos Digitais, que tem como principal objetivo aumentar o caráter científico desses artefatos que, normalmente, se apresentam como objetos focados para o entretenimento. A metodologia do estudo é de natureza qualitativa. Neste trabalho, mesclamos referências da área de designer de Jogos Digitais, Gamificação e protocolos que serviram como base estrutural das nossas sugestões. O resultado final dessa investigação possibilitou a construção de um modelo cuja aplicação potencializa a especialização de metodologias e frameworks de desenvolvimento de Jogos Digitais.

**Palavras-chave:** Desenvolvimento, Scrum, Jogos Digitais, *Poker Planning*, estimativa, Modelo, Metodologia, Ágil.

---

## Abstract

---

This thesis proposes, based on a model that defined the development process of Digital Games, two practices that aim to eliminate problems caused by Dualization of work, increase cooperation between members of the development team during planning meetings and reduce problems caused by wrong estimates of *user stories*. The proposed model contemplates the different perspectives of players and *Game Developers*, and presents a structure that favors conception of games. From the proposed model, we present a protocol for description of digital games, that has as its main objective increased the scientific value of these artifacts that are usually presented as objects focused on entertainment. The methodology of the study is qualitative. In this study, we used references from area like game designer of digital games, gamification and protocols that served as structural basis of our suggestions. The end result of this investigation presents a model whose application enhances the specialization of methodologies of Game Development.

**Keywords:** Development, Scrum, Digital Games, Poker Planning , Estimate, Model, Methodology, Agile.

---

# Sumário

---

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	Definição do problema . . . . .	1
1.2	Objetivo . . . . .	6
1.2.1	Objetivos específicos . . . . .	6
1.3	Limites e limitações . . . . .	6
1.4	Pressupostos . . . . .	7
1.5	Aspectos metodológicos . . . . .	8
1.6	Organização da Tese de doutorado . . . . .	12
<b>2</b>	<b>Os Jogos Digitais e as Metodologias de Desenvolvimento</b>	<b>14</b>
2.1	O que é um jogo? . . . . .	14
2.2	Por que gostamos dos Jogos Digitais ? . . . . .	17
2.3	A mecânica dos jogos . . . . .	19
2.4	Metodologias de Desenvolvimento de Jogos Digitais . . . . .	21
2.4.1	Os primeiros passos . . . . .	21
2.4.2	O framework Scrum . . . . .	25
2.5	Definindo os requisitos de um Jogo Digital . . . . .	30
2.6	Instrumentalização . . . . .	33
2.7	O protocolo ODD . . . . .	33
2.8	A pirâmide de elementos da gamificação . . . . .	37
2.9	O modelo MDA . . . . .	41
<b>3</b>	<b>Modelo de Desenvolvimento de Jogos Digitais</b>	<b>46</b>
3.1	O Modelo de Desenvolvimento de Jogos Digitais . . . . .	46
3.1.1	O objetivo . . . . .	46
3.1.2	As categorias e a relação entre os seus itens . . . . .	47
3.1.3	As perspectivas dos desenvolvedores e jogadores . . . . .	49
3.1.4	O protocolo para descrição dos Jogos Digitais . . . . .	50
<b>4</b>	<b>Sugestões para o Framework Scrum</b>	<b>54</b>
4.1	Apresentação . . . . .	54
4.2	O <i>Poker Planning</i> com jogadas colaborativas . . . . .	55
4.3	Verificação multidisciplinar . . . . .	60
4.4	Visualizando as sugestões no <i>framework</i> Scrum . . . . .	65
<b>5</b>	<b>Conclusões e Considerações finais</b>	<b>72</b>
5.1	Conclusões . . . . .	72
5.2	Contribuições diretas . . . . .	75
5.3	Contribuições indiretas . . . . .	75
5.4	Contribuições enquanto pesquisador . . . . .	77
5.5	Atividades Futuras de Pesquisa . . . . .	77
<b>A</b>	<b>Um exemplo de utilização do Protocolo para descrição de Jogos Digitais</b>	<b>78</b>
A.1	Apresentação do Jogo Digital Tetris . . . . .	78
A.2	Descrição do Tetris . . . . .	79

**Referências**

**83**

---

## Lista de Tabelas

---

1.1	Pressupostos e mecanismos de verificação . . . . .	7
2.1	As características da motivação intrínseca dos jogos . . . . .	19
2.2	Objetivos dos eventos no Scrum . . . . .	30
2.3	Elemento de PBL da trilogia Mass Effect e suas relações com as categorias da pirâmide de elementos de gamificação . . . . .	41
3.1	Blocos e seções do protocolo para descrição de Jogos Digitais . . . . .	53
4.1	As cartas do baralho de <i>Poker Planning</i> e seus significados . . . . .	57
A.1	Descrição do Jogo Digital Tetris . . . . .	80
A.2	Continuação da Tabela <a href="#">A.1</a> . Fonte: elaborado pelo autor . . . . .	81

---

## Lista de Figuras

---

1.1	Ciclo básico da Pesquisa-ação . . . . .	9
1.2	Principais autores estudados nesta tese . . . . .	11
1.3	Fluxo da pesquisa . . . . .	13
2.1	Principais sensações do indivíduo na realização de uma atividade até atingir o estado de <i>Flow</i> . . . . .	18
2.2	Metodologia de desenvolvimento em cascata . . . . .	22
2.3	Interação entre usuários e os diferentes tipos de artefatos (Estórias/Sistemas Computacionais e Jogos Digitais) . . . . .	25
2.4	<i>Framework</i> de desenvolvimento Scrum . . . . .	26
2.5	Personagens envolvidos no fluxo de trabalho do <i>framework</i> Scrum . . . . .	27
2.6	Eventos e artefatos do <i>framework</i> de desenvolvimento Scrum . . . . .	29
2.7	Estrutura de uma <i>user storie</i> . . . . .	31
2.8	Dois tipos de protótipo. . . . .	31
2.9	Etapas da metodologia Google <i>Sprint</i> . . . . .	32
2.10	Estrutura do protocolo ODD . . . . .	34
2.11	Comparação entre as duas versões do protocolo ODD. . . . .	37
2.12	Relação entre jogos, gamificação, brinquedos e designer lúdico . . . . .	39
2.13	Pirâmide de elementos de gamificação . . . . .	39
2.14	Elementos do processo de construção de um JD. . . . .	42
2.15	Fases do <i>design</i> iterativo . . . . .	43
2.16	Principais elementos de design sugeridos pelo <i>framework</i> MDA (Mecânica, Dinâmica e Estética) representados pelas letras M, D, A respectivamente . . . . .	45
3.1	Modelo de Desenvolvimento de Jogos Digitais (MDJD). . . . .	49
3.2	Protocolo para descrição de Jogos Digitais . . . . .	51
4.1	Cartas do baralho de <i>Poker Planning</i> . . . . .	57
4.2	Novas cartas para nova dinâmica do <i>Poker Planning</i> . . . . .	58
4.3	Exemplo de estimativa de uma equipe com três desenvolvedores . . . . .	59
4.4	Exemplo de estimativa após o <i>Showdown</i> . . . . .	60
4.5	Ciclo de Desenvolvimento guiado por testes, TDD . . . . .	61
4.6	Exemplo de Programação em pares . . . . .	62
4.7	Ciclo de Verificações Multidisciplinar . . . . .	63
4.8	Aplicação do <i>Poker Planning</i> Colaborativo . . . . .	66
4.9	Cenas do Jogo Digital LIPISpace . . . . .	67
4.10	<i>User stories</i> das cenas ilustradas na Figura 4.9 . . . . .	67
4.11	Jogadas para mensurar <i>user stories</i> . . . . .	68
4.12	Nova <i>User storie</i> criada para adicionar uma nova funcionalidade na cena "A" da Figura 4.9 . . . . .	68
4.13	Aplicação da Verificação Multidisciplinar . . . . .	69
4.14	Cenas do Jogo Digital SimGE . . . . .	69
4.15	Exemplo de utilização de uma das seções do Protocolo para descrição de Jogos Digitais . . . . .	70
4.16	Aplicação do Protocolo para descrição de Jogos Digitais . . . . .	71

---

A.1	<i>Game Boy</i> , console portátil da Nintendo lançado em 1989 . . . . .	79
A.2	As sete peças originais do Jogo Digital Tetris lançado em 1989 . . . . .	82
A.3	Conexão entre dois consoles <i>Game Boy</i> . . . . .	82

---

## Lista de Siglas

---

2D .....	Duas dimensões
3D .....	Três dimensões
ARG .....	Alternate Reality Games
CAPES ..	Coordenação de Aperfeiçoamento de Pessoal de Nível Superior
DOD .....	Definition of Done
EaD .....	Educação a Distância
GV .....	Google Ventures
IBM .....	Models based on individuals
IDE .....	Integrated Development Environment
IFBA ....	Instituto Federal de Educação, Ciência e Tecnologia da Bahia
INPI .....	Instituto Nacional da Propriedade Industrial
JD .....	Jogos digitais
LIPI .....	Laboratório Interdisciplinar de Práticas Inovadoras
MDA ....	Mechanics, Dynamics and Aesthetics
MDJD ...	Modelo de Desenvolvimento de Jogos Digitais
MEC ....	Ministério da Educação
MJ .....	Mecânica dos Jogos
MMO ....	Massive Multiplayer Online
NPC .....	Non-Playable Character
ODD .....	Overview, Design concepts, and Details
UML .....	Unified Modeling Language
PBL .....	Points, Badges e Leaderboards
ROI .....	Retorno do Investimento
SEBRAE	Serviço Brasileiro de Apoio às Micro e Pequenas Empresas
TDD .....	Test Driven Development
TIC .....	Tecnologias de Informação e Comunicação
XP .....	eXtreme Programming

## Introdução

---

Apresentamos, aqui, o caminho da pesquisa, a definição do problema, os objetivos, os pressupostos que a nortearam, os aspectos metodológicos, os limites e limitações do estudo e a organização da tese.

### *1.1 Definição do problema*

Desde a década de 1970, o mercado dos vídeo games observa a popularidade dos Jogos Digitais (JD) aumentar. Pesquisas de agências especializadas, como SEBRAE e PGT, apontam que o mercado mundial de JD movimentou US\$ 57 bilhões em 2010, enquanto o de cinema movimentou US\$ 31.8 bilhões. Em 2011, o setor movimentou US\$ 74 bilhões e no ano de 2015 as cifras ultrapassaram os US\$ 90 bilhões (SEBRAE, 2014). Para se ter uma noção mais precisa do impacto de um JD, em 2013, apenas o lançamento de Grand Theft Auto V teve o custo de US\$ 225 milhões e faturou US\$ 800 milhões em apenas 24 horas. De acordo com SEBRAE (2014), o jogo Angry Birds já foi instalado em mais de 500 milhões de celulares. Para o mercado brasileiro, em 2016 um montante de, aproximadamente, US\$ 1,4 bilhões foram movimentados, e as estimativas para 2017 apontam que o mercado brasileiro movimentará US\$ 1,7 bilhões (FLEURY; NAKANO; CORDEIRO, 2014). Especialistas estimam que em 2016 o mercado mundial de JD movimentou, aproximadamente, US\$ 86 bilhões e em 2017 os recursos movimentados por esse ramo da indústria do entretenimento vão ultrapassar os US\$ 106 bilhões com jogos voltados para dispositivos móveis como smartphones e tablets (NEWZOO, 2016).

Outro ponto importante que ajuda a explicar a evolução dos JD é o público-alvo. Esse tipo de entretenimento, antes voltado para o coletivo masculino, ganhou, e vem ganhando, espaço entre as mulheres e as famílias. Pesquisas realizadas entre 2013 e 2016 mostram o número crescente de mulheres que jogam vídeo games. Em 2013, elas eram 14% do público *gamer*. Em 2015, esse número subiu para 47,1% e em 2016 elas ultrapassaram os homens e, atualmente, são 52,6% do público consumidor dos JD (BRASIL, 2015). Além disso, os dados das pesquisas nos apresentam um fato interessante: 85,8% dos pais costumam jogar com os seus filhos. Ou seja, os JD já se apresentam como um entretenimento familiar (BRASIL, 2015; BRASIL, 2016). Esse caso de estudo se explica porque a maioria dos jogadores, 62%, estão em idade adulta, entre 25 e 54 anos (BRASIL, 2016), fato que contribui para perpetuação da cultura dos vídeo games de uma geração para outra.

Questões tecnológicas também influenciaram a indústria e o público-alvos dos JD. Antes focados apenas em consoles e computadores, os JD estavam restritos a essas plataformas, isso limitava a abrangência desse artefato devido às limitações e necessidades operacionais desses equipamentos. Porém o surgimento de novas tecnologias e dos dispositivos móveis, expandiu as opções de plataformas computacionais (FLEURY; NAKANO; CORDEIRO, 2014). Devido a isso, atualmente podemos encontrar JD em computadores, smartphones, tablets, consoles e Smarts Tvs. A plataforma mais utilizada pelos jogadores brasileiros é o smartphone, 77.2% dos usuários. Logo após vem o computador, com 66.9%, os consoles, que é a escolha de 58.6% dos jogadores, os tablets, 24.7%, e por último as smarts Tv, com 10.1% dos usuários (SEBRAE, 2014; NEWZOO, 2016; BRASIL, 2015; BRASIL, 2016).

Esses três fatores (movimentação financeira, público-alvo e fatores tecnológicos) criaram muitas oportunidades de negócio devido à pluralidade do público-alvo e à grande aproximação que os atuais consumidores de JD têm das Tecnologias de Informação e Comunicação (TIC), fato que proporcionou o crescimento vertiginoso do mercado de JD e o fortalecimento de estúdios de desenvolvimento de games de pequeno, médio e grande porte (MCGONIGAL, 2011; MASTROCOLA, 2015).

O processo de desenvolvimento de JD é um assunto que singulariza grande parte dos *games developments* devido ao produto final que esse processo objetiva construir. De acordo com Crawford (1984), os JD representam o mundo real a partir de duas perspectivas que não são excludentes: a subjetiva e a objetiva. A perspectiva subjetiva brota e se alimenta da perspectiva objetiva. Esse ciclo se sustenta por causa de uma peculiaridade que os JD possuem, que é a de concretizar anseios e desejos humanos através das experiências que o jogador vivencia durante o jogo. Devido a essa característica, não podemos considerar os JD como simplesmente *softwares*. Os JD são mais do que *softwares* (KEITH, 2010).

Conseqüentemente, os JD não devem seguir os mesmos padrões e metodologias utilizados para o desenvolvimento de softwares, objetivando evitar insucessos de planejamento, como desperdício de tempo, esforço e dinheiro, ou fracassos comerciais como construir jogos que não vão atender às demandas do seu público-alvo (SCHELL, 2014; SALEN; ZIMMERMAN, 2012).

As metodologias de desenvolvimento mais indicadas para a construção de JD são as metodologias interativas (SCHELL, 2014; FULLERTON, 2014). É possível utilizar metodo-

logias lineares como, por exemplo, a metodologia cascata (ROYCE, 1970). Porém essa metodologia só é bem aplicada para o desenvolvimento de jogos simples, como um jogo de cartas ou até mesmo um JD pequeno. Nesses casos, é admissível pensar em todo planejamento e estratégias de gamificação antes de começar o desenvolvimento do jogo. Porém, em jogos de médio ou grande porte, cujos protótipos necessitam de dias para serem construídos, devido ao trabalho intenso de arte e programação, o processo de construção é necessariamente iterativo e, além disso, não é possível definir quantas iterações serão necessárias para conclusão satisfatória do produto (SCHELL, 2014).

O uso das metodologias Ágeis para o desenvolvimento de JD tornou-se uma prática muito comum devido a características como: desenvolvimento incremental, cooperação e adaptação (GODOY; BARBOSA, 2010). Além dessas propriedades, encontramos, no conjunto de princípios que norteiam as metodologias Ágeis, valores que evidenciam o produto final como o foco principal dessas metodologias, e não o processo. Os princípios que norteiam as metodologias Ágeis foram sumarizados em 2001 por um conjunto de desenvolvedores experientes, e deu origem ao manifesto Ágil, que valoriza a interação entre os indivíduos, a colaboração com o usuário final e a prototipagem (ÁGIL, 2011).

Esse conjunto de propriedades possibilita que *frameworks* como Lean, XP e Scrum sejam bem indicados para o desenvolvimento de JD (KEITH, 2010). Neste trabalho, focamos a nossa atenção no *framework* Scrum. Iremos sugerir adaptações em dois eventos importantes do Scrum, *Sprint Planning* e *Sprint Execution*. Essas sugestões nascem da necessidade de diminuir, ou eliminar, problemas causados por estimativas malfeitas no momento da definição do escopo dos requisitos dos JD e pela individualização do trabalho.

Embora muitos desenvolvedores já tenham experiência no processo de desenvolvimento de sistemas, e até na utilização do Scrum como processo de desenvolvimento, existem especificidades e características peculiares ao processo de criação de JD que são cruciais para construção de jogos bem-sucedidos (GODOY; BARBOSA, 2010). Devido a isso, o aperfeiçoamento de metodologias e práticas voltadas para o desenvolvimento de JD podem evitar problemas de planejamento, reduzir os custos de desenvolvimento e evitar atrasos nas entregas e no lançamento dos jogos (GREGORY, 2010).

De acordo com Keith (2010), Scrum é um *framework* para o desenvolvimento de produtos complexos. Não é uma metodologia porque não possui um conjunto de práticas que determinam o que os desenvolvedores têm que fazer. Esse *framework* impulsiona o desenvolvimento através de uma metodologia incremental e interativa, auto gerenciável,

multidisciplinar e cooperativa.

O Scrum possui um conjunto de eventos (*Sprint Planning*, *Sprint Execution*, *Daily Scrum*, *Sprint Review e Retrospective*) e artefatos (*Product Backlog*, *Sprint Backlog* e Incremento ou Produtos) que alicerçam as suas bases fundamentais. Desde que esses atributos e os princípios da metodologia Ágil sejam mantidos, é possível fazer adaptações no Scrum para a realidade de sua empresa ou projeto (KEITH, 2010; SUTHERLAND, 2016).

Petrillo *et al.* (2008) afirma que os principais problemas encontrados durante o desenvolvimento de JD estão relacionados a falhas de planejamento, definição do escopo dos jogos e gerenciamento de problemas causados pela característica multidisciplinar da equipe de desenvolvimento. Embora equipes compostas por membros de diferentes áreas do conhecimento proporcionem um ambiente criativo, as características laborais das profissões terminam dividindo o grupo entre programadores e artistas, fato que dificulta a comunicação entre os membros da equipe, favorece a individualização do trabalho e prejudica a busca de um estado de equilíbrio entre os aspectos artísticos e técnicos do JD, geralmente chamado pela literatura de *fun of the game* ou *find the fun*, no qual os desejos e anseios do público-alvo, os jogadores, são atendidos (HUNICKE; LEBLANC; ZUBEK, 2004; PETRILLO *et al.*, 2008; KANODE; HADDAD, 2009; GODOY; BARBOSA, 2010; KEITH, 2010).

Devido a isso, tendo como premissa que os JD são artefatos cujo escopo superam os sistemas computacionais e que o trabalho colaborativo de uma equipe multidisciplinar é um fator indispensável para o desenvolvimento de JD, o problema que esta tese pretende resolver é: Como adaptar o *framework* de desenvolvimento Scrum, para evitar problemas de gestão causados por falhas na definição de escopo dos requisitos dos JD e pela individualização do trabalho durante o processo de desenvolvimento de JD?

Assim, o presente estudo defende a necessidade de formalizar práticas colaborativas para definição do escopo dos requisitos dos JD e para evitar a individualização do trabalho no desenvolvimento de JD. Para tanto, utilizamos o protocolo ODD (GRIMM *et al.*, 2006; GRIMM *et al.*, 2010), a Pirâmide de Elementos da Gamificação (WERBACH; HUNTER, 2012) e o modelo MDA (HUNICKE; LEBLANC; ZUBEK, 2004) como base para construção de um modelo que define como funciona o processo de desenvolvimento de um JD, considerando a perspectiva dos jogadores e dos desenvolvedores, o Modelo de Desenvolvimento de Jogos Digitais, MDJD. A partir desse modelo, apresentamos um protocolo para descrição de JD que pode ser utilizado como um documento de *design* ou como uma

ferramenta para aumentar o caráter científicos de *serious games*, JD que tem objetivos focados na educação, informação e treinamento (ABT, 1987; MICHAEL; CHEN, 2005).

Além disso, apresentamos duas sugestões que objetivam adaptar o *framework* Scrum para o desenvolvimento de JD. A primeira modifica a prática gamificada *Poker Planning* (GRENNING, 2002), e tem o propósito de aumentar a participação de todos os membros da equipe de desenvolvimento durante o processo de definição do escopo dos requisitos dos JD. A segunda sugestão é novo item de verificação no documento de *Definition of Done*, definição de pronto, (DOD). Ela se chama Verificação Multidisciplinar, e tem o intuito de evitar a individualização do trabalho e a perda de aspectos multidisciplinares que foram projetados durante a reunião de planejamento do *sprint*. Ela foi idealizada com base em duas características fundamentais da metodologia *eXtreming Programming* (XP), a Programação em pares e o Desenvolvimento guiado por teste (TDD) (BECK, 2009) e coloca os jogadores - os usuários finais dos JD - dentro do processo de desenvolvimento dos jogos.

O ineditismo deste trabalho está nos quatro produtos aqui apresentados. Durante a pesquisa bibliográfica que efetuamos para efetivação desta pesquisa, encontramos trabalhos que apresentam adaptações para gestão de projetos de desenvolvimento de JD (GODOY; BARBOSA, 2010; SCHILD; WALTER; MASUCH, 2010; ALBINO; SOUZA; PRADO, 2014), trabalhos com o foco voltado para colaboração da equipe de desenvolvimento (MUSIL *et al.*, 2010), soluções para trabalhar de forma distribuída durante o processo de mensuração do tamanho dos requisitos (MOLØKKEN-ØSTVOLD; HAUGEN; BENESTAD, 2008; ORACLE, 2014), trabalhos que objetivam unir as estimativas de diferentes desenvolvedores experientes (BLESS, 2010) e trabalhos que têm o objetivo de descrever JD (HENSE; MANDL, 2012; PETRY *et al.*, 2013). Porém não encontramos modelos, protocolos e práticas que solucionem o problema que estamos tratando da forma que nós o apresentamos.

Além disso, destacamos que as práticas que apresentamos nesta tese não criam ou adicionam novas estruturas ao *framework* Scrum. E, devido a isso, elas podem ser utilizadas por qualquer equipe de desenvolvimento sem ferir a atual cultura de trabalho do time de desenvolvedores.

Sendo assim, o presente trabalho defende que a aplicação de um modelo que relacione a equipe de desenvolvedores como os usuários finais, a partir das suas diferentes perspectivas, viabilizará a gestão mais eficiente do processo de desenvolvimento de Jogos Digitais.

A presente pesquisa se alinha com a proposta de trabalho do PPG MCTI visto que ela busca estudar, compreender e aprimorar a estrutura do processos de desenvolvimento de JD mediado por modelagem. A estratégia utilizada nesta pesquisa permite a reflexões e a construção de contribuições teórico-metodológicas para o aprimoramento de um processo industrial.

## 1.2 *Objetivo*

Propor adaptações no *framework* Scrum, tendo como base um modelo que define o processo de desenvolvimento de JD e práticas colaborativas, a partir da perspectiva dos jogadores e dos desenvolvedores.

### 1.2.1 *Objetivos específicos*

- Analisar fragilidades no processo de desenvolvimento de JD;
- Construir um modelo que defina o processo de desenvolvimento de JD, a partir da perspectiva dos jogadores e dos desenvolvedores;
- Elaborar adaptações no *framework* Scrum baseadas nas fragilidades encontradas no processo de desenvolvimento de JD.

## 1.3 *Limites e limitações*

O primeiro limite admitido nesta pesquisa é que os JD, cujos processos de desenvolvimento foram analisados para construção das sugestões, foram JD educacionais de pequeno porte e sem fins lucrativos.

O segundo limite é que ainda não aplicamos os produtos gerados nesta pesquisa (Modelo, protocolo, e as duas sugestões de adaptação do *framework* Scrum) no desenvolvimento de um JD. Esse conjunto de artefatos serão validados por profissionais e pesquisadores da área em um dos trabalhos futuros após esta pesquisa.

O terceiro limite foi que, durante o processo de desenvolvimentos dos dois JD, cujo processos foram analisados, não realizamos ou utilizamos todos os eventos e artefatos do

*framework* Scrum. Nós adaptamos o processo para realidade da empresa e das equipes de desenvolvimento. Nos dois projetos realizamos *Sprints* de três semanas, reuniões semanais (sempre nas segundas às 9 horas da manhã), *Sprint review*, *Product backlog* e *Sprint backlog*.

## 1.4 Pressupostos

De acordo com [Carneiro \(2015\)](#) os pressupostos funcionam como uma linha-guia de uma pesquisa empírica. Considerando que os JD são objetos de estudo que transcendem o escopo dos sistemas computacionais convencionais e possuem especificidades que precisam ser consideradas durante o seu desenvolvimento ([HUNICKE; LEBLANC; ZUBEK, 2004](#); [KEITH, 2010](#)), foram formulados três pressupostos, apresentados na Tabela 1.1, que nortearam todo o desenvolvimento do estudo. Para cada pressuposto, foi determinado um mecanismo de verificação e operacionalização.

Pressupostos	Mecanismo de verificação (A+B)	
	Procedimento (A)	Operacionalização (B)
1. Um modelo que relacione e aproxime o jogador do processo de desenvolvimento de JD é um fator preponderante para construção de experiências emocionais engajadoras.	Estudo teórico sobre Modelos, Jogos Digitais e Gamificação.	Análise dos dados e construção de um modelo.
2. Existem especificidades no processo de desenvolvimento de JD que precisam ser tratadas para evitar fracassos de planejamento e a construção de jogos enfadonhos.	Estudo teórico sobre processos de desenvolvimento de jogos digitais.	Análise dos dados e identificação dos principais problemas relatados durante o desenvolvimento de Jogos Digitais.
3. A colaboração é o ponto central para efetivação do projeto com equipes multidisciplinares.	Estudo teórico sobre o <i>framework</i> Scrum.	Análise dos eventos e artefatos do Scrum e definição das sugestões de modificação.

Tabela 1.1: Pressupostos e mecanismos de verificação. Fonte: Adaptado de [Carneiro \(2015\)](#).

O primeiro pressuposto apresentado na Tabela 1.1 está relacionado com o atores envolvidos no desenvolvimento de JD e com os conceitos que norteiam esse processo. Como a incerteza é um parâmetro presente em grande parte do processo de desenvolvimento de JD, a inclusão do jogador no processo de construção dos JD viabiliza a definição de uma linha guia para equipe de desenvolvimento. Com isso, poderemos definir um modelo que apresente os principais conceitos do desenvolvimento de JD, considerando as perspectivas dos jogadores e desenvolvedores.

Posteriormente, apresentamos um pressuposto que expõe a nossa preocupação com as especificidades do processo de desenvolvimento de JD. O maior objetivo do segundo pressuposto é evitar fracassos dos JD. Para isso, realizaremos um estudo teórico sobre processos de desenvolvimento de JD e, a partir desse estudo e da análise dos dados, identificaremos um dos principais problemas encontrados durante o desenvolvimento de JD.

Por fim, apresentamos o terceiro pressuposto e nesta suposição falamos sobre um ponto importante para o desenvolvimento de JD, a colaboração. Chamamos a atenção para esse ponto por que existem aspectos multidisciplinares que são fundamentais para construção de JD bem sucedidos. Geralmente, os conceitos multidisciplinares são perdidos por mensurações mal feitas e falta de comunicação durante o desenvolvimento do JD. O estudo teórico do *framework* de desenvolvimento Scrum possibilitará a apresentar propostas que tem o objetivo de eliminar problemas caudados por essas falhas.

## 1.5 Aspectos metodológicos

A proposta metodológica aplicada nesta pesquisa foi a pesquisa-ação. Essa escolha se justifica, devido a natureza dessa pesquisa. De acordo com [Thiollent \(2011\)](#), a Pesquisa-ação é um tipo de pesquisa de base empírica que é concebida e executada com estreita relação com ação ou com a solução de um problema no qual o pesquisador esta envolvido.

Os objetos de estudo desta pesquisa foram os processos de desenvolvimento de dois JD. O SimGe <sup>1</sup>, foi desenvolvido durante a Coordenação do Programa Profucionário, uma ação do Instituto Fedaral da Bahia, IFBA, em parceria com o Ministério da Educação, MEC, que ofertou 3000 vagas em quatro cursos técnicos na modalidade EaD para trabalhadores das escolas pública do Estado da Bahia. O segundo JD foi o LIPISpace <sup>2</sup>, desenvolvido no Laboratório Interdisciplinar de Práticas Inovadoras, LIPI, e tem o foco voltado para o ensino da Física Moderna no Enino Médio. Esses JD foram desenvolvidos entre os meses de Fevereiro de 2013 e Julho de 2015. Em nenhum momento os membros das equipes de desenvolvimento foram analisados, todas as observações foram feitas sobre os processos de desenvolvimento dos JD.

De acordo com [Westbrook \(1995\)](#), [Coughlan & Coughlan \(2002\)](#), [Tripp \(2005\)](#), [Thiollent](#)

---

<sup>1</sup>Jogo Digital Educacional registrado no INPI: Número de Registro: 512015000928-1

<sup>2</sup>Jogo Digital Educacional registrado no INPI: Número de Registro: 512016001116-5

(2011) o ciclo básico da Pesquisa-ação tem quatro fases (Planejar, Agir, Descrever e Avaliar). Porém, para apresentar mais detalhadamente o percurso metodológico realizado durante esta pesquisa, utilizaremos um conjunto de cinco fases apresentado em [Mello et al. \(2012\)](#), ilustrado na Figura 1.1, que adaptou o ciclo básico da Pesquisa-ação a partir das quatro referências citadas acima.

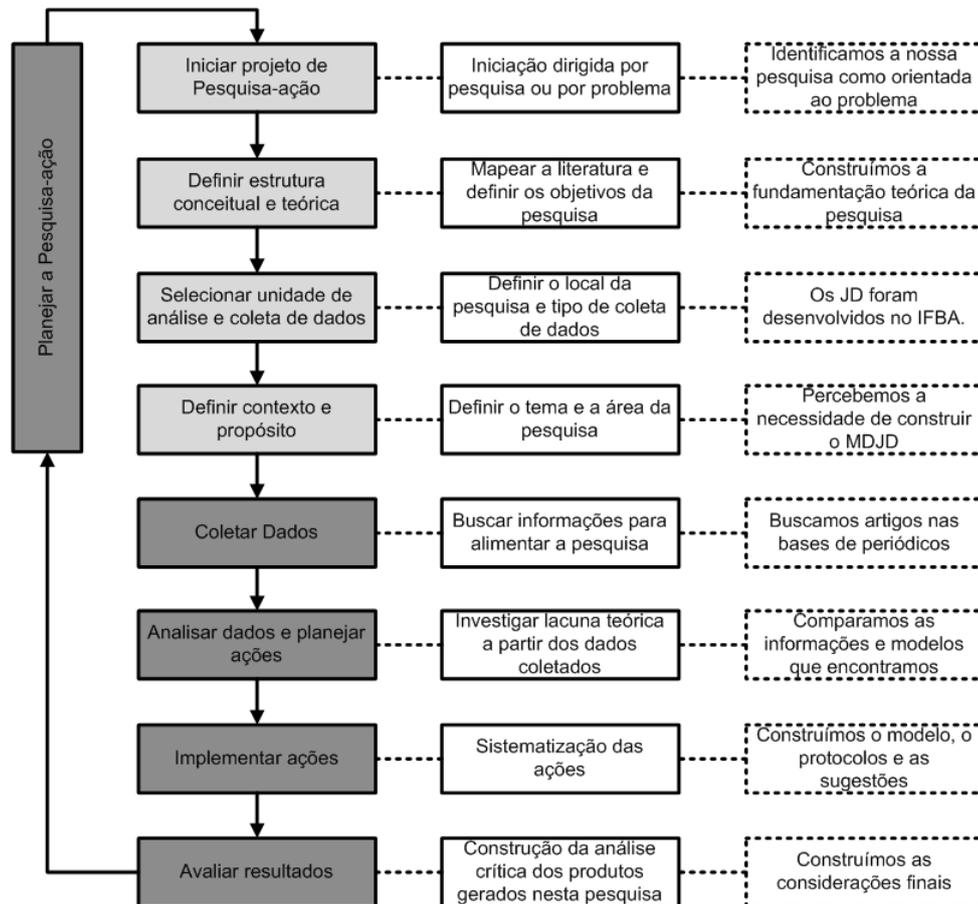


Figura 1.1: Ciclo básico da Pesquisa-ação. Fonte: Adaptado de ([MELLO et al., 2012](#))

A fase de **Planejamento da Pesquisa-ação** é composta de quatro etapas. A primeira é *Iniciar o projeto de Pesquisa-ação* e pode ser efetivada por dois tipos diferentes de projetos: projetos orientados à pesquisa ou projetos orientados a problemas. Nos projetos orientados a pesquisa, o pesquisador inicialmente realiza uma pesquisa bibliográfica e encontra lacunas que causam problemas para área de estudo da sua pesquisa. Posteriormente, o pesquisador irá definir cenários que serão utilizados para propor soluções para o problema encontrado ([MELLO et al., 2012](#)).

A nossa pesquisa se enquadra na segunda possibilidade: trata-se da iniciação dirigida por problema. Nessa abordagem, os integrantes de uma organização ou de um grupo de trabalho se deparam com um problema e um especialista se predispõe a resolvê-lo ([MELLO](#)

*et al.*, 2012). A peculiaridade desta pesquisa é que o pesquisador vivenciou o problema, os processos de desenvolvimento dos JD SimGe e LIPISpace, e depois efetuou as análises necessárias para propor as sugestões que fizemos nesta pesquisa

A segunda etapa da fase de Planejamento da Pesquisa-ação é *definir a estrutura conceitual e teórica*. Nesta fase, nós construímos a fundamentação teórica da pesquisa. Definimos o Estado da Arte do campo de pesquisa que estávamos estudando a partir da bibliografia clássica e das obras mais recentes da área. Durante esta fase, identificamos o conjunto de conceitos e publicações que se alinhavam com as lacunas que observamos durante o desenvolvimento dos JD SimGE e LIPISpace. Neste momento, nós construímos os objetivos (geral e específicos) desta tese e definimos os três pressupostos que nos guiaram durante a execução desta pesquisa.

A terceira etapa da primeira fase do ciclo de Pesquisa-ação é *selecionar unidade de análise e coleta de dados*. Nesta etapa, precisamos responder duas perguntas importantes: Onde a Pesquisa-ação será realizada (local)? e Como será realizada a coleta de dados? Como a iniciação da nossa pesquisa foi motivada por um problema, o local da pesquisa já estava definido. Os dois JD - cujos processos de desenvolvimento foram, devidamente, analisados - foram construídos no Instituto Federal da Bahia (IFBA). Não aplicamos questionários para e não fizemos entrevistas.

A última etapa da fase de Planejamento da Pesquisa-ação é *definir contexto e propósito*. Seguindo as definições apresentadas em Thiollent (2011), nós definimos o tema e a área de atuação desta pesquisa nesta etapa. Aqui percebemos a necessidade de construir ou definir modificações em um modelo que obtivesse características estruturantes devido à natureza da atividade de desenvolvimento de JD. Foi neste ponto que percebemos que o *Framework Scrum* seria uma estrutura importante dentro do andamento desta tese.

Na fase de **Coleta de Dados**, utilizamos as bases de periódicos online PUBME, Scilo, *Web of Science*, Google Acadêmico, Thomson Reuters e Academia.edu, além do Banco de Teses da CAPES. Em nenhuma das bases de conhecimento pesquisadas, foram utilizados critérios de recorte temporal. O nosso principal objetivo era encontrar todas as publicações que trataram desse assunto. Para isso, utilizamos palavras-chave pertinentes à temática definida e algumas variações e correlações com outros temas de pesquisa como: *game designer*, *game development*, desenvolvimento de jogos digitais, engenharia de software, *software engeniaring*, gamificação e *gamification*.

Vale destacar que, além das bases de conhecimento utilizadas nesta pesquisa, utilizamos também livros clássicos amplamente utilizados e referenciados da área. Além disso, afirmamos que, embora muitas vezes o Google Acadêmico e Academia.edu sejam contestados pela comunidade acadêmica, no contexto desta pesquisa eles foram utilizados para ampliar a abrangência da investigação em busca de publicações sobre o tema estudado.

O conjunto de trabalhos que encontramos foi organizado em seis categorias: Jogos, Jogos Digitais, *Game Designer*, Gamificação, Modelagem e Scrum. Essa divisão facilitou o trabalho exploratório e possibilitou a identificação dos limites teóricos que existem entre as diferentes áreas que estávamos pesquisando. A Figura 1.2 apresenta os principais autores pesquisados, segundo a sua contribuição para este trabalho.



Figura 1.2: Principais autores estudados nesta tese. Fonte: elaborado pelo autor

Depois de definir o conjunto de artigos e autores que formariam o universo de obras que sustentariam os nossos modelos e sugestões, iniciamos a fase de **Análise de dados e Planejamento de ações**. Durante essa etapa, nós comparamos o conjunto de informações e modelos que encontramos. Logo, percebemos a convergência entre as observações que fizemos durante o desenvolvimento dos JD, SimGE e LIPISpace, e a literatura que encontramos. A lacuna teórica que nós enxergamos no decorrer da construção dos JD

ficou evidente e, a partir dessa constatação, começamos a projetar as ações que seriam necessárias para sanar essas falhas e evitar fracassos no desenvolvimento de JD.

Em seguida, iniciamos a fase de **Implementar ações** e sistematizamos as sugestões que adicionaram duas práticas colaborativas, voltadas para a construção de JD, no *framework* de desenvolvimento Scrum. Por fim, analisamos os resultados obtidos neste projeto e concluímos a pesquisa, expondo os trabalhos futuros que podem dar continuidade a este projeto de pesquisa. Os resultados parciais dessa tese foram publicados em artigos científicos que estão disponíveis nos anexos deste trabalho.

Por fim, nós **Avaliamos os resultados** obtidos, fazendo uma reflexão sobre todos os produtos gerados por esta tese (o modelo, o protocolo e as sugestões) e construímos as considerações finais nesta pesquisa.

## 1.6 Organização da Tese de doutorado

Este documento está estruturado em cinco capítulos, os quais estão conectados em função do objetivo da pesquisa doutoral. Destaca-se que todo o processo foi interconectado e mesmo que aqui seja apresentado como um fluxo contínuo com início, meio e fim, como ilustrado na Figura 1.3, esta construção não ocorreu de forma linear. Na verdade, foi uma construção onde se permitiu constantes visitas e revisitas aos capítulos inicialmente propostos, e onde a revisão bibliográfica permeou toda a construção do texto.

- **Capítulo 1 - Introdução:** Contextualiza o âmbito, no qual a pesquisa proposta está inserida. Apresenta a definição do problema, objetivos e justificativas da pesquisa;
- **Capítulo 2 - Os Jogos Digitais e as Metodologias de Desenvolvimento:** Apresenta os referenciais teóricos que baseiam a proposta do estudo desenvolvido;
- **Capítulo 3 - Modelo de Desenvolvimento de Jogos Digitais:** Apresenta o Modelo que relaciona os principais conceitos envolvidos no processo de Desenvolvimento de Jogos Digitais e o protocolo para descrição dos Jogos Digitais;
- **Capítulo 4 - Sugestões para o Framework Scrum:** Apresenta as sugestões que objetivam potencializar a colaboração entre os membros da equipe de desenvolvimento;

- **Capítulo 5 - Considerações Finais:** Apresenta as conclusões, contribuições e algumas sugestões de atividades de pesquisa a serem desenvolvidas no futuro.

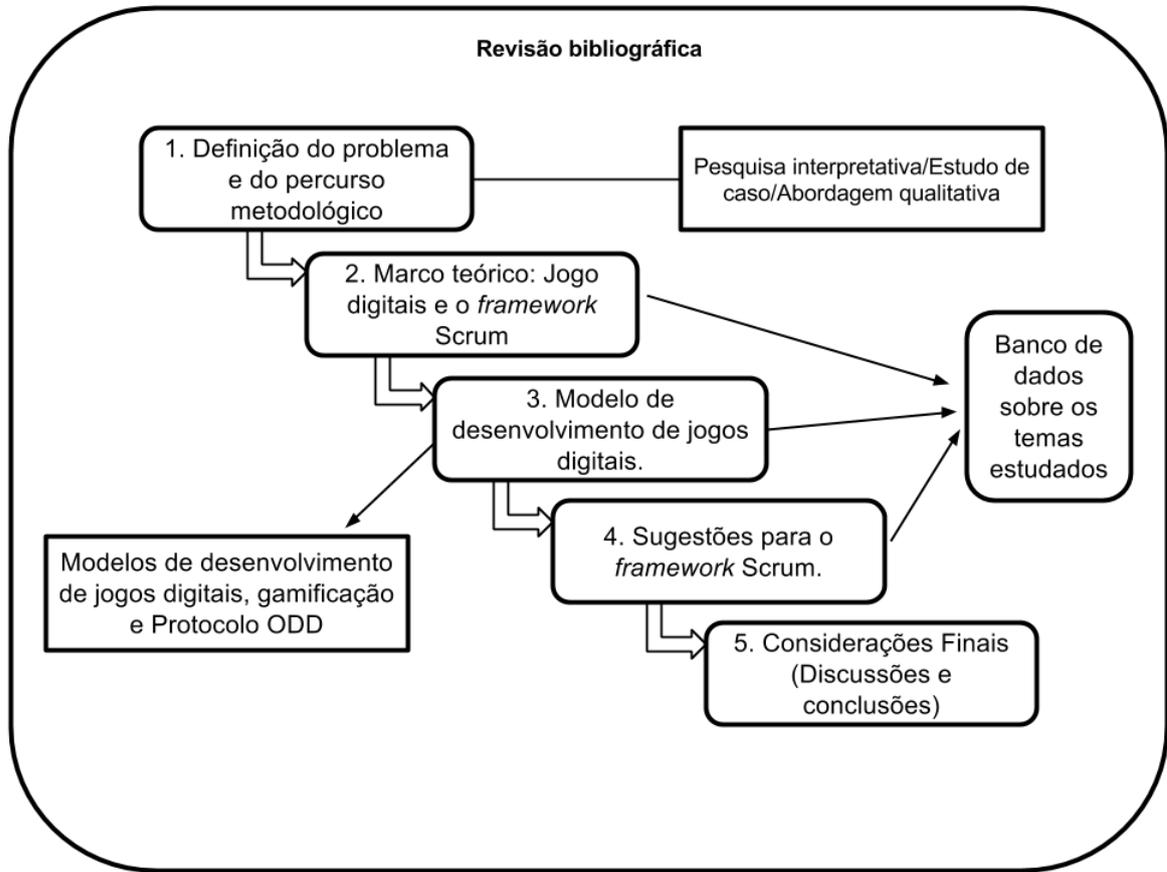


Figura 1.3: Fluxo da pesquisa. Fonte: Adaptado de (CARNEIRO, 2015)

## Os Jogos Digitais e as Metodologias de Desenvolvimento

---

Aqui serão apresentados os referenciais teóricos que irão basear a definição dos conceitos jogos, jogos digitais e *framework* Scrum, que serão as bases para a proposta do estudo desenvolvido.

### 2.1 O que é um jogo?

Nas últimas décadas, pesquisadores tentaram encontrar uma definição formal e fechada para os jogos ([ABT, 1987](#); [CRAWFORD, 1984](#); [COSTIKYAN, 1998](#); [PARLETT, 1999](#); [SALEN](#); [ZIMMERMAN, 2012](#); [SUITS, 2014](#)). Todas elas falharam por não conseguirem construir uma definição que contemplasse as diferentes perspectivas e natureza de todos os jogos. Porém, as diferentes linhas de pensamento utilizadas para conceituar um jogo permitem uma análise interessante dos aspectos importantes e necessários para compreender a essência dos jogos.

O estudo acadêmico e teórico dos jogos foi iniciado pelo filósofo e antropólogo Johan Huizinga na sua obra *Homo Ludens*. Neste trabalho, Huizinga afirma que os jogos fazem parte da formação cultural da humanidade e ressalta que eles precedem o homem. Para sustentar as suas afirmações, Huizinga alega que, durante a observação de animais (cães e lobos) brincando, poderíamos perceber regras (por exemplo, a mordida não pode ser forte) e ainda uma boa dose de satisfação. Outro aspecto muito importante que Huizinga traz em sua obra são as características de um jogo. Elas são: a voluntariedade (o jogador não é obrigado a jogar), o espaço (um jogo tem um espaço próprio para ser jogado), o estado mental (o jogador fica completamente focado no jogo), a não seriedade (os problemas do dia a dia são esquecidos durante o jogo) e a formação de grupos sociais (grupos e comunidades formadas para o jogo). Essas cinco características levam o jogador a um estado de concentração que Huizinga chamou de círculo mágico, que é teoricamente definido como o espaço físico e conceitual onde os jogos acontecem ([HUIZINGA, 1931](#)).

O autor Roger Caillois também trouxe grandes contribuições para o estudo dos jogos com a publicação *Os jogos e os Homens*. Em sua obra, Caillois afirma que o jogo é uma atividade livre, dinâmica, improdutiva, regida por regras e ficção. Todavia, o grande foco

do trabalho de Caillois está no estudo das atitudes psicológicas impulsionadas pelos jogos. Para o autor, os jogos possuem quatro fundamentos primários: *agon* (competição), *ilinx* (vertigem), *mimicry* (simulação) e *alea* (sorte). É justamente sobre esses princípios que os impulsos lúdicos dos jogos atuam no ser humano. A competição, a sensação de adrenalina, a fuga da realidade e a imprevisibilidade nos levam a um estado mental que nos proporciona sentimentos prazerosos. No entanto, Caillois também chama a atenção para os problemas que os jogos trazem. A deturpação do *agon*, *ilinx*, *mimicry* e do *alea* pode levar respectivamente à violência, ao desejo de poder, à superstição e à mudança de personalidade (CALLOIS, 1990).

Falando de um ponto de vista muito mais técnico e voltado para o *designer* e a jogabilidade (*gameplay*) de um jogo, Salen & Zimmerman (2012) afirmam que um jogo é um sistema no qual os jogadores se envolvem em um conflito artificial, definido por regras, que resulta em uma saída quantificável. Para os autores, os jogos possuem uma estrutura que possibilita o seu entendimento. Essa estrutura é dividida em três esquemas primários: as regras estão focadas nas estruturas norteadoras e limitantes intrínsecas dos jogos, na interação lúdica que enfatiza a interação do jogador com o jogo e com os outros jogadores, e na cultura que contextualiza e destaca os aspectos sociais em que o jogo está imerso.

McGonigal (2011) enxerga os jogos de um ponto de vista diferente. Para a autora, os jogos são caminhos que trilhamos para atingir um objetivo maior. Ela valoriza muito o trabalho colaborativo e afirma que os jogos possuem quatro características básicas: objetivo, regras, sistema de feedback e participação voluntária. O objetivo é o ponto focal que faz os jogadores agirem em função de um propósito maior. As regras são as limitações que os jogadores precisam obedecer para atingir os seus objetivos. O sistema de *feedback* possibilita que o jogador tenha a noção exata do seu atual estado dentro do jogo. Por fim, a participação voluntária implica na aceitação das regras e do sistema de funcionamento do jogo.

Porém, a grande contribuição de McGonigal (2011) está no entendimento que ela apresenta sobre a influência que os jogos têm nas vidas das pessoas. Entre 2001 e 2012, McGonigal desenvolveu uma série de jogos, todos desafiando os jogadores a resolver problemas reais, os chamados *Alternate Reality Games* (ARGs), Jogos de Realidade Alternativa. Escolhemos dois exemplos para ilustrar esse tipo de iniciativa.

Em 30 de Abril de 2007, McGonigal e um grupo de voluntários espalhados por 32 países iniciaram o jogo *World Without Oil*. Esse jogo tinha o objetivo de criar um ambiente

no qual os jogadores poderiam experimentar um mundo sem combustíveis derivados do petróleo. Todas as situações vividas deram origem a um banco de informações que relata possibilidades e ações para promover uma adaptação a uma possível mudança de fonte energética. O *World Without Oil* ganhou a menção honrosa do evento *Prix Green award for Environmental Art* em 2008 (MCGONIGAL, 2014).

O segundo exemplo de aplicação dos jogos na vida real escolhido foi o projeto *Superbetter*. Lançado em 2012, o *Superbetter* tinha o objetivo ajudar na reconstrução da autoestima de seus jogadores. Mais de 250.000 jogadores participaram desse jogo para transpor problemas como depressão, anorexia, insônia, dores crônicas no corpo e traumatismos cranianos. Os jogadores foram convidados a criar uma identidade secreta, um avatar, baseado em seu super-herói preferido. Esse avatar ajudava os jogadores a realizar tarefas (divididas em várias sub-tarefas) antes consideradas impossíveis (MCGONIGAL, 2014).

Kapp (2012), baseado nos conceitos de Salen & Zimmerman (2012), e utilizando a teoria da diversão de Koster (2013) como um guia para o desenvolvimento de jogos, define jogos como:

um sistema no qual jogadores se envolvem em um desafio abstrato, definido por regras, interatividade e *feedback*, que resulta em uma saída quantificável e frequentemente provoca uma reação emocional (KAPP, 2012).

Partindo de uma definição bem ampla encontrada em Camargo (2006), Magne Gasland define jogos como: um espaço de decisão com regras, recompensas e perdas. O espaço de decisão refere-se às diversas possibilidades que um jogador tem em diferentes situações do jogo. Uma escolha leva o jogador a outros espaços de decisão, e as sucessivas decisões tomadas por um jogador define o seu desempenho durante o jogo (GÅSLAND, 2011).

As regras são o conjunto de limitações a que o jogador deve obedecer dentro do espaço de decisão. Elas definem a dinâmica do jogo e estão diretamente relacionadas com a interação dos jogadores com o ambiente. As recompensas são os benefícios dados aos jogadores durante o jogo. Elas podem ser ilustradas com caminhos mais curtos para atingir o objetivo, itens extras, medalhas, pontuações e novas áreas de exploração. As perdas estão relacionadas a punições e também à falta de recompensas. Elas podem aparecer como caminhos errados que obrigam o jogador a refazer tarefas, um número maior de adversários ou penalização na pontuação. Porém o ponto mais importante relacionado às

recompensas e perdas é o *feedback* que o jogo dá ao jogador do seu atual estado no jogo (GÅSLAND, 2011).

## 2.2 Por que gostamos dos Jogos Digitais ?

De um modo geral, a presença dos jogos digitais em computadores, dispositivos móveis ou consoles cresceu muito nas últimas décadas (SEBRAE, 2014; FLEURY; NAKANO; CORDEIRO, 2014; NEWZOO, 2016). Esse aumento pode ser notado no meio comercial e em ambientes acadêmicos, através de iniciativas nacionais e internacionais, que utilizam os jogos como fonte de estudos e elementos norteadores de práticas pedagógicas (ALVES, 2012). O poder inovador dos JD cativa e motiva pessoas devido às diferentes formas de diversão que eles podem proporcionar (GÅSLAND, 2011).

No livro intitulado “A teoria da diversão para designer de jogos”, Koster (2013) defende que a diversão é um caminho para mudar o comportamento das pessoas para melhor. Segundo o autor, a diversão é causada por estímulos físicos e químicos no sistema nervoso, que ocorrem através de descargas de endorfinas no cérebro. Essas descargas proporcionam emoções quando executamos alguma atividade que nós gostamos, como escutar música, assistir a filmes ou ler livros. Com os jogos digitais não é diferente.

O conceito de diversão, que gira em torno dos jogos, encontra suporte na teoria do *Flow* (CSIKSZENTMIHALYI, 1997). A definição de *Flow* está pautada em um estado de concentração que uma pessoa se encontra quando está executando uma atividade, a ponto de mais nada ao seu redor ser importante, pois a própria atividade proporciona uma experiência de prazer e um estado de satisfação. Para Csikszentmihalyi (1997), o estado de *Flow* é:

a forma como as pessoas descrevem seu estado de espírito quando a consciência está harmoniosamente ordenada e elas querem seguir o que estão fazendo para seu próprio bem (CSIKSZENTMIHALYI, 1997).

Este estado de espírito descrito por Mihaly é alcançado quando encontramos o equilíbrio entre as potencialidades e as limitações das pessoas através de atividades que as levam a novos desafios. Conforme ilustra a Figura 2.1, ao iniciar um novo desafio o indivíduo se encontra em um estado de ansiedade (E1), porém em pouco tempo o indivíduo adquire

habilidade e essa situação muda para o estado de tédio (E2), porque o desafio não está mais à altura da habilidade do indivíduo. Quando um novo desafio é posto ao indivíduo, ele volta para um estado de ansiedade alto (E3) e vai adquirindo mais habilidades para conseguir superar o novo desafio e atingir um novo estado de satisfação (E4).

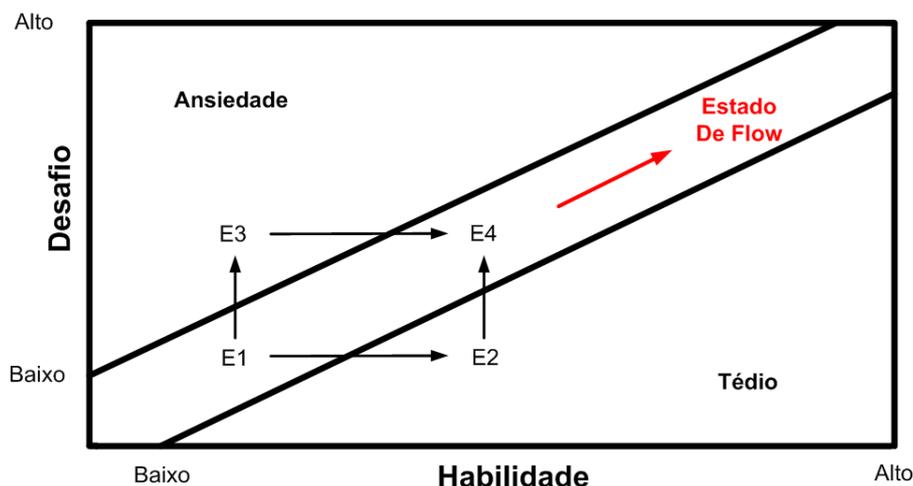


Figura 2.1: Principais sensações do indivíduo na realização de uma atividade até atingir o estado de *Flow*. Fonte: Adaptado de (CSIKSZENTMIHALYI, 1997).

Analisando a Figura 2.1, verifica-se que o indivíduo atinge o estado de *Flow* em dois momentos (E1) e (E4). Esse equilíbrio entre os desafios e as habilidades de uma pessoas é atingido quando, pelo menos, um dos oito elementos da teoria do *Flow* (desafios viáveis, concentração, objetivos claros, *feedback*, imersão, controle da situação, catarse, perda da sensação de tempo) é citado após uma atividade prazerosa (CSIKSZENTMIHALYI, 1997).

Devido a isso, podemos fazer uma associação direta entre a teoria do *Flow* e os JD. Todos os jogos que possuem uma grande aceitação do seu público-alvo têm a capacidade de manter o equilíbrio entre o nível de desafio e o nível de habilidade do jogador. Eles mantêm o jogador em um constante estado de *Flow*, ou seja, o jogo é a sua própria fonte de prazer, jogando sem expectativa de algum benefício futuro, mas simplesmente pelo desejo de participar da experiência (FARDO, 2013).

Diferentes autores estudam os JD a partir de pontos de vista particulares, mas existe um ponto de interseção onde as diferentes opiniões convergem para o mesmo lugar. Os JD são cativantes devido ao impacto direto que eles têm sobre os aspectos cognitivos, sociais e emocionais dos jogadores (MCGONIGAL, 2011; ALVES, 2012; GEE, 2003).

## 2.3 A mecânica dos jogos

Assim como na definição de jogos, é possível encontrar muitos pontos de ambiguidade na definição de Mecânica dos Jogos (MJ) (LUNDGREN; BJORK, 2003; KOSTER, 2013). Para iniciar a análise do conceito de MJ, vamos utilizar uma definição ampla, sem nos preocuparmos com explicações precisas de cada elemento. Para isso, utilizaremos o conceito apresentado por Gåsland (2011) que define MJ como:

o conjunto de elementos, estratégias ou padrões que compõem as regras de um jogo e contribuem para construção de ciclos de *feedback* (GÅSLAND, 2011).

Falando de um ponto de vista empresarial e com o foco voltado para os negócios, porém com uma grande base conceitual na mecânica dos jogos, Werbach & Hunter (2012) apresentaram um conjunto de exemplos e criaram um ambiente que direciona o leitor para a definição de MJ de forma intuitiva. Na sua obra, Werbach & Hunter (2012) utilizaram o termo PBL (*Points, Badges e Leaderboards*) para referenciar os elementos mais primitivos dos jogos (pontos, medalhas e tabelas de posicionamento) e os relaciona com a criação de ambientes (virtuais ou empresariais) que privilegiam a competência, a autonomia e o relacionamento. Essas aptidões possuem propriedades (ilustradas na Tabela 2.1) que possibilitam a construção de atividades motivantes. Elas concretizam a MJ.

Competência	Autonomia	Relacionamento
Resolução de problemas	Foco no jogador/usuário	Objetivos claros
Senso de progresso	Experimentação (sem medo de cometer erros)	Atividades com significado (a importância do conjunto de tarefas)
<i>feedback</i> Constantes	Personalização	Interação social (com os pares)

Tabela 2.1: As características da motivação intrínseca dos jogos. Fonte: (WERBACH; HUNTER, 2012)

Ou seja, para Werbach & Hunter (2012), a competência do jogador é evidenciada pela resolução de problemas que deixam claro para ele seu progresso dentro do ambiente de jogo, através de *feedbacks* constantes, que expressam o atual estado do jogador. A autonomia está baseada nas atividades que centralizam o jogador em uma constante experimentação em um ambiente que pode ser adaptado para as suas necessidades. E o relacionamento é evidenciado pelas relações sociais que são construídas em torno do propósito maior que o conjunto de atividades buscam alcançar.

Partindo da definição de Gåsland (2011) e utilizando a ambientalização de Werbach &

Hunter (2012), definimos Mecânica dos Jogos como: um conjunto de estruturas e estratégias que se relacionam para proporcionar experimentação, através das regras e ciclos constantes de *feedback*, ambientalização, pelas interações do jogador com seus pares e com o jogo e aperfeiçoamento por meio do progresso do jogador no ambiente de jogo.

A partir desse ponto, é possível notar que esse conceito de MJ não é exclusivo dos jogos. Podemos encontrar exemplos do uso de MJ em nossas vidas. Empresas já utilizavam as lógicas das recompensas e da pontuação para treinamento de seus funcionários; programas de televisão mantinham ou aumentavam o número de espectadores utilizando esses elementos; empresas de vendas de produtos e propaganda utilizavam MJ para aumentar a sua malha de vendedores com as famosas pirâmides. Os elementos da MJ nos dão o senso de conquista que nos diferencia perante outras pessoas (ALVES; MINHO; DINIZ, 2014), disponível no apêndice ??.

Uma outra forma de perceber a utilização de elementos da MJ é pensar no experimento de Burrhus Frederic Skinner conhecido como *Skinner Box*, a caixa de Skinner (SKINNER, 1953). Neste experimento, Burrhus Skinner modificou o comportamento de pombos e ratos utilizando recompensas e punições. Os animais utilizavam alguns dispositivos mecânicos como botões e alavancas para ganhar recompensas, comida, ou para evitar punições - pequenas descargas elétricas. Neste ponto, é possível fazer uma relação direta entre os jogos, especialmente os JD, e uma câmara de condicionamento. Ambos são estruturados e amparados por um conjunto de elementos de MJ para deixar claro ao jogador o seu atual estado dentro do ambiente de jogo.

Entretanto, destacamos que utilizar conjunto de elementos e estratégias para construir ciclos de *feedback* não é necessariamente algo ruim. A aplicação da MJ aprimora habilidades e aumenta a experiência dos sujeitos sobre um determinado assunto. Essa é uma das razões pelas quais pessoas passam de 4 a 8 horas por dia jogando (MCGONIGAL, 2011). Isso está intimamente associado ao desejo de alcançar o objetivo final do jogo e perceber o seu progresso nos níveis do jogo (ROUSE, 2010). Além disso, existe uma grande diferença entre os objetivos e conceitos apresentados no trabalho de Skinner (1953) e a definição de jogo de Gåslund (2011). O primeiro está claramente baseado no comportamentalismo, o segundo foca suas atenções na experimentação e nas livres escolhas do jogador.

Além dos argumentos acima, existem fontes acadêmicas que corroboram a favor dos JD. Bransford, Brown & Cocking (1999) e Schank (1999) apresentam, em seus trabalhos dados, resultados e argumentos para afirmar que o processo de aprendizagem acontece através de

experiências. Este atual entendimento é completamente oposto ao antigo, que afirmava que a mente humana funcionava como um grande computador, fazendo associações simbólicas utilizando regras para tomadas de decisão (BRANSFORD; BROWN; COCKING, 1999).

## 2.4 Metodologias de Desenvolvimento de Jogos Digitais

O ambiente, as condições e a organização dos anos pioneiros do desenvolvimento de JD foram completamente extintos. Tudo mudou: a exigência dos clientes, a natureza dos jogos, as plataformas computacionais, os profissionais envolvidos e as cifras empenhadas nesse processo. A figura solitária do desenvolvedor, que era responsável por toda concepção do JD (programação e artística), deu lugar a um exército de especialistas, que custa caro, e precisa desenvolver um produto que atenda às expectativas dos jogadores e gere lucro para os seus empregadores. Essas mudanças foram traumáticas, muitos erros foram cometidos devido a utilização de metodologias que não eram adequadas para o desenvolvimento de JD (KEITH, 2010). Nesta seção, iremos apresentar os principais eventos e metodologias que contribuíram para evolução do processo de construção de JD.

### 2.4.1 Os primeiros passos

No início, os jogos digitais não exigiam muito esforço artístico ou de programação durante o seu desenvolvimento. Eles eram projetados para serem executados em equipamentos específicos e tinham um ciclo de desenvolvimento muito curto. Nos anos 1970, e no início dos anos 1980, uma pequena fração do custo do lançamento de um JD estava no desenvolvimento. A maior parte dos recursos eram aplicados na construção do *hardware*, que geralmente eram caixas de metal ou de madeira, com um conjunto de componentes eletrônicos, que eram projetadas para executar um único jogo (COHEN, 1984; BAGNALL, 2005).

Como grande parte do investimento estava no *hardware*, nesta época é muito comum construir o equipamento e somente depois iniciar a construção do JD. Empresas como a Atari aplicavam um modelo de negócio no qual o JD era desenvolvido em um curto espaço de tempo, aproximadamente um mês. Se no final desse período o JD não fosse bom, ele era simplesmente jogado fora, e o desenvolvimento de um novo JD era iniciado. Essas práticas mudaram quando as definições da lei de Moore (MOORE, 2006) começaram a influenciar a produção de computadores domésticos que possuíam processadores de baixo

custo e grande poder de processamento. Essa nova plataforma computacional possibilitou o desenvolvimento de JD mais sofisticados e levou à produção de JD a um novo patamar.

Com o desenvolvimento e o barateamento dos computadores domésticos, novas possibilidades surgiram para indústria dos JD. Os Jogos Digitais passaram a explorar os novos recursos de multimídia dos computadores, e em pouco mais de 10 anos as equipes de desenvolvimento passaram de um grupo de quatro ou cinco programadores para times multidisciplinares de 40 a 50 desenvolvedores. Neste ponto, o custo de desenvolvimento dos JD já era muito maior do que o custo do *hardware*, e para diminuir os risco de falhas e desperdício de recursos, as empresas de desenvolvimento de JD passaram a adotar as metodologias de desenvolvimento de *software*, em particular a metodologia de desenvolvimento em cascata (KEITH, 2010).

A ideia geral da metodologia em cascata, ilustrada na Figura 2.2, é que o processo de desenvolvimento de *softwares* deve acontecer seguindo uma série de etapas sequenciais. Cada etapa dá origem a um produto, e à medida que o projeto vai avançando entre as fases, o seu custo vai aumentando, devido ao número de recursos e horas necessárias para execução do projeto. As fases iniciais da metodologia em cascata estão associadas a etapas de planejamento. O desenvolvimento de *software* ocorre nas etapas do meio, e as etapas finais são destinadas a integração e teste do *software* (ROYCE, 1970).

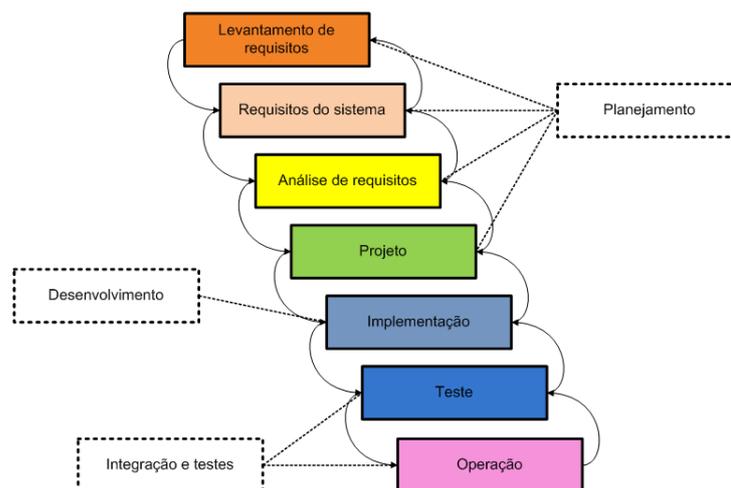


Figura 2.2: Metodologia de desenvolvimento em cascata. Fonte: Adaptado de Royce (1970)

Porém existe um problema que torna essa metodologia inapropriada para o desenvolvimento de JD. Embora exista a possibilidade de iteração entre as etapas da metodologia cascata, grande parte do planejamento é feita nas etapas iniciais do projeto. O oposto acontece com os testes, a maioria é efetuada nas etapas finais (KEITH, 2010; SCHELL,

2014). A distância que existe entre o planejamento e os testes, e a natureza linear das etapas de planejamento, implementação e teste, dificulta a busca de um estado de equilíbrio entre os aspectos artísticos e técnicos do JD, que Godoy & Barbosa (2010) costumam chamar de *fun of the game* e Keith (2010) refere-se como *find the fun*, no qual os desejos e anseios do público-alvo, os jogadores, são atendidos. Quando esse estado de equilíbrio não é encontrado, geralmente os JD não alcançam os indicadores (número de vendas, margem de lucro, aceitação do público, dentre outros) que a empresa esperava.

Além dos motivos anteriores, existem outras razões que impulsionam a busca de métodos mais eficientes para o desenvolvimento de JD. A necessidade de compor equipes multidisciplinares para construção de JD sofisticados aumentou bastante o custo de produção deles (KEITH, 2010; SALEN; ZIMMERMAN, 2012; SCHELL, 2014). Devido a isso, empresas de desenvolvimento buscaram soluções para contornar esses problemas que tiveram impacto direto na produção dos JD. A primeira solução encontrada pelas empresas de desenvolvimento de JD, foi lançar jogos com as mesmas características (objetivos, narrativas e mecânica de jogos semelhantes) objetivando fidelizar seus clientes. Porém essa medida teve repercussão direta na **falta de inovação** dos JD.

A segunda solução teve um impacto ainda maior nos jogos. Para reduzir os custos, as empresas de desenvolvimento começaram a diminuir o **conteúdo dos jogos** (fases menores, narrativas curtas, menos trabalho artístico), fato que teve repercussão direta na qualidade dos JD. Por fim, as empresas começaram a montar equipes pequenas para construção dos jogos, e contratar profissionais com menos experiência. Isso aumentou a quantidade de trabalho que cada desenvolvedor tinha que executar e afastou os desenvolvedores mais experientes (programadores e artistas) do mercado de desenvolvimento de JD, fato que ajudou a **deteriorar o ambiente de trabalho** (COHEN, 1984; BAGNALL, 2005; KEITH, 2010).

A junção desses três problemas (falta de inovação, redução do conteúdo e degradação do ambiente de trabalho) causados por tentativas frustradas de empresas que não conseguiram se adaptar a uma nova realidade do mercado de desenvolvimento de JD, aliada ao grande fracasso comercial da maior empresa de desenvolvimento de JD da época, a Atari, e a popularização dos computadores domésticos, deu origem a um evento que a literatura da área chama de A Crise dos Vídeo Games. (KEITH, 2010; RABIN, 2012).

Porém, esse ambiente adverso forçou a mudança abrupta que gerou bons frutos para a indústria dos JD. Novos nichos de mercados, criados principalmente pelas ações de mer-

cado e produtos lançados pela empresa Nintendo, surgiram e uma nova abordagem foi empregada para o desenvolvimento de JD (RABIN, 2012).

Embora seja possível encontrar diversas similaridades no processo de desenvolvimento de JD e de Sistemas Computacionais (SC), o ponto chave para encontrar a metodologia ou processo de desenvolvimento ideal para construção de JD, está justamente na identificação das especificidades que estão envolvidas na construção dos jogos (KEITH, 2010; GODOY; BARBOSA, 2010; SALEN; ZIMMERMAN, 2012; SCHELL, 2014).

As diferenças se apresentam com clareza quando identificamos o conceito e a natureza dos JD. A definição de Jogos Digitais apresentada em Gåsland (2011) diz que os JD são espaços de decisão com regras, recompensas e perda. Esta definição esta alinhada com a ideia de interação que Crawford (1984) apresenta ao comparar a dinamicidade dos JD com a linearidade das estórias.

Podemos fazer o mesmo paralelo entre os JD e os SC. Conforme ilustrado na figura 2.3, os Sistemas Computacionais apresentam uma série de eventos em sequência dentro de um espaço de tempo que sugere uma relação de causa e efeito entre as suas funcionalidades. Por outro lado, os JD possibilitam aos jogadores um conjunto de alternativas a cada tomada de decisão dentro do jogo. Nos SC os usuários experimentam uma sequência fixa de eventos, nos JD os jogadores são encorajados a explorá-los (KEITH, 2010; GODOY; BARBOSA, 2010; SALEN; ZIMMERMAN, 2012; SCHELL, 2014).

Quando unimos a definição de JD de Gåsland (2011), os conceitos de dinamicidade de interação e as perspectivas subjetiva e objetiva de Crawford (1984), o Círculo Mágico de Huizinga (1931) e a definição de mecânica dos jogos construída nesta pesquisa, poderemos observar a natureza dinâmica e exploratória dos Jogos Digitais. A principal ideia que a reunião desses conceitos apresenta é que os JD são artefatos cujo conteúdo está relacionado a comportamentos e experiências que atendem os desejos e anseios dos jogadores. Essas características não são observadas nos SC, e, devido a isso, classificamos os JD como artefatos que transcendem o escopo dos SC (KEITH, 2010).

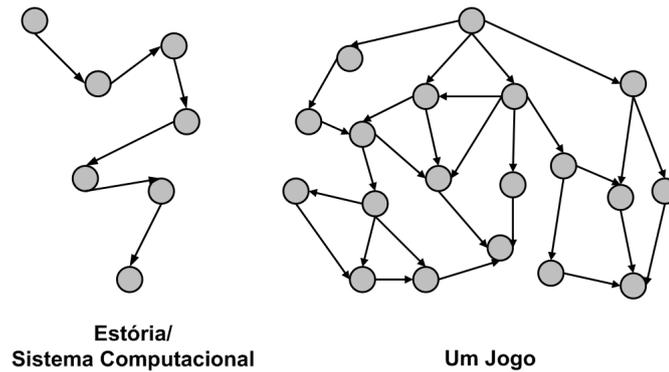


Figura 2.3: Interação entre usuários e os diferentes tipos de artefatos (Estórias/Sistemas Computacionais e Jogos Digitais). Fonte: Adaptado de [Crawford \(1984\)](#)

As características peculiares do processo de desenvolvimento de JD exigiam uma forma diferente de abordagem ([SCHELL, 2014](#)). Era preciso trabalhar em equipes e com foco voltado no talento e na criatividade para buscar inovação e aumentar o valor do produto desenvolvido todos os dias. Foi com essa motivação que a indústria de desenvolvimento de JD encontrou nas metodologias Ágeis, a flexibilidade necessária para construir jogos trabalhando com equipes multidisciplinares, livre de planejamentos e estimativas ineficientes, próximo do cliente final e com processos voltados para o aprimoramento constante do produto final ([KEITH, 2010](#); [SUTHERLAND, 2016](#)).

### 2.4.2 O framework Scrum

A primeira coisa que devemos entender antes de começar a abordar o tema Scrum, ilustrado na Figura 2.4, é que não iremos falar sobre uma metodologia. Scrum é um **framework**. O Scrum não é um conjunto de instruções, não é algo pronto que iremos utilizar para resolver algum problema ou construir um produto ([DUARTE, 2016](#)). Uma boa forma de idealizar o *framework* Scrum é pensar em uma estrutura básica que será utilizada para construir algo. A definição formal apresentada pelos seus criadores diz que o Scrum é

Um *framework* dentro do qual pessoas podem tratar e resolver problemas complexos e adaptativos, enquanto produtiva e criativamente entregam produtos com o mais alto valor possível ([SCHWABER; SUTHERLAND, 2013](#)).

Conforme ilustrado na Figura 2.4, a estrutura básica do Scrum é enxuta e simples. Ela possui um conjunto de eventos, artefatos e personagens que viabilizam o desenvolvimento do produto, contando com o **feedback**, quase que imediato, do usuário final, priorizando

as funcionalidades mais importantes, dando foco às pessoas e ao produto final. Essa é grande diferença do Scrum: o processo é um meio, não um fim. O objetivo maior é o produto final (KEITH, 2010).

Uma das características mais interessantes do Scrum está no seu empirismo. Esse *framework* permite que seja possível trabalhar com a incerteza e a criatividade através de uma estrutura que prioriza a aprendizagem, permitindo a análise do que já foi produzido e a forma que foi produzido. Devido a isso, o processo de desenvolvimento evolui com base nas suas experiências passadas. Esse processo de evolução empírica acontece devido à sustentação de três pilares: transparência, inspeção e adaptação (SUTHERLAND, 2016). Outro aspecto muito importante do *framework* Scrum é o conceito de *Timebox*. Todos os eventos do Scrum (*sprint planning*, *daily scrum*, *sprint review* e *sprint retrospective*) possuem um tempo definido de duração, isso evita reuniões longas, enfadonhas e improdutivas e mantém o foco dos participantes no objetivo principal da reunião (SUTHERLAND, 2016).

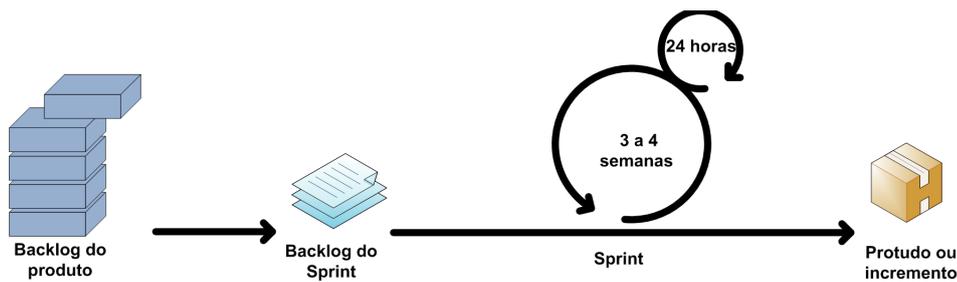


Figura 2.4: *Framework* de desenvolvimento Scrum. Fonte: Adaptado de Sutherland (2016)

Antes de apresentar todos os artefatos, eventos e o fluxo de trabalho do Scrum, vamos identificar os personagens que estão direta e indiretamente relacionados com o Scrum. Ilustrados na Figura 2.5, observamos no canto superior esquerdo, destacados em laranja, os *stakeholder*, pessoas que são importantes para o desenvolvimento de JD, mas estão fora do Time Scrum. Elas são responsáveis pela divulgação, vendas, distribuição, financiamento do projeto e relacionamento com os consumidores.

Em verde, no canto inferior direito, os consumidores finais dos jogos digitais. Eles são a válvula de escape do mercado. A linha de consumo deles tem influência direta na criação e no processo de construção dos JD. Alguns autores os coloca dentro da equipe de desenvolvimento (FULLERTON, 2014). No centro, destacados em vermelho, está o Time Scrum. Ele é composto pelo *Product Owner*, *Scrum Master* e Desenvolvedores. Eles são os responsáveis diretos pela construção dos JD. Vamos detalhar um pouco mais as responsabilidades de cada um deles.

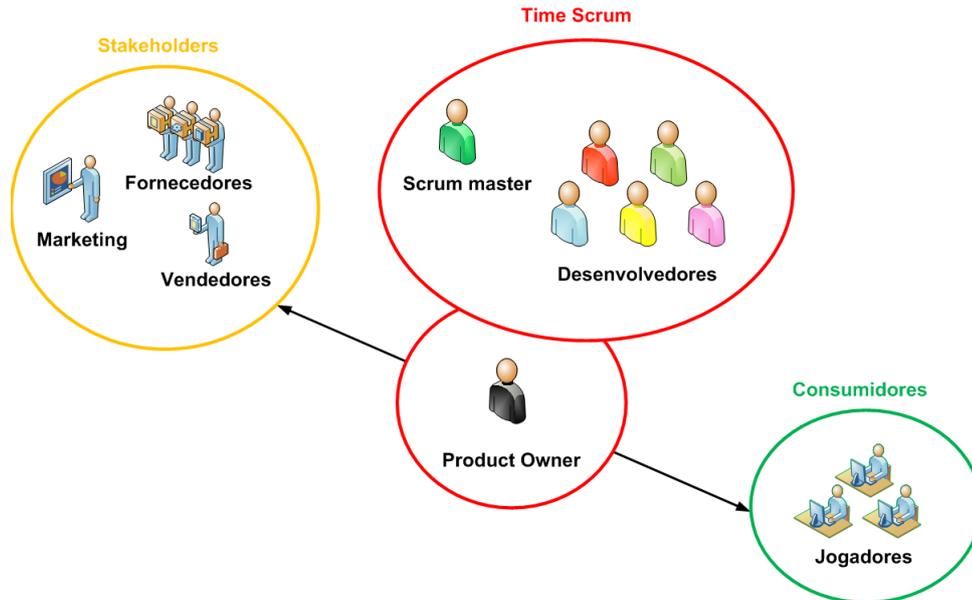


Figura 2.5: Personagens envolvidos no fluxo de trabalho do *framework* Scrum. Fonte: Adaptado de Sutherland (2016)

O *Product Owner*, embora não faça parte da equipe de Desenvolvedores, é uma das figuras centrais do Scrum. Ele é o responsável pelo contato entre o Time Scrum e as pessoas que estão fora da equipe (*stakeholders* e consumidores). Devido a isso, é ele quem possui o entendimento dos anseios e desejos do mercado e dos jogadores. Ele é responsável por passar para os Desenvolvedores todos os requisitos dos JD e precisa garantir o Retorno do Investimento (ROI) feito pelos *stakeholders*. As suas ações estão relacionadas à definição de prioridades, prazos e participação das reuniões de planejamento (*Sprint planning* e *Sprint review*) (KEITH, 2010).

O *Scrum Master* é o responsável por garantir que o Time de Desenvolvedores siga as práticas do Scrum. O trabalho dele está relacionado a três linhas de ações: eliminar obstáculos que impedem o trabalho dos Desenvolvedores; acompanhamento e alinhamento do atual estado do desenvolvimento através de ferramentas de gestão (artefatos) e encorajar a equipe a sempre buscar melhoria contínua. Ele participa de todos eventos do Scrum (*Sprint planning*, *Daily scrum*, *Sprint review* e *Sprint retrospective*) e ajuda o *Product Owner* na comunicação do Time com os *stakeholder* (KEITH, 2010; DUARTE, 2016).

Os Desenvolvedores são os responsáveis pela implementação do conjunto de itens que estão definidos no *backlog* do *Sprint*. Essa equipe é formada por um conjunto de profissionais que possuem as habilidades necessárias para executar as tarefas planejadas. Eles trabalham em equipe, gerenciam o próprio trabalho, definem quantas funcionalidades podem entregar até o final do *Sprint* e se comprometem a efetuar todo trabalho até a data

planejanda (SUTHERLAND, 2016).

O fluxo de trabalho do Scrum, ilustrado na Figura 2.6, é iniciado com a reunião de planejamento do *Sprint*, *Sprint planning*, que geralmente tem a duração de 8 horas. Somente os membros do Time Scrum participam dessa reunião. O *Product Owner* apresenta a lista de requisitos dos JD, *product backlog*, ordenada por prioridade. Os desenvolvedores, auxiliados pelo *scrum master*, definem quantas funcionalidades eles podem implementar dentro do *Sprint*, que geralmente tem a duração de 3 a 4 semanas. As funcionalidades escolhidas definem o *backlog* do *Sprint*, *Sprint backlog*, que é o conjunto de itens que serão implementados no *Sprint*. Além disso, existe outro artefato muito importante que é definido ou atualizado na reunião de planejamento do *sprint*: o documento de Definição de Pronto (*Definition of Done - DOD*). Ele contém todos os critérios que serão utilizados para definir se uma funcionalidade está pronta. Ele também é feito com base na experiência e necessidades dos Desenvolvedores.

Depois que esses dois artefatos (*Sprint backlog* e *DOD*) estiverem prontos, o *Sprint* é iniciado. Durante a execução do *Sprint*, diariamente, no início ou no fim do dia de trabalho, uma reunião chamada *Daily scrum* deve acontecer. Ela também é conhecida como Reuniões diárias, *Daily Meeting*, *Stand-Up Meeting* e *Daily Stand-Up*. Ela é bem curta, 15 minutos no máximo, e todos os participantes devem estar de pé. Isso garante que ela seja objetiva e rápida. Somente os Desenvolvedores e o *Scrum Master* participam dela. Nesta reunião, cada desenvolvedor deve responder três perguntas: O que eu fiz ontem? O que eu farei hoje? e Existe algum impedimento que atrapalhe o desenvolvimento do meu trabalho? A *Daily scrum* é um método eficaz de manter o alinhamento entre os membros do Time e permite que problemas e desvios sejam rapidamente descobertos. Ela deve ser realizada sempre no mesmo local e horário para entrar na rotina diária dos Desenvolvedores. Ela precisa ser respeitada e executada diariamente (DUARTE, 2016).

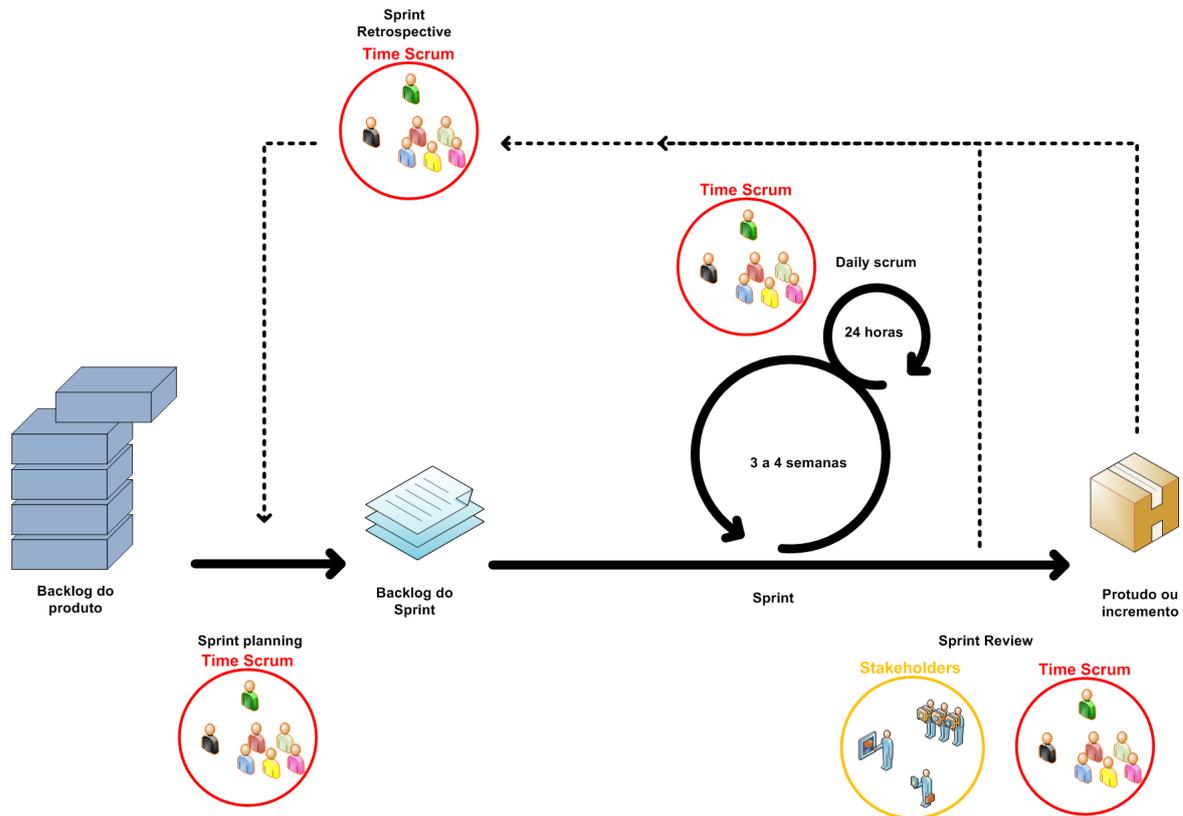


Figura 2.6: Eventos e artefatos do *framework* de desenvolvimento Scrum. Fonte: Adaptado de Sutherland (2016)

No final de cada *Sprint*, o produto ou incremento desenvolvido é apresentado ao *Product Owner* e aos *stakeholders* em uma reunião chamada *Sprint Review*. Essa reunião, de aproximadamente 4 horas de duração, é o evento de inspeção mais importante. É nesta reunião que o cliente final dá um *feedback* do produto que está sendo desenvolvido. Todas informações e críticas feitas pelo *Product Owner* e pelos *stakeholders* são utilizadas para refazer o *product backlog* e aperfeiçoar o JD em desenvolvimento.

O último evento do ciclo de desenvolvimento do Scrum é o *Sprint Retrospective*. Nesta reunião de aproximadamente 3 horas de duração, a equipe de Desenvolvimento e o *Scrum Master* falam sobre os resultados obtidos no *Sprint*, evidenciam e discutem as lições aprendidas e apresentam as ações que devem ser tomadas para melhorar o processo de desenvolvimento. Essa reunião contribui muito para a adaptação do Scrum à realidade do projeto. O grande desafio dessa reunião é apresentar ações exequíveis para evitar que os problemas reapareçam (SUTHERLAND, 2016).

Para sumarizar o nosso entendimento, sintetizamos os eventos, artefatos e os três pilares que sustentam o *framework* de desenvolvimento Scrum na tabela 2.2. Ela apresenta a re-

lação que existe entre os eventos Scrum, os artefatos utilizados em cada um dos eventos e a relação que eles têm com uma das sustentações ideológicas do Scrum. Vale ressaltar que, mesmo sem apresentar o pilar transparência na referida tabela, o vínculo que existe entre os componentes Scrum cria um ciclo que se retroalimenta. Ou seja, as ações de inspeção impulsionam as adaptações do processo de desenvolvimento que, por sua vez, evidencia a necessidade de transparência e o trabalho em grupo.

Evento	Inspeção	Adaptação
<i>Sprint planning</i>	- <i>Backlog</i> do produto - Acordos da retrospectiva - Definição de pronto	- <i>Sprint goal</i> - <i>Sprint backlog</i> - Novidades
<i>Daily Scrum</i>	- Progresso do <i>Sprint goal</i>	- <i>Sprint backlog</i> - <i>Daily scrum</i>
<i>Sprint Review</i>	- Incremento do produto - <i>Backlog</i> do produto - Análise do mercado	- <i>Backlog</i> do Produto
<i>Sprint Retrospective</i>	- Colaboração do time - Técnicas e tecnologias - Definição de pronto	- Adaptações no processo

Tabela 2.2: Objetivos dos eventos no Scrum. Adaptado de Schwaber & Sutherland (2013), Keith (2010)

## 2.5 Definindo os requisitos de um Jogo Digital

Definir os requisitos de um JD é uma atividade bastante semelhante à construção dos requisitos de um Sistema Computacional (FILHO *et al.*, 2013). Podemos definir requisitos utilizando o conjunto de documentos de análise de requisitos como Casos de Uso e Diagramas UML (caso de uso, classes, objetos, colaboração, sequência, atividades e outros). Em empresas que possuem processos de desenvolvimento com baixo grau de maturidade, essa atividade é feita até por *e-mail* (SOMMERVILLE, 2010). Neste trabalho, iremos focar a nossa atenção em duas técnicas bem aplicadas e utilizadas para definir requisitos de JD, as *user stories* e a prototipação (KEITH, 2010; FILHO *et al.*, 2013).

As *user stories* são pequenos documentos que possuem descrições curtas de desejos, necessidades ou características que um usuário espera encontrar em um sistema. Elas possuem uma estrutura básica, ilustrada da Figura 2.7, que apresenta o ator que a solicitou, a motivação, o tamanho e os benefícios que o conteúdo proporciona aos usuários (SCHETINGER *et al.*, 2011). Essa estrutura básica, simples e direta, permite que pessoas que não possuem experiência com o desenvolvimento de JD possam se comunicar com a equipe de desenvolvimento através das *user stories*. Esse fato é importante porque possibilita o contato direto com o usuário final, o consumidor dos JD, e com as pessoas que são

responsáveis pela construção de cenários que irão imergir o jogador na experiências do JD (KEITH, 2010).

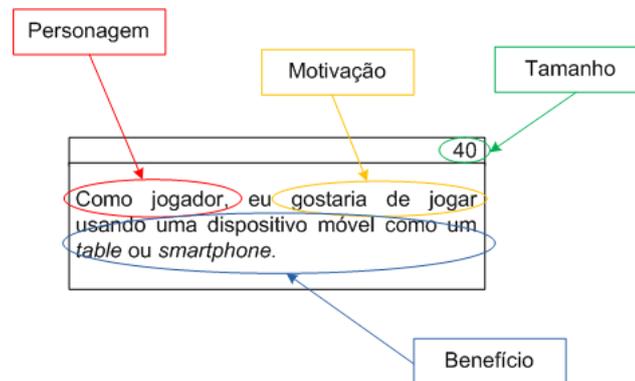


Figura 2.7: Estrutura de uma *user storie*. Fonte: Adaptado de Keith (2010)

A segunda técnica bem aplicada para definição de requisitos para construção de JD são os protótipos (FILHO *et al.*, 2013). Os protótipos são modelos que têm o objetivo de validar e concretizar as ideias que estão sendo desenvolvidas durante a produção. Eles podem ser generalizados em dois tipos: o conceitual e o físico, ilustrados na Figura 2.8. O protótipo conceitual geralmente é bem distante do produto final, ele objetiva estabelecer o funcionamento e comportamento do produto. Por outro lado, o protótipo físico, que no ambiente de desenvolvimento de JD é chamado de protótipo funcional ou *Build*, tem caráter mais voltado para validação e tende a se aproximar do produto final à medida que os ciclo de desenvolvimento, ou *Sprints*, são terminados (SHARP; ROGERS; PREECE, 2005; KEITH, 2010; SCHELL, 2014).



Figura 2.8: À esquerda, o protótipo conceitual, e à direita o protótipo funcionalDois tipos de protótipo segundo Sharp, Rogers & Preece (2005). À esquerda, o protótipo conceitual, e à direita o protótipo funcional. Fonte: Pinterest (2010)

A junção dessas duas técnicas possibilitam à equipe de desenvolvimento uma comunicação

simples, direta e eficiente com os seus clientes e com os jogadores, os usuários finais dos JD, além de proporcionar economia de recursos como dinheiro, dos *stakeholders*, e tempo dos desenvolvedores (SHARP; ROGERS; PREECE, 2005; KEITH, 2010; FILHO *et al.*, 2013; SCHELL, 2014).

Um exemplo da junção de técnicas, incluindo as *user stories* e a prototipação, para aprimorar a definição de requisitos, é o Google *Sprint*, ilustrada na Figura 2.9. Essa metodologia criada pela GV (Google Ventures), empresa da Google com o foco voltado para o empreendedorismo e teste de ideias inovadoras, possui cinco fases (Entendimento, Esboço, Definição, Prototipação e Validação) que devem ser executadas em cinco dias (Segunda-feira, Terça-feira, Quarta-feira, Quinta-feira e Sexta-feira) (KNAPP; ZERATSKY; KOWITZ, 2016).

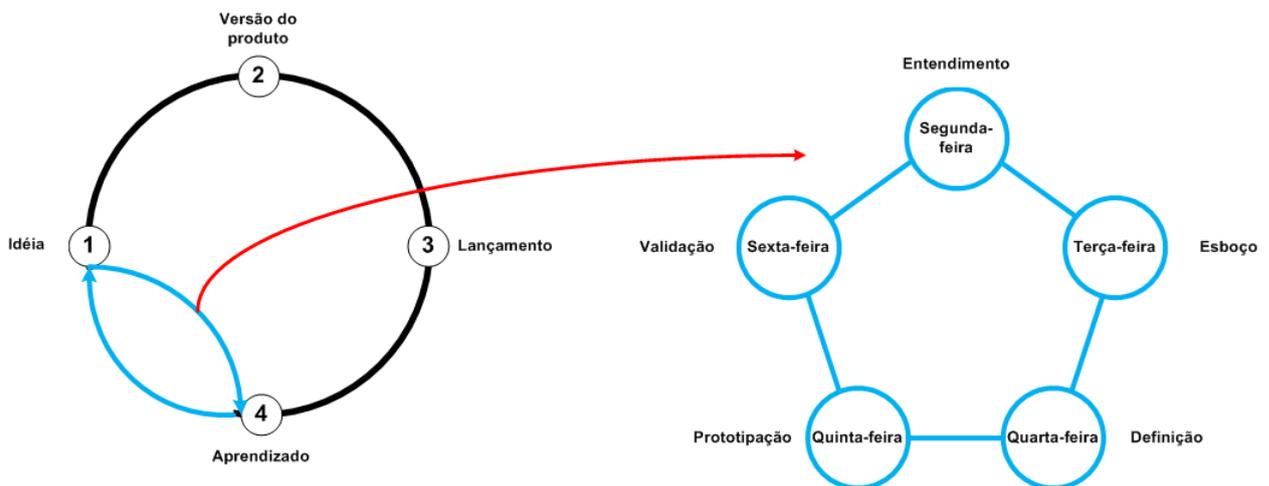


Figura 2.9: À esquerda visualizamos o processo geral das etapas de lançamento de um produto. À direita apresentamos as cinco etapas da metodologia Google *Sprint*. À esquerda visualizamos o processo geral das etapas de lançamento de um produto. À direita apresentamos as cinco etapas da metodologia Google *Sprint*. Fonte: Knapp, Zeratsky & Kowitz (2016)

Contudo devemos ressaltar que a metodologia Google *Sprint* não é bem aplicada para todo o processo de desenvolvimento de um JD. Afirmamos isso porque, da mesma forma que acontece na metodologia Cascata (ROYCE, 1970), todo planejamento da metodologia Google *Sprint* é efetivado no início do processo. Porém, conforme já dito anteriormente, a incerteza e o aprendizado são características intrínsecas ao processo de desenvolvimento de JD de médio ou grande porte. Devido a isso, JD de maior porte precisam de diversos ciclos de planejamento, desenvolvimento e aperfeiçoamento para atender às demandas e anseios dos jogadores (SCHELL, 2014).

## 2.6 Instrumentalização

Nesta seção, apresentaremos os modelos, protocolos, metodologias e práticas utilizadas para construção do modelo que define o processo de desenvolvimento de JD. Durante essa investigação, encontramos diversas publicações que poderiam ser aplicadas para definição do modelo que explica o funcionamento do processo de construção de JD e das sugestões de adaptação no *framework* Scrum que fizemos nesta obra. Dentre elas, podemos citar [Bartle \(2004\)](#), [LeBlanc \*et al.\* \(2004\)](#), [Schell \(2014\)](#), [McGonigal \(2011\)](#), [Salen & Zimmerman \(2012\)](#), [Chou \(2015\)](#), [Beck \(2009\)](#), [Poppendieck & Poppendieck \(2009\)](#).

Entretanto, optamos por publicações com características estruturantes para que o nosso modelo adquirisse as propriedades de um *framework*, que define o que devemos fazer para construir JD, não como devemos fazer. Devido a isso, escolhemos as seguintes obras: [Grimm \*et al.\* \(2006\)](#), [Grimm \*et al.\* \(2010\)](#), [Werbach & Hunter \(2012\)](#) e [Hunicke, LeBlanc & Zubek \(2004\)](#).

## 2.7 O protocolo ODD

O ponto de partida para construção do nosso modelo é o protocolo ODD (*Overview, Design concepts e Details*) ([GRIMM \*et al.\*, 2006](#)). Esse protocolo foi proposto para descrever Modelos Baseados em Indivíduos (IBM). Porém, antes de iniciar a apresentação do protocolo acima citado, precisamos entender dois conceitos que impulsionaram a necessidade do desenvolvimento do ODD: modelos e IBM.

Um modelo é uma representação de algum sistema real ou de uma abstração conceitual, e o principal motivo que nos impulsiona a construir modelos é a possibilidade de estudar, de forma efetiva e detalhada, como eles se comportam quando são submetidos a mudanças ([STARFIELD, 1990](#)). Um bom exemplo da aplicação de modelos é o estudo do desenvolvimento de cidades. Os parâmetros necessários para analisar a evolução de uma cidade são tão diversos e se modificam em uma escala tão pequena que seria praticamente impossível estudar esse evento de forma empírica e experimental. Porém, utilizando um modelo computacional, é possível emular o desenvolvimento da cidade, analisando as modificações dos parâmetros em diferentes escalas e, a partir desses dados, estudar desenvolvimento do modelo ([RAILSBACK; GRIMM, 2011](#)).

A definição clássica define IBM como modelos que descrevem entidades individuais e autô-

nomas (DEANGELIS; GROSS, 1992). Porém o grande problema dessa definição é que ela não diferencia os IBM dos outros modelos. Para solucionar esse problema, Uchmański & Grimm (1996) criaram quatro parâmetros que caracterizam os IBM, diferenciando-os dos outros modelos. Os quatro parâmetros são: (1) grau de complexidade do ciclo de vida do agente/indivíduo no modelo, (2) dinâmica de utilização dos recursos por parte do agente/indivíduo, (3) representação do número de agente/indivíduo na população e (4) variedade de agente/indivíduo da mesma idade na população.

O grau de complexidade do ciclo de vida representa as mudanças que um indivíduo sofre no decorrer de sua vida. Geralmente, as necessidades de consumo muda, a capacidade de reprodução diminui e a capacidade de se adaptar ao ambiente é reduzida. A dinâmica de utilização dos recursos caracteriza as necessidades da população de indivíduos e os diferentes tipos de recursos dispostos no ambiente. A representação do número de indivíduos ilustra o crescimento ou diminuição da população de acordo com as taxas de natalidade e mortalidade do sistema. E, por fim, a variedade de agentes da mesma idade representa a individualização do agente. Esse parâmetro possibilita a criação de indivíduos únicos na população, diferenciados pelas suas propriedades (UCHMAŃSKI; GRIMM, 1996).

Embora a utilização e aplicação de IBM fosse diversa e a natureza desse tipo particular de modelo fosse mais complexa, não existia um protocolo que padronizasse a descrição desse tipo de modelo. Devido a isso, a definição de modelos baseados em indivíduos era feita de forma verbal, detalhes matemáticos e a concepção estrutural dos modelos eram apresentadas através de longos textos, fato que dificultava o entendimento e a reprodução dos IBM. É essa lacuna que o ODD se propõe a preencher. Ele simplifica a leitura e o entendimento da descrição dos IBM, o que facilita o desenvolvimento, a atualização e proporciona uma maior credibilidade científica aos IBM. É justamente por causa dessa característica que nós escolhemos o ODD para ser uma das bases estruturantes do nosso modelo. Conforme ilustrado na Figura 2.10, o ODD é composto por três blocos principais que são subdivididos em sete (GRIMM *et al.*, 2006).

<b>Visão Geral</b>	<b>1 - Propósito</b>
	<b>2 - Variáveis de estado e escala</b>
	<b>3 - Visão do processo e escalonamento</b>
<b>Conceitos de Projeto</b>	<b>4 - Conceitos de Projeto</b>
<b>Detalhes</b>	<b>5 - Inicialização</b>
	<b>6 - Entrada</b>
	<b>7 - Submodelos</b>

Figura 2.10: Estrutura do protocolo ODD. Fonte: Adaptado de Grimm *et al.* (2006)

O bloco **Visão geral** é composto por três elementos (propósito; variáveis de estado e escala; e visão do processo e escalonamento) e tem como objetivo passar para o leitor a visão geral e a finalidade do modelo, dando a ele a noção de complexidade dos agentes que serão implementados. Propósito é o primeiro bloco de detalhamento do protocolo ODD. Ele contém o motivo pelo qual o modelo que será descrito precisa ser construído. Além disso, ele apresenta as informações básicas sem as quais o leitor não vai entender os princípios fundamentais do modelo.

As variáveis de estado e escala são o conjunto de variáveis mais importantes do modelos. Elas descrevem informações de configuração do ambiente e caracterizam os agentes que fazem parte do sistema. A apresentação dessas variáveis deve ser clara e direta, evitando a necessidade de deduções. O último elemento do bloco Visão Geral é a visão do processo e escalonamento. Ele descreve de forma direta e verbal, sem usar formalismos matemáticos, os conceitos que serão implementados em cada processo do modelo. Nesta etapa, é muito comum utilizar diagramas para facilitar a visualização do fluxo das informações e a execução dos processos.

O bloco **Conceitos de projeto** apresenta o esqueleto do modelo que será construído. Ele contém a descrição de todas as questões funcionais, dos aspectos comportamentais e dos esquemas de comunicação do modelo. Os métodos matemáticos que serão implementados no modelo também são apresentados nesta seção, mas o formalismo matemático não é exposto neste momento.

Por fim a bloco **Detalhes**. Ele também é composto por três elementos (inicialização, entrada e submodelos) e tem o objetivo de apresentar todos os detalhes que foram omitidos nas seções anteriores. Na seção inicialização, são definidos os valores iniciais do sistema e do ambiente. As variáveis descritas na seção variáveis de estado e escala recebem os seus valores iniciais.

O bloco entrada é um dos mais importante do protocolo. Os IBM estão imersos em ambiente computacional que contém outros agentes e uma série de variáveis e equações que simulam as condições desse ambiente. Devido a isso, os dados que entram e saem do ambiente e dos IBM são extremamente dinâmicos. Todas as relações de entrada e saída de dados do ambiente e dos IBM devem ser minuciosamente descritas nesta seção.

O bloco submodelos contém todos os detalhes e especificações matemáticas do modelo.

As explicações que outrora foram verbais nos blocos anteriores, aqui ganham detalhes. Geralmente essa seção é matematicamente densa. Os próprios autores [Grimm \*et al.\* \(2010\)](#) sugerem que sejam utilizadas equações e regras matemáticas dispostas em tabelas para facilitar o entendimento do modelo. E, se isso não for possível, o autor recomenda que essas explicações sejam feitas em outra publicação.

Quatro anos após a sua publicação, o protocolo ODD foi atualizado. A nova publicação do protocolo tinha o objetivo de aperfeiçoar e esclarecer alguns pontos do ODD que inviabilizaram a descrição de alguns modelos de IBM. Conforme ilustrado na Figura 2.11, as mudanças apresentadas na nova versão do protocolo ODD foram sutis, apenas 25% das publicações que utilizaram o ODD se equivocaram durante a descrição dos seus modelos, porém essas falhas foram essenciais para garantir o valor do ODD perante a comunidade científica ([GRIMM \*et al.\*, 2010](#)).

Além de modificar o nome de dois elementos (o bloco Variáveis de estado e escala passou a se chamar Entidades, variáveis de estado e escala e o bloco Entrada passou a se chamar Entrada de dados), o bloco Conceitos de projeto ganhou dois elementos (princípios básicos e aprendizagem) e sofreu a alteração de um dos seus elementos (o elemento Meta passou a se chamar Objetivo). Além disso, todos os blocos e elementos que compõem o protocolo ODD foram explicados detalhadamente para dirimir todas as possibilidades de equívocos.

	Elementos originais do ODD (GRIMM, 2006)	Elementos atualizados do ODD (GRIMM, 2010)
Visão Geral	1 - Propósito	1 - Propósito
	2 - Variáveis de estado e escala	2 - <b>Entidades</b> de estado e escala
	3 - Visão do processo e escalonamento	3 - Visão do processo e escalonamento
Conceitos de Projeto	4 - Conceitos de projeto <ul style="list-style-type: none"> <li>• Emergência</li> <li>• Adaptação</li> <li>• Meta</li> <li>• Predição</li> <li>• Sentido</li> <li>• Interação</li> <li>• Capacidade de armazenamento</li> <li>• Coletivo</li> <li>• Observação</li> </ul>	4 - Conceitos de projeto <ul style="list-style-type: none"> <li>• <b>Princípios básicos</b></li> <li>• Emergência</li> <li>• Adaptação</li> <li>• <b>Objetivos</b></li> <li>• <b>Aprendizagem</b></li> <li>• Predição</li> <li>• Sentido</li> <li>• Interação</li> <li>• Capacidade de armazenamento</li> <li>• Coletivo</li> <li>• Observação</li> </ul>
Detalhes	5 - Inicialização	5 - Inicialização
	6 - Entrada	6 - <b>Entrada de dados</b>
	7 - Submodelos	7 - Submodelos

Figura 2.11: Em amarelo destacam-se as diferenças entre as duas versões do protocolo Comparação entre as duas versões do protocolo ODD. Em amarelo destacam-se as diferenças entre as duas versões do protocolo. Fonte: Adaptado de Grimm *et al.* (2010)

## 2.8 A pirâmide de elementos da gamificação

O Segundo modelo que utilizamos como base estruturante do nosso modelo é a pirâmide de Werbach & Hunter (2012). Esse modelo foi construído para facilitar a introdução e o desenvolvimento de atividades e produtos, que utilizam a gamificação com o objetivo aumentar o engajamento dos recursos humanos da empresa e de seus clientes.

A formalização conceitual mais aceita sobre a gamificação afirma que gamificar consiste em utilizar a mecânica dos jogos em atividades que não estão dentro do contexto dos jogos (DETERDING *et al.*, 2011). Essa abordagem muda completamente a forma de construção das atividades, porque, neste caso, o foco das atividades são as pessoas, os aspectos motivacionais, a participação e o envolvimento dos sujeitos no processo. Desta forma, os elementos da mecânica dos jogos (desafio, objetivos, níveis, sistema de *feedback* e recompensa) são utilizados para criar situações que mobilizam e engajam os sujeitos para a realização de tarefas (MCGONIGAL, 2011). Para se ter uma ideia mais precisa do alcance da gamificação, apresentaremos um exemplo concreto da aplicação dessa metodologia.

Em 2010, pesquisadores da Universidade de Washington utilizaram o jogo Foldit para entender o funcionamento e o comportamento de uma determinada proteína que poderia

ser utilizada no combate ao vírus da AIDS. Esta iniciativa conseguiu contar com a cooperação de 46 mil participantes com experiência em diferentes áreas do conhecimento e muitos deles em áreas distantes da medicina. O esforço de trabalho desse grupo resolveu, em apenas 10 dias, o problema que os pesquisadores da área trabalhavam há mais de 15 anos sem sucesso (KHATIB *et al.*, 2011).

Para facilitar o entendimento da tênue fronteira que existe entre os jogos, as atividades gamificadas e os outros conceitos que estão intrinsecamente relacionados a eles, utilizaremos o diagrama ilustrado na Figura 2.12, de Deterding *et al.* (2011). Neste diagrama, estão representados, no eixo horizontal, os elementos dos jogos (em amarelo) e o jogo completo (em vermelho). No eixo vertical, representamos o formalismo de um jogo (em azul) e a liberdade de uma brincadeira (em branco). Os conceitos de jogos, gamificação, brincadeira e design lúdico são representados com as cores criadas a partir da junção de duas cores dos eixos horizontal e vertical.

De acordo com a Figura 2.12, a gamificação, representada pela cor verde, surge a partir da união do formalismo e dos os elementos dos jogos. É sobre o equilíbrio entre essas duas forças que as atividades gamificadas bem projetadas se sustentam. No quadrante ao lado, estão os jogos sérios, que são conceitualmente concebidos pela junção do formalismo dos jogos e de todos os elementos básicos que formam um jogo. Na parte inferior do diagrama, estão conceitos que estão essencialmente libertos de regras e estruturas rígidas - o formalismo não se aplica a eles. Os conceitos de brinquedo e design lúdico são livres e imbuídos de muita experimentação (DETERDING *et al.*, 2011).

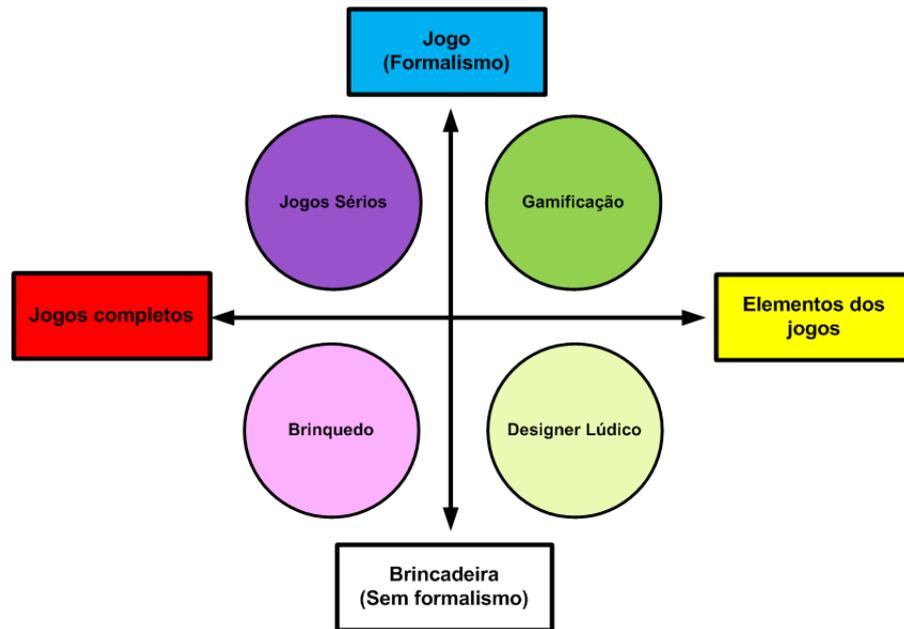


Figura 2.12: Relação entre jogos, gamificação, brinquedos e designer lúdico. Fonte: Adaptado de [Deterding et al. \(2011\)](#)

Conforme apresentado anteriormente, [Werbach & Hunter \(2012\)](#) usam o termo PBL (*Points, Badges e Leaderboards*) para se referir aos elementos da gamificação mais comuns (pontos, medalhas e tabelas de posicionamento) e utilizam-nos como ponto de partida para construção de uma estrutura hierárquica que alicerça a criação de estratégias gamificadas. A grande vantagem que essa estrutura oferece está no caminho que ela cria, através da associação direta, entre os elementos das diferentes categorias, fato que diminui a possibilidade de erros e mantém o foco da estratégia gamificada voltada para a solução do problema ou criação da atividade. A pirâmide de elementos de gamificação, ilustrada na figura 2.13, agrupa uma série de elementos organizados de forma decrescente em três categorias: dinâmica, mecânica e componentes.

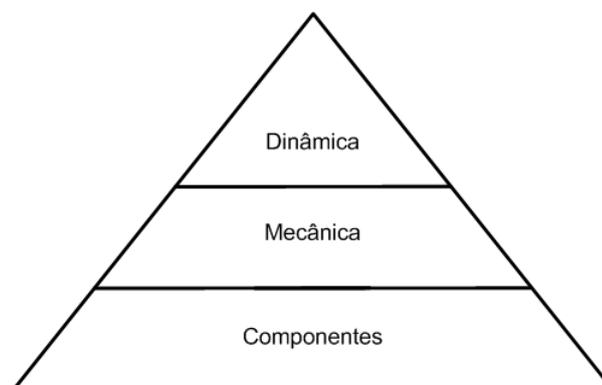


Figura 2.13: Pirâmide de elementos de gamificação. Fonte: Adaptado de [Werbach & Hunter \(2012\)](#).

A categoria dinâmica agrupa os elementos mais conceituais de um jogo. É nesta categoria que estão os suportes que, implicitamente, estruturam o jogo e as estratégias de gamificação. Aqui são definidos fatores limitantes do jogo como número de jogadores; tempo máximo de cada jogada e quantidade de vidas de um jogador; a narrativa que ambienta e emerge ludicamente os jogadores e, por fim, a forma de relacionamento entre os jogadores, que muitas vezes varia entre a competição e a cooperação.

A segunda categoria, a mecânica, está diretamente relacionada às ações que podem acontecer durante o jogo. Elas são as forças que guiam os jogadores dentro do ambiente de jogo. Aqui são agrupados os desafios, competições, ações cooperativas e todas as atividades que os jogadores vivenciam dentro do jogo. Neste ponto, vale a pena chamar a atenção para o conceito de regra. Na pirâmide de elementos de gamificação de [Werbach & Hunter \(2012\)](#), as regras do jogo não estão declaradamente dentro de uma categoria, elas estão implícitas nas definições das estratégias de gamificação e das ações de jogo.

A categoria componentes armazena os elementos que concretizam os conceitos definidos na dinâmica e mecânica do jogo. São elementos primitivos com os quais os jogadores irão interagir diretamente. São eles os pontos, medalhas, tabelas de posicionamento, níveis, avatares, dentre outros. Outro ponto muito importante que devemos ressaltar na pirâmide de elementos de gamificação de [Werbach & Hunter \(2012\)](#) é a relação que existe entre os elementos das três categorias. Os elementos da categoria dinâmica estão relacionados com os elementos da categoria mecânica. Da mesma forma, os elementos da categoria mecânica estão relacionados aos elementos dos componentes.

Para alcançar um melhor entendimento sobre a relação entre os elementos das categorias da pirâmide de elementos de gamificação de [Werbach & Hunter \(2012\)](#), apresentamos na Tabela 2.3 os elementos de PBL utilizando os JD da trilogia *Mass Effect*, relacionando-os com as categorias da pirâmide de elementos de gamificação.

Componentes	Mecânica	Dinâmica
Itens	São objetos que os jogadores ganham ou encontram durante o jogo como armas, dispositivos eletrônicos, códigos, senhas, armaduras, minerais, combustível.	Os jogadores utilizam os itens para obter alguma vantagem como, por exemplo, causar mais dano aos oponentes, acessar áreas especiais e atingir o objetivo com maior facilidade.
Pontos	O jogador ganha pontos quando realiza uma ação nobre (paladino) ou quando toma alguma decisão mais fria e calculista (renegado).	A moral do jogador (paladino ou renegado) possibilita diferentes espaços de decisão. Isso pode tornar o jogo mais difícil ou fácil.
Níveis	Um conjunto de missões é passado ao jogador (divisão do objetivo principal do jogo), dando ao jogador desafios progressivos e condizentes com as suas habilidades.	Em cada nível, o jogador encontra diferentes motivações para aumentar a capacidade de combate de sua tripulação.
Medalhas	São insígnias dadas ao jogador quando ele realiza um grupo de tarefas específicas durante o jogo	As medalhas destacam o jogador dentro dos ciclos sociais que ele frequenta (redes sociais, fóruns, sites, etc)

Tabela 2.3: Elemento de PBL da trilogia Mass Effect e suas relações com as categorias da pirâmide de elementos de gamificação de [Werbach & Hunter \(2012\)](#)

Contudo, devemos ressaltar que a pirâmide de elementos de gamificação de [Werbach & Hunter \(2012\)](#) não cobre a totalidade do assunto gamificação. Ela constrói uma associação direta entre os elementos dos jogos e possibilita a construção eficiente de atividades gamificadas, relacionando os princípios mais elementares (os componentes) com as regras que serão utilizadas nas atividades (a mecânica) e as estratégias que proporcionam o engajamento dos usuários (a dinâmica da atividade). Fora das fronteiras desse modelo está a experiência e estética de jogo provocada pela atividade gamificada.

## 2.9 O modelo MDA

O Terceiro e último modelo estruturante utilizado neste trabalho foi o MDA (*Mechanics, Dynamics and Aesthetics*) ([HUNICKE; LEBLANC; ZUBEK, 2004](#)). Esse modelo foi criado com o objetivo de diminuir as lacunas e fortalecer a interação entre os desenvolvedores de JD (artistas e programadores) e os usuários (jogadores, críticos e a comunidade acadêmica). Existem diversas metodologias para construção de JD, elas apresentam diferentes elementos que, independente da plataforma, são importantes para o desenvolvimento de um jogo, digital ou analógico ([MASTROCOLA, 2015](#)). Porém, a falta de experiência técnica dos usuários finais os fazem pensar que os diferentes tipos de jogos são construídos

da mesma forma. O MDA corrige essa distorção, e traz a possibilidade de avaliar os jogos independente do seu tipo ou plataforma (HUNICKE; LEBLANC; ZUBEK, 2004). Para deixar clara a importância do MDA, faremos uma reflexão sobre um modelo que define os elementos básicos que estão envolvidos no processo de desenvolvimento de JD.

Schell (2014), em sua obra *The Art of Game Design: A book of lenses*, apresenta os cinco elementos, ilustrados na Figura 2.14, que norteiam o processo de desenvolvimento de um JD. Ela apresenta, destacado em azul, os elementos do processo cujo foco estão voltados para o jogo. Em verde, destacam-se os elementos focados no jogador e, em vermelho, o elemento responsável pela concepção do JD e das estratégias gamificadas.

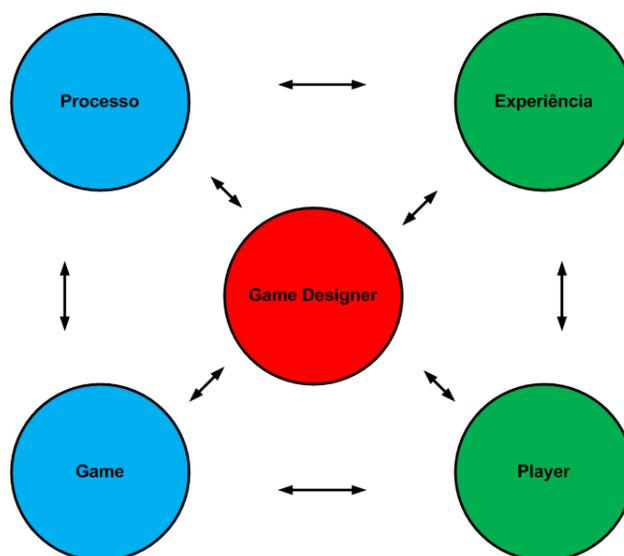


Figura 2.14: Em azul destacam-se os elementos do processo focados no jogo, em verde destacam-se os elementos focados no jogador e, em vermelho, o elemento responsável pela concepção e pelas estratégias gamificadas. Fonte: Adaptado de Schell (2014).

O *Game Designer* é um profissional que tem uma função central no processo de desenvolvimento de um jogo digital. Ele trabalha com um grupo de especialistas, ou sozinho em um projeto pequeno, que desenvolvem as estratégias que irão produzir a experiência de jogo que será vivenciada pelo jogador. Além de ser responsável pelo desenvolvimento e pelo sucesso do jogo, ele efetua funções administrativas lidando com prazos, documentação e clientes. Esse profissional, naturalmente, tem muitas habilidades; porém, a mais importante é saber ouvir a sua equipe e os jogadores que estão testando o jogo e o seu público-alvo. A compreensão dos anseios das pessoas que opinam sobre o jogo aproxima o *game designer* da experiência que ele deseja transmitir ao jogador.

O elemento **Processo** refere-se à metodologia utilizada para construção do *game*. Schell

(2014) utiliza o processo chamado *design* iterativo, ilustrado na Figura 2.15. Um método que realiza testes constantes no produto que está em processo de desenvolvimento, e prioriza a construção de protótipos para aprimorar as ideias e conceitos criados pelo *Game Designer*.

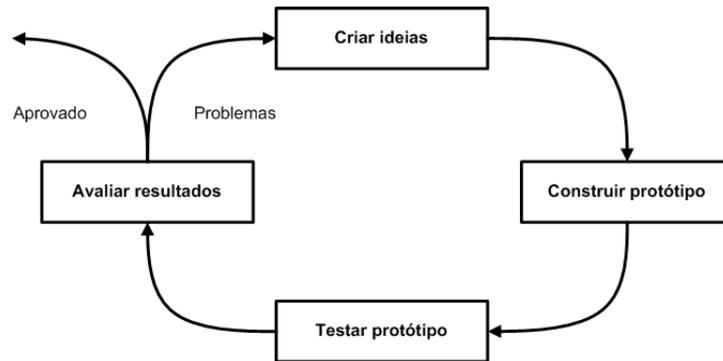


Figura 2.15: Fases do *design* iterativo. Fonte: Adaptado de Schell (2014).

A base conceitual dessa metodologia de desenvolvimento está no modelo espiral de desenvolvimento de software (BOEHM, 1988). O *design* iterativo possui três pontos básicos: análise de riscos, prototipação e *loop* (repetição). Em linhas gerais, durante o *loop* de iteração você executa as seguintes atividades (SCHELL, 2014):

- Apresentar as ideias básicas do design;
- Construir um protótipo para analisar os riscos;
- Testar o protótipo;
- Aprimorar o design com a experiência aprendida;
- Retornar ao passo 2.

O *Game* é o momento no qual a equipe de programadores trabalham na construção efetiva do jogo. Personagens, objetivos, procedimentos, fases, desafios, regras, recursos, artefatos e as limitações do jogo começam a ser materializados. Os protótipos são criados, testados, analisados e aperfeiçoados até alcançar o grau de maturidade e ludicidade idealizados pela equipe do projeto.

*Player* é uma etapa que está em execução constantemente. Nela são efetuadas pesquisas para precisar o perfil dos jogadores que o JD vai atender. Geralmente os *Game Designers* categorizam os jogadores por grupos para classificar os seus anseios. Schell (2014) agrupa

os jogadores por gênero (masculino e feminino), devido aos diferentes interesses que existem entre homens e mulheres, e por faixa de idade (noves grupos que são iniciados com crianças de 3 anos até adultos com mais de 50 anos) por causa das limitações de assuntos e temas que a idade impõe aos jogadores. Além disso, a faixa de idade tem um influência direta na mecânica e na jogabilidade do JD.

A **Experiência** do jogador é a razão principal do trabalho do *Game Designer*. Os JD nos proporcionam experiências memoráveis devido ao impacto direto que eles têm sobre os aspectos cognitivos, sociais e emocionais dos jogadores (MCGONIGAL, 2011; ALVES, 2012; GEE, 2003). Como existe um grande número de áreas das ciências que exploram os segredos da mente humana, Schell (2014) investiga três ramos do conhecimento para construir experiências para os jogadores: a psicologia para entender a mente humana, a antropologia para entender o comportamento humano e o *designer* para construir experiências prazerosas.

A união dos cinco elementos, apresentados por Schell (2014), e ilustrado na Figura 2.14, acontece de forma fluida no desenvolvimento de JD devido a duas características: metodologia e a interdisciplinaridade. As metodologias iterativas de desenvolvimento possibilitam que o produto idealizado pelo *Designer* seja avaliado em duas direções. O resultado final pode aperfeiçoar a implementação, e a implementação é capaz de refinar o produto final. Essa peculiaridade é muito importante, especialmente quando estamos trabalhando com JD. Esses artefatos propiciam manifestações de comportamentos complexos, que devem ser estudados antes de o JD ser construído, fato que exige uma equipe multidisciplinar e proximidade entre desenvolvedores e jogadores (HUNICKE; LEBLANC; ZUBEK, 2004).

Outra característica relevante está na forma como a equipe de desenvolvimento vai trabalhar. O foco é nunca perder de vista o *link* entre as diferentes áreas do conhecimento. A natureza multidisciplinar da equipe pode se perder durante o processo de desenvolvimento porque, muitas vezes, é necessário focar a atenção na sua área de conhecimento. Devido a isso, detalhes e aspectos importantes de outras áreas são ignorados. Essa fato é indesejado durante a construção e concepção de um JD devido ao propósito maior da equipe de multidisciplinar. O grupo precisa trabalhar de forma cooperativa, considerando questões fora da sua área de conhecimento (HUNICKE; LEBLANC; ZUBEK, 2004; SALEN; ZIMMERMAN, 2012).

É justamente neste ponto que a proposta apresentada pelo MDA se mostra importante (HUNICKE; LEBLANC; ZUBEK, 2004). Esse modelo possibilita a utilização coerente de

elementos dos JD que, aparentemente, são vistos como contraditórios, através do entendimento dos seus conceitos, analisados a partir da visão do *Designer* e do jogador. Essa necessidade surge da diferença que existe entre os JD e as outras mídias voltadas para o entretenimento como livros e vídeos. Os JD não são lineares, não é possível prever a sequência de eventos que irão acontecer durante uma partida. O jogador determina o seu caminho dentro do game (HUNICKE; LEBLANC; ZUBEK, 2004).

No MDA, os elementos dos jogos são agrupados em três componentes: **Regras, Sistemas e Diversão**. Esses componentes possuem contrapostos que os relacionam com o design: **Mecânica, Dinâmica e Estética**. Partindo do ponto de vista dos *Game Designers*, os JD são construídos a partir da **Mecânica** (conjunto de regras) que sustenta a **Dinâmica** do jogo (o sistema de jogo). A união desses dois elementos proporcionam a **Estética** ao jogador, ou seja, a experiência que o *Game Designer* deseja proporcionar. Por outro lado, a partir da perspectiva do jogador, a **Estética** é a resposta emocional que os JD provocam no jogador, tendo como base o seu comportamento perante a **Dinâmica** do jogo criada pelos componente, algoritmos e tecnologia utilizada na construção da **Mecânica** do jogo (HUNICKE; LEBLANC; ZUBEK, 2004).

A Figura 2.16, ilustra o objetivo final do MDA. Somente os componentes mecânica, dinâmica e estética são apresentados, mas são enxergados a partir do ponto de vista dos criadores e consumidores dos jogos. No acrônimo MDA apresentado na Figura 2.16, a letra M representa Mecânica (*Mechanics*), a letra D representa Dinâmica (*Dynamics*) e a letra A representa Estética (*Aesthetics*).

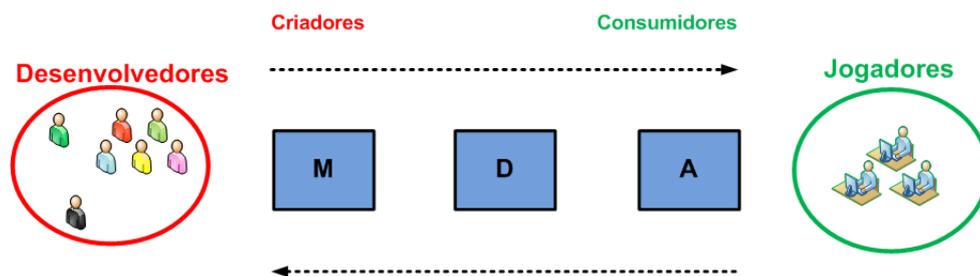


Figura 2.16: Principais elementos de design sugeridos pelo *framework* MDA (Mecânica, Dinâmica e Estética) representados pelas letras M, D, A respectivamente. Fonte: Adaptado de Hunicke, LeBlanc & Zubek (2004).

---

## Modelo de Desenvolvimento de Jogos Digitais

---

Aqui analisaremos o Modelo que relaciona os principais conceitos envolvidos do processo de Desenvolvimento de Jogos Digitais e o protocolo para descrição dos Jogos Digitais.

### 3.1 O Modelo de Desenvolvimento de Jogos Digitais

#### 3.1.1 O objetivo

O principal objetivo do Modelo de Desenvolvimento de Jogos Digitais (MDJD) é construir um *link* que relacione os desenvolvedores, os jogadores e os diversos conceitos que estão envolvidos no processo de desenvolvimento dos JD. Para isso, utilizamos a categorização de [Werbach & Hunter \(2012\)](#) e definimos as três categorias básicas para construção de atividades gamificadas. O conjunto de componentes do MDJD foram criados a partir as estrutura e organização de [Grimm et al. \(2006\)](#), a relação que existe entre as categorias do MDJD foram adaptadas de [Werbach & Hunter \(2012\)](#), e a visão do produto final do desenvolvimento a partir da perspectiva dos desenvolvedores e dos jogadores foi adaptado de [Hunicke, LeBlanc & Zubek \(2004\)](#).

O MDJD, ilustrado na Figura 3.1, nasceu do amadurecimento do modelo apresentado em [Diniz, Monteiro & Carneiro \(2016\)](#), que embora tenha o objetivo de orientar a concepção de Objetos de Aprendizagem Gamificados e o foco voltado para construção de componentes pedagógicos, nos ajudou a perceber que o principal objetivo do MDJD deveria ser estruturante. Deveríamos construir um modelo que se apresentasse como um arcabouço, um conjunto de lacunas com definições bem construídas para que os desenvolvedores pudessem definir o seu conteúdo. Dessa forma, o nosso modelo estaria definindo: o que deve ser feito e não como deve ser feito. Essa abordagem evidencia que não existe uma metodologia ótima para o desenvolvimento de JD ([SCHELL, 2014](#); [FULLERTON, 2014](#); [SALEN; ZIMMERMAN, 2012](#)): cada projeto deve utilizar uma metodologia que seja compatível com o tipo de jogo e as necessidade do grupo de pessoas envolvidas no processo de desenvolvimento.

### 3.1.2 As categorias e a relação entre os seus itens

Conforme descrito por [Werbach & Hunter \(2012\)](#), as três categorias básicas (Dinâmica, Mecânica e Componentes) agrupam, respectivamente, estratégias, regras e elementos que são responsáveis pelo Sistema de Jogo e pela resposta emocional dos jogadores. Elas são responsáveis pela construção de um Sistema Fechado e Formal, no qual um conjunto de regras irá orientar as ações dos jogadores dentro de um ambiente que representa, de forma subjetiva, um subconjunto do mundo real ([CRAWFORD, 1984](#)).

No MDJD, a categoria Dinâmica de Jogo é composta por duas subcategorias: a primeira, Propósito, é bem conceitual e tem o objetivo de definir as experiências que os jogadores irão vivenciar. Todo esforço e dedicação que o jogador investe no JD está diretamente relacionado ao objetivo principal, o propósito maior que ele tem que atingir ao final do jogo. As Estratégias de Gamificação são os mecanismos que o *game designer* vai utilizar para impelir o jogador de sempre buscar atingir o objetivo final do jogo.

Nos jogos da trilogia *Mass Effect*, o jogador precisa derrotar um exército de máquinas inteligentes chamadas Reapers, para evitar a extinção de todas as espécies inteligentes da nossa galáxia. Para isso, o jogador utiliza naves espaciais, veículos de guerra, faz alianças com povos de diferentes espécies, visita planetas de diferentes sistemas solares para obter recursos e solucionar problemas, ganha medalhas, pontos, tem acesso a novas partes do jogo que antes estavam inacessíveis, tudo isso sempre visando aumentar as suas chances de derrotar os Reapers. No exemplo citado acima, podemos notar a sutileza da relação entre os dois elementos da categoria. O vínculo é tão sutil que, fora do aspecto teórico, ou seja, durante a execução do jogo, é difícil perceber a fronteira entre esses dois elementos.

A segunda categoria do MDJD, Mecânica do Jogo, reúne o conjunto de regras e restrições que delimitam as ações dos jogadores dentro do ambiente de jogo. Esses itens são de extrema importância para concepção e desenvolvimento dos JD. Eles são responsáveis pelas relações entre os jogadores, estimulam a cooperação e trabalho em equipe e são responsáveis pelo aprimoramento dos jogadores. Essas características, e a influência que elas têm dentro dos JD e entre os jogadores, são observadas nos jogos da franquia *Destiny*. Nesses jogos, existem uma série de equipamentos (armas, armaduras e veículos) e fases que o jogador só tem acesso depois que ele atingir um certo grau de habilidade ou efetuar um conjunto de tarefa. Além disso, existem fases que, necessariamente, só podem ser concluídas quando dois ou mais jogadores trabalham juntos para derrotar um inimigo ou resolver um problema que precisa de habilidades que não são encontradas em somente um personagem.

A terceira e última categoria do MDJD, Componentes dos Jogos, agrupa os itens básicos da mecânica dos JD. Eles são como uma interface direta com o jogador. Ela possui três subcategorias que classifica os seus itens a partir da natureza de cada um deles. A subcategoria Fatores Gráficos é composta por itens que os jogadores têm contato direto durante o jogo. São os modelos, 2D ou 3D, que compõem o conjunto de objetos artísticos do Jogo como itens dos jogadores, veículos, paisagens, filmes, NPCs, medalhas, tabelas de pontuação, entre outros.

A Narrativa é a história que ambientaliza o jogador dentro do jogo. Ela constrói uma linha guia dentro do ambiente de jogo. Ela está diretamente relacionada com a categoria Dinâmica do Jogo. Por fim, temos a subcategoria Fatores técnicos. Nesta categoria, está o grupo de técnicas e tecnologias utilizado para desenvolver os JD e possibilitar a interação dos jogadores com o ambiente e com outros jogadores. Para melhor entendimento sobre essa subcategoria iremos citar o JD *Battlefield 4*. Neste jogo de tiro em primeira pessoa, os jogadores estão em um campo de batalha duelando entre times. Uma das funcionalidades desse JD possibilita utilizar um dispositivo móvel, *smartphone* ou *tablet*, para acessar um módulo do jogo chamado “Modo Commander”. Com essa funcionalidade, é possível entrar no jogo como um técnico e auxiliar a movimentação do seu time dentro do campo de batalha.

Todas as categorias, e subcategorias, acima descritas representam e agrupam conceitos importantes para o desenvolvimento de JD. Porém, isoladamente, elas trazem pouco sentido prático para desenvolvedores com pouca experiência na construção de JD. Para resolver esse problema, decidimos aplicar o mesmo princípio utilizado por [Werbach & Hunter \(2012\)](#) e criamos relacionamentos entre os itens das diferentes categorias/subcategorias. Para isso, os itens funcionais da categoria Componentes dos Jogos devem estar relacionados a pelo menos um dos itens da categoria Mecânica dos Jogos. Isso garante que todos os elementos básicos com os quais o jogador precisa interagir durante o jogo tenha funcionalidade. Essa regra não se aplica a todos Componentes dos Jogos, apenas aos componentes funcionais.

Da mesma forma, todos os itens da categoria Mecânica dos Jogos devem estar relacionados com pelo menos um dos itens da categoria Dinâmica dos Jogos. Neste caso, não existe exceção. Todos os elementos relacionados às questões Mecânicas do Jogo devem estar relacionados com o propósito ou com uma das estratégias de gamificação do JD.

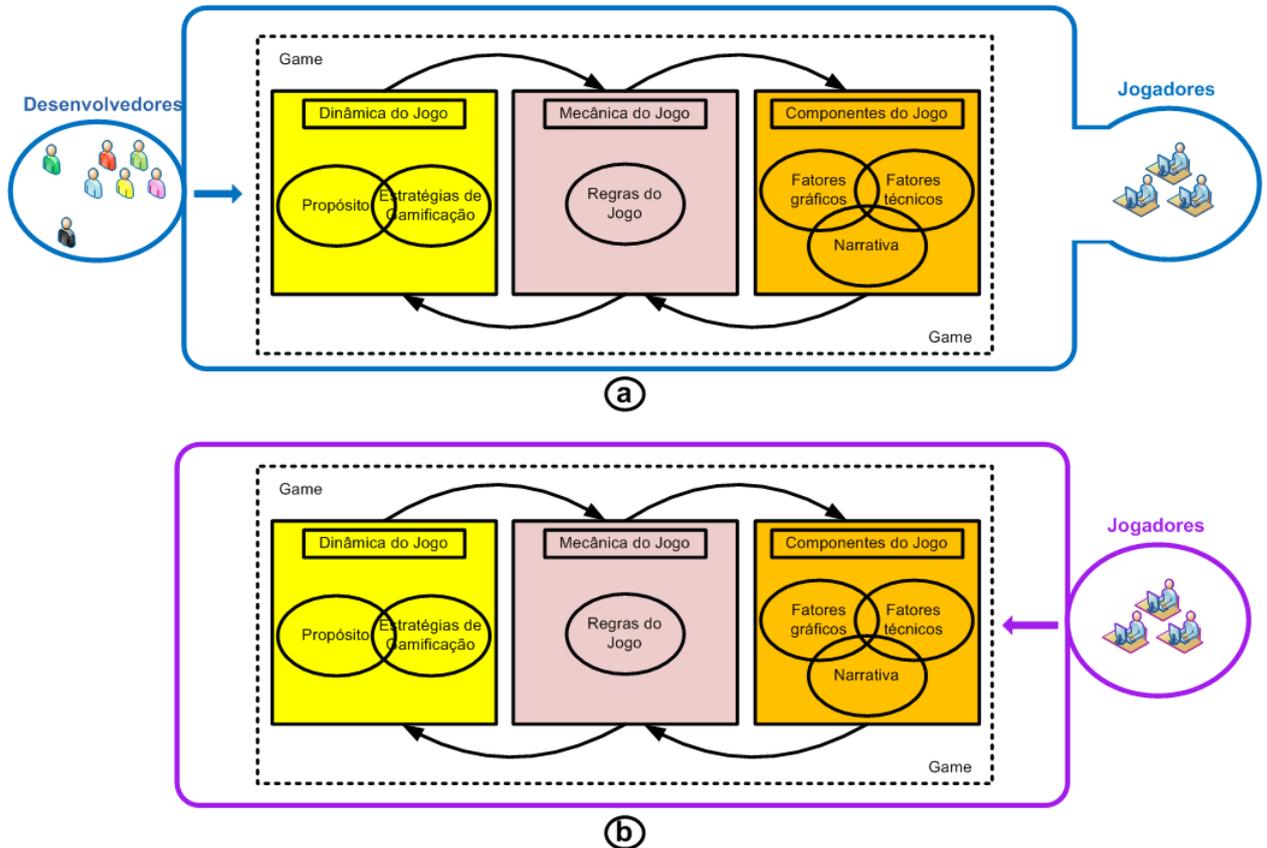


Figura 3.1: Na imagem "A", observamos a perspectiva dos Desenvolvedores. Em "B", observamos a perspectiva dos Jogadores. Fonte: elaborado pelo autor.

### 3.1.3 As perspectivas dos desenvolvedores e jogadores

Um ponto muito importante e que precisa ser considerado durante o processo de desenvolvimento de JD são as diferentes percepções dos agente envolvidos na construção dos JD. Hunicke, LeBlanc & Zubek (2004) apresentam uma sugestão que convida estudiosos e pesquisadores da área a decompor os JD, e com isso entendê-los, a partir da perspectiva dos desenvolvedores e dos jogadores. Fullerton (2014) apresenta uma abordagem que ela chama de “*a Playcentric approach*”, na qual os jogadores são colocados dentro do processo de desenvolvimento dos JD como uma figura ativa com a função de validar as experiências que os JD proporcionam a eles.

Seguindo a mesma linha de raciocínio de Hunicke, LeBlanc & Zubek (2004) e Fullerton (2014), apresentamos no MDJD duas figuras, ambas com a atenção voltada para o jogo, mas com preocupações e anseios diferentes. Os desenvolvedores têm como principal objetivo construir o conjunto de experiências que os jogadores irão vivenciar, partindo da Dinâmica do Jogo, seguindo as regras e restrições da Mecânica, e utilizando os Compo-

nentes dos Jogos. Fullerton (2014) apresenta três exemplos que ilustram o ponto central da preocupação da equipe de desenvolvimento. Eles são:

... os jogadores terão de cooperar para vencer, mas o jogo será estruturado de forma que eles nunca possam confiar uns nos outros [...] os jogadores vão sentir uma sensação de felicidade e descontração ao invés de competitividade [...] os jogadores terão a liberdade de escolher quais objetivos eles querem perseguir ... (FULLERTON, 2014).

Já os jogadores, os consumidores dos JD, enxergam os jogos a partir de outras lentes. Eles têm contato com o jogo através dos Componentes, e a partir deles vivenciam as experiências do ambiente projetadas na Dinâmica dos Jogos e regidas pela Mecânica. Podemos encontrar um bom exemplo da interação dos jogadores com o ambiente de jogo em McGonigal (2011), quando ela descreve o conceito de produtividade prazerosa:

... produtividade prazerosa é a sensação de estar profundamente imerso no trabalho que produz resultados imediatos e óbvios. Quanto mais claros os resultados, e quanto mais rápido os alcançarmos, mais felizes e produtivos nos sentiremos. E nenhum jogo nos dá uma sensação melhor de conseguir visualizar o trabalho feito do que *World of Warcraft* [...] sua missão principal em *World of Warcraft* é auto-aperfeiçoamento - um tipo de trabalho que quase todos nós achamos naturalmente atraente. Você tem um avatar e seu objetivo é fazer com que ele seja melhor, mais forte e mais rico de todas as maneiras possíveis: mais experiência, mais habilidades, armadura mais forte, mais habilidades, mais talento e uma maior reputação ... (MCGONIGAL, 2011).

### 3.1.4 O protocolo para descrição dos Jogos Digitais

Motivados pelo desejo de apresentar propostas que interfiram positivamente no processo de desenvolvimento de JD, sugerimos um método criado a partir de uma das possíveis leituras e interpretações do MDJD. Trata-se de um protocolo, ilustrado na Figura 3.2, que possibilita a construção de um documento de *designer* que descreve todas as ideias, regras, componentes e modelos matemáticos que precisam ser implementados durante o

desenvolvimento de um JD. Ele foi idealizado a partir das duas versões do protocolo ODD (GRIMM *et al.*, 2006; GRIMM *et al.*, 2010).

Ele consiste de oito elementos que estão agrupados em três blocos (Dinâmica, Mecânica e Componentes) e seguem as mesmas especificações das categorias (Dinâmica de Jogo, Mecânica do Jogo e Componentes do Jogos) apresentadas na Seção 3.1.1. Porém o bloco Mecânica possui três elementos (Interatividade, Regras e Modelos Matemáticos), para retratar todos os aspectos envolvidos na interação entre o jogador e o ambiente de jogo. A ideia básica deste protocolo é que qualquer pessoa possa, rapidamente, ter noção do trabalho que deve ser feito, lendo as especificações de cada elemento do protocolo e seguindo o fluxo de conexões sugerido pelo MDJD. Para alcançar um melhor entendimento a respeito das seções do protocolo, apresentamos cada uma delas na Tabela 3.1 e disponibilizamos no Apêndice A, um exemplo de aplicação do Protocolo para descrição de JD caracterizando um dos jogos mais populares do mundo (TETRIS, 1996).

<b>Dinâmica</b>	<b>1 - Propósito</b>
	<b>2 - Estratégias Gamificadas</b>
<b>Mecânica</b>	<b>3 - Interatividade</b>
	<b>4 - Regras</b>
	<b>5 - Modelos matemáticos</b>
<b>Componentes</b>	<b>6 - Narrativa</b>
	<b>7 - Fatores gráficos</b>
	<b>8 - Fatores técnicos</b>

Figura 3.2: Protocolo para descrição de Jogos Digitais. Fonte: elaborado pelo autor.

A utilização deste protocolo, em conjunto com duas práticas colaborativas que iremos apresentar no próximo capítulo, viabiliza a redução de problemas de comunicação e planejamento. A ideia central é que esse protocolo seja utilizado como uma ferramenta pelo *Product Owner*, e sirva como um guia para definir a prioridade dos requisitos que irão compor o *Product Backlog* do JD.

Outro ponto positivo que esse protocolo proporciona, apresenta-se quando deixamos de enxergar os JD como artefatos com foco voltado para o entretenimento e passamos a observá-los como objetos de estudos científicos. Os *serious games* (ABT, 1987; MI-

CHAEL; CHEN, 2005), jogos cujo propósito não está exclusivamente voltado para o entretenimento, são utilizados com finalidades terapêuticas, educacionais e socioculturais desde o fim da década de 1980 (WILKINSON, 2016). Partindo desse ponto de vista, um dos grandes problemas enfrentados pelos estudiosos da área está na análise e descrição dos JD.

Mesmo contando com a contribuição de trabalhos com os de Hense & Mandl (2012) e Petry *et al.* (2013) a análise dos JD ainda é realizada de forma verbal, fato que dificulta a reprodução das experiências construídas no JD. Devido a isso, a utilização de um protocolo padrão para descrição de JD contribuirá para aumentar a credibilidade científica dos JD, devido à possibilidade de reprodução das experimentações vivenciadas pelos jogadores.

Bloco	Seção	Definição
Dinâmica	1- Propósito	Essa seção apresenta o objetivo principal do JD e as tarefas que serão realizadas para alcançar o objetivo principal
	2- Estratégias Gamificadas	Aqui apresentamos as estratégias e mecanismos utilizados para manter o jogador vinculado à história que ambienta ao JD.
Mecânica	3- Interatividade	Apresenta de forma sucinta as tecnologias utilizadas para possibilitar a interação dos jogadores com o ambiente de jogos e com outros jogadores.
	4- Regras	Esta seção armazena todas as regras e restrições que serão implementadas no JD.
	5- Modelos matemáticos	Esta seção apresenta a descrição detalhada de todos os modelos e relações matemática que serão implementadas nos JD.
Componentes	6- Narrativa	Aqui apresentamos a história que orienta o jogador do início ao fim dos JD. Ela também apresenta as histórias pontuais que o jogador vivencia para executar tarefas dentro do jogo.
	7- Fatores gráficos	Esta seção apresenta a descrição detalhada de todos os objetos, modelos 2D ou 3D, que possuem funcionalidade dentro dos JD.
	8- Fatores técnicos	Aqui apresentamos todas as técnicas e tecnologias, com riqueza de detalhes, utilizadas para desenvolver os JD.

Tabela 3.1: Blocos e seções do protocolo para descrição de Jogos Digitais. Fonte: elaborado pelo autor

---

## Sugestões para o Framework Scrum

---

Aqui será apresentado o conjunto de sugestões que acreditamos que potencializará a colaboração entre os membros da equipe de Desenvolvedores. As sugestões estão diretamente relacionadas com a reunião de planejamento do sprint, *Sprint planning*, e a verificação dos itens produzidos durante o *Sprint*.

### 4.1 Apresentação

As sugestões que estão descritas neste capítulo foram construídas com base em experiências empíricas vivenciadas durante o desenvolvimento JD SimGE e LIPISpace, e na pesquisa bibliográfica efetivada neste projeto. Inicialmente, julgamos que os pontos que deveriam ser aprimorados se originavam de situações específicas, mas encontramos publicações com observações e preocupações semelhantes à nossa, fato que nos impulsionou a continuar realizando esta pesquisa.

A maior parte dos profissionais que estavam envolvidos nos projetos de desenvolvimento dos JD que foram analisados, não tinham experiência com o Scrum (SCHWABER; SUTHERLAND, 2013). As equipes eram formadas por pedagogos, professores e programadores. Todos eles tinham estudado ou utilizado, em algum momento de vida acadêmica ou profissional, alguma metodologia baseada na metodologia cascata (ROYCE, 1970). Para eles, dividir o projeto em etapas, e executá-las uma após a outra era, natural e indispensável. Devido a isso, não foi possível aplicar todos os artefatos e eventos do Scrum. Somente alguns foram aplicados com adaptações e restrições.

Esse ambiente de trabalho restringia o fluxo de atividades ideal para o desenvolvimento de JD e gerou dois grandes problemas: os Desenvolvedores não eram ouvidos durante a construção ou reconstrução do planejamento, e os trabalhos (de programação e artístico) eram realizados separadamente. A interação entre os Desenvolvedores durante a produção e verificação dos artefatos construídos era pequena.

Esses mesmos problemas foram encontrados em publicações acadêmicas. Preocupados com a forma que os JD são desenvolvidos, Hunicke, LeBlanc & Zubek (2004), Godoy

& Barbosa (2010), Gregory (2010) falam da importância de se trabalhar com processos ou metodologia que privilegiam a possibilidade de executar diversas tentativas, ou interações, até encontrar as características que fazem os JD prazerosos para os jogadores (*find the fun* ou *fun of the game*). Além disso, Petrillo *et al.* (2008), Godoy & Barbosa (2010), Keith (2010) falam da importância do planejamento e a necessidade de trabalhar com a incerteza. Essas práticas exigem grande interação entre os membros da equipe de desenvolvimento e participação ativa dos Desenvolvedores em reuniões de planejamento, fato que propicia a construção de um ambiente de trabalho no qual a aprendizagem e a cooperação ocorrem de forma contínua.

Encontramos também autores que relatam falhas no desenvolvimento de JD, devido à perda do caráter multidisciplinar das equipes de desenvolvimento. Esse problema acontece quando as práticas laborais de cada especialidade são realizadas isoladamente, quando o profissional se "fecha" dentro da sua *expertise* para relizar o seu trabalho e esquece o caráter multidisciplinar (diferentes áreas do conhecimento trabalhando juntas para resolver um problema) que existe intrínseco ao processo de desenvolvimento de JD (HUNICKE; LEBLANC; ZUBEK, 2004; GODOY; BARBOSA, 2010).

Essas constatações científicas, e o contato direto com equipes de desenvolvimento, deram origem a duas sugestões. A primeira, que tem o objetivo de aumentar o nível de detalhamento e a participação de todos os membros da equipe de desenvolvimento durante a estimativa do trabalho, trata-se de uma modificação na prática gamificada chamada *Poker Planning* (GRENNING, 2002). A segunda diz respeito a uma das etapas de trabalho do *sprint*, e foi adaptada com base na metodologia de desenvolvimento XP (BECK, 2009) e dará origem a um item de verificação do documento com as Definições de Pronto, DOD.

Acreditamos que essas duas sugestões diminuirão os erros de estimativas do trabalho que deve ser desenvolvido, fato que tem impacto direto no planejamento, e minimizarão a possibilidade de individualização do trabalho durante a construção dos JD, fato que contribuirá para produção de JD que expressam os aspectos multidisciplinares criados e discutidos pela equipe de desenvolvimento.

## 4.2 O *Poker Planning* com jogadas colaborativas

A prática apresentada por Grenning (2002), chamada de *Poker Planning*, tem como principal objetivo definir uma estimativa para cada um dos requisitos apresentados pelo cliente. Não existe o comprometimento com a precisão, o propósito maior é construir o escopo do

projeto a partir de uma visão geral dos requisitos que serão implementados.

De acordo com [Cohn \(2005\)](#), o *Poker Planning* ([GRENNING, 2002](#)) é uma prática bem aplicada para a definição do escopo de requisitos de qualquer tipo de Sistema. É muito difícil, e pouco provável, que no início de um projeto seja possível determinar todas as nuances dos requisitos que serão implementados. As metodologias Ágeis não pregam isso; e, além disso, grande parte dos requisitos apresentados pelo cliente não serão implementados ([SUTHERLAND, 2016](#)).

Porém, quando estamos falando sobre planejamento de JD, existem aspectos multidisciplinares que não podem ser ignorados quando estamos estimando os requisitos de um JD. Devido a isso, apresentamos uma dinâmica cuja análise da estimativa leva em consideração as jogadas de todos os jogadores para formar um resultado final, similar a um jogo de *Poker*. As cartas, ou jogadas, que representam a estimativa de cada jogador, serão combinadas durante a discussão entre os membros da equipe para construir a estimativa. Segue abaixo o passo a passo de uma jogada de *Poker Planning* segundo a nossa sugestão:

1. Cada membro da equipe utiliza um conjunto de cartas;
2. O Cliente ou *Product Owner* lê o requisito e o discute com a equipe de desenvolvimento;
3. Cada membro escolhe **as cartas com as suas estimativas**. Duas cartas de tipo (programação ou artes) e duas cartas de tamanho (números);
4. Todos juntos mostram as suas cartas;
5. **No *showdown*, as cartas são combinadas para definir conjuntamente a melhor estimativa para a *user story* apresentada.**

Ao analisar o passo a passo descrito acima, percebemos que somente os itens destacados em vermelho, três e cinco, são diferentes da proposta original. No ponto três, o desenvolvedor escolhe quatro cartas para relizar a estimativa. No ponto cinco, os desenvolvedores utilizam o conjunto de jogadas para, colaborativamente, compor a melhor estimativa, combinando as cartas exibidas pelo desenvolvedores de forma similar ao *showdown* do jogo de *poker* ([HARRINGTON, 2005](#)).

Neste ponto, temos que ressaltar as duas mudanças. A primeira diz respeito ao conjunto de cartas do baralho. Na proposta original de [Grenning \(2002\)](#), as cartas tinham os valores 1, 2, 3, 5, 7, 10 e infinito. Publicações atuais utilizam letras e até animais nos baralhos ([COHN, 2005](#)). Nesta proposta, utilizaremos as cartas 0, 1/2, 1, 2, 3, 5, 8, 13,

20, 40, 100, infinito e ?, ilustradas na figura 4.1, apresentadas em Cohn (2005) e mais duas cartas: uma para representar o trabalho de programação e outra para o trabalho artístico, ilustradas na figura 4.2.

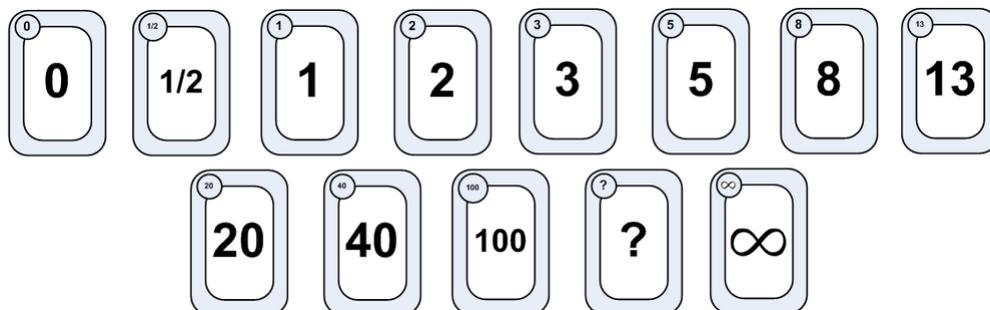


Figura 4.1: Cartas do baralho de *Poker Planning*. Fonte: (COHN, 2005)

Para facilitar o entendimento e dar significado a cada carta, a Tabela 4.1 apresenta a interpretação de cada uma delas. Os valores apresentados na coluna "cartas" da tabela 4.1 representam a quantidade de horas que a tarefa precisa para ser concluída (GRENNING, 2002; DUARTE, 2016). Todavia, devemos ressaltar que esses números servem apenas para dar uma ideia do tempo necessário para execução da tarefa; não existe um modelo ou fórmula que transforme os valores das estimativas do *Poker Planning* em horas, não tem precisão nessa prática e as metodologias Ágeis não pregam isso (SUTHERLAND, 2016). Geralmente, o *Scrum Master* leva dois ou três *sprints* para encontrar a velocidades que a equipe consegue trabalhar, e dessa forma definir quantos pontos a equipe desenvolve em um *sprint* (KEITH, 2010).

Cartas	Interpretação
0	A tarefa já está completa
1/2	A tarefa é muito pequena
1, 2, 3	A tarefa é pequena
5, 8, 13	A tarefa é de tamanho médio
20, 40	A tarefa é grande
100	A tarefa é muito grande
∞	A tarefa é enorme
?	Não sei o tamanho da tarefa

Tabela 4.1: As cartas do baralho de *Poker Planning* e seus significados. Fonte: Adaptado de Cohn (2005)

As duas cartas que a nossa sugestão recomenda adicionar na dinâmica do *Poker Planning*, trabalho de programação e trabalho artístico, ilustradas na Figura 4.2, serão utilizadas para que cada desenvolvedor possa estimar o esforço, de programação e artístico, necessário para realizar a tarefa descrita no requisito do cliente ou *Product Owner* em uma mesma jogada.

A segunda mudança que a nossa dinâmica sugere é a utilização de todas as jogadas, as estimativas de todos os Desenvolvedores, para compor a jogada vencedora e a estimativa com mais detalhes. Todos os Desenvolvedores poderão combinar as cartas, de diferentes jogadas, para construir as estimativas que mais se aproximem da *user story* descrita pelo cliente ou *Product Owner*. Vale ressaltar que não estamos falando sobre precisão. Neste ponto, é importante reafirmar que o objetivo principal dessa dinâmica é aumentar o nível de detalhamento e a participação de todos os membros da equipe de desenvolvimento.

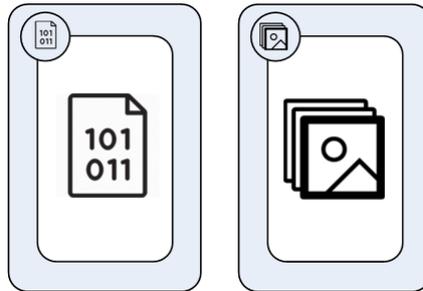


Figura 4.2: A carta da esquerda representa o trabalho de programação. A carta da direita representa o trabalho artístico.

Fonte: elaborado pelo autor

A diferença entre a nossa sugestão e a proposta original está na forma que o requisito é mensurado. [Grenning \(2002\)](#) afirma que o tamanho do requisito é definido após sucessivas interações entre os desenvolvedores, que apresentam argumentos para justificar a sua estimativa. As interações acabam quando todos chegam a um consenso e apresentam a mesma estimativa. Na nossa dinâmica, o esforço do trabalho de programação e artístico é mensurado simultaneamente, cada desenvolvedor apresenta argumentos para aumentar o nível de detalhamento da atividade. Isso proporciona três vantagens:

- O foco da dinâmica não está na "vitória" de um desenvolvedor, que convence seus colegas e encaminha o consenso. A colaboração é o centro da atividade;
- Os aspectos multidisciplinares que surgem na reunião de planejamento do *Sprint*, *Sprint planning*, ficam evidenciados na estimativa;
- Ao final da dinâmica, a estimativa total também apresentará as estimativas das atividades (estimativas parciais) que serão realizadas para o desenvolvimento do requisito definido pelo cliente ou *Product Owner*.

Para exemplificar, vamos criar o seguinte cenário: O *Product Owner* lê um dos requisitos do jogo para uma equipe de três desenvolvedores. Os desenvolvedores tiram suas dúvidas, escolhem as suas cartas e apresentam as jogadas ilustradas na Figura 4.3. A primeira

coisa que podemos notar é que o perfil do desenvolvedor terá influência direta na sua estimativa. Neste cenário hipotético, vamos assumir que a jogada "a" (cartas vermelhas) apresentada na figura 4.3 é a estimativa de um programador, e a jogada "c" (cartas laranjas) a estimativa de um desenhista. Ambos não apresentam estimativas para o esforço necessário para realização do trabalho fora da sua área de experiência. Porém, a jogada "b" (cartas verdes) ilustra a estimativa de um profissional que possui experiência nas duas áreas (programação e desenho). As três jogadas expressam a perícia de diferentes profissionais, é justamente por isso que as três precisam ser analisadas para definição de uma estimativa que possua mais detalhes e apresente o caráter multidisciplinar dessa atividade.

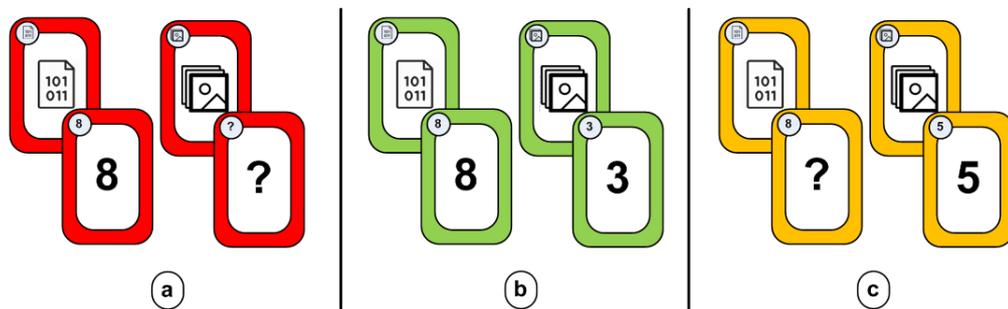


Figura 4.3: Exemplo de estimativa de uma equipe com três desenvolvedores. Fonte: elaborado pelo autor

Ainda utilizando o cenário descrito acima, vamos fazer algumas combinações com as jogadas ilustradas na Figura 4.3 para exemplificar o funcionamento da dinâmica proposta pela nossa sugestão e exibir alguns dos resultados possíveis. Os resultados "a" e "b" retratados na Figura 4.4, apresentam uma curiosidade. Os desenvolvedores concordam quanto à estimativa do trabalho de programação (tamanho 8), mas utilizam a mesma prática definida por Grenning (2002) para definição do trabalho artístico. Na jogada "a", eles definem que o esforço necessário para o desenvolvimento do trabalho artístico é 3, e na jogada "b" eles chegaram a um consenso que o esforço necessário é 5. É importante ressaltar esse tipo de resultado, porque ele demonstra que a dinâmica que nós estamos sugerindo também permite que um desenvolvedor convença os seus colegas de trabalho que uma atividade foi superestimada ou subestimada.

Os resultados "c" e "d" ilustrados na Figura 4.4 apresentam resultados que demonstram as vantagens da nossa sugestão. No resultado "c", os desenvolvedores concordaram que o tamanho do trabalho de programação é 8, porém as estimativas do trabalho artístico, ilustradas na Figura 4.3, apresentaram contribuições distintas (ressaltando que as jogadas foram feitas por desenvolvedores com perfis diferentes) e, devido a isso, as duas foram agregadas à estimativa do trabalho artístico. O resultado "c" da Figura 4.4 define o tamanho 8 para o trabalho artístico, 3 de um aspecto apresentado pelo desenvolvedor "b" e

5 de um ponto levantado pelo desenvolvedor "c", ambos ilustrados no escopo da figura 4.3.

No resultado "d", da Figura 4.4, os desenvolvedores apresentam contribuições distintas para os dois tipos de trabalho. Devido a isso, as contribuições são somadas à estimativa. Como resultado final, temos o tamanho 10 ( $8 + 2$ ) para o trabalho de programação e 8 ( $5 + 3$ ) para o trabalho artístico. Notem que as estimativas finais ilustradas nas jogadas "c" e "d" da Figura 4.4 (estimativa total) também apresentam as estimativas das atividades (estimativas parciais) necessárias para desenvolvimento da *user story*.

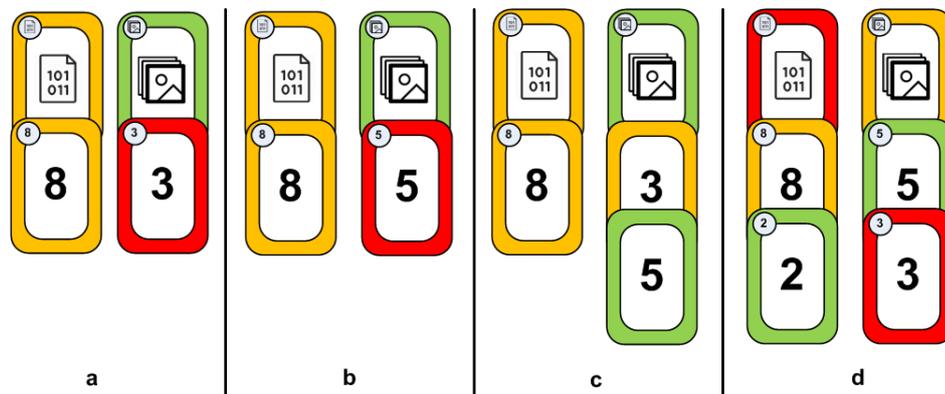


Figura 4.4: Exemplo de estimativa após o *Showdown*. Fonte: elaborado pelo autor

### 4.3 Verificação multidisciplinar

O conjunto de valores e princípios apresentados por Beck (2009) tem o objetivo de criar um estilo de programação que prioriza as interações e os constantes *feedbacks* entre os membros da equipe e o cliente final. Entre as práticas que concretizam os valores e princípios da XP, destacaremos duas que serão a base da nossa segunda sugestão: o Desenvolvimento Guiado por Teste, TDD e a Programação em Pares, *Pair Programming*.

Todo o ciclo de desenvolvimento do XP é voltado para testes. Conforme ilustrado na Figura 4.5, nesta metodologia, você primeiro cria e executa os teste que irão validar a funcionalidade, depois produz as linhas de código para construir a funcionalidade e por fim refatora o seu código para aperfeiçoá-lo. Essa abordagem muda por completo a forma que o *software* é desenvolvido por duas razões: os teste passam a ser o início do trabalho do desenvolvedor. Antes de começar a codificar, o programador é obrigado a pensar no comportamento que o *software* pode assumir. Além disso, depois que os testes estão construídos, eles servirão como um guia de desenvolvimento. Eles induzem os desenvolvedores

a codificar, evitando os erros que os testes provocam (PAULK, 2001; BECK, 2009).

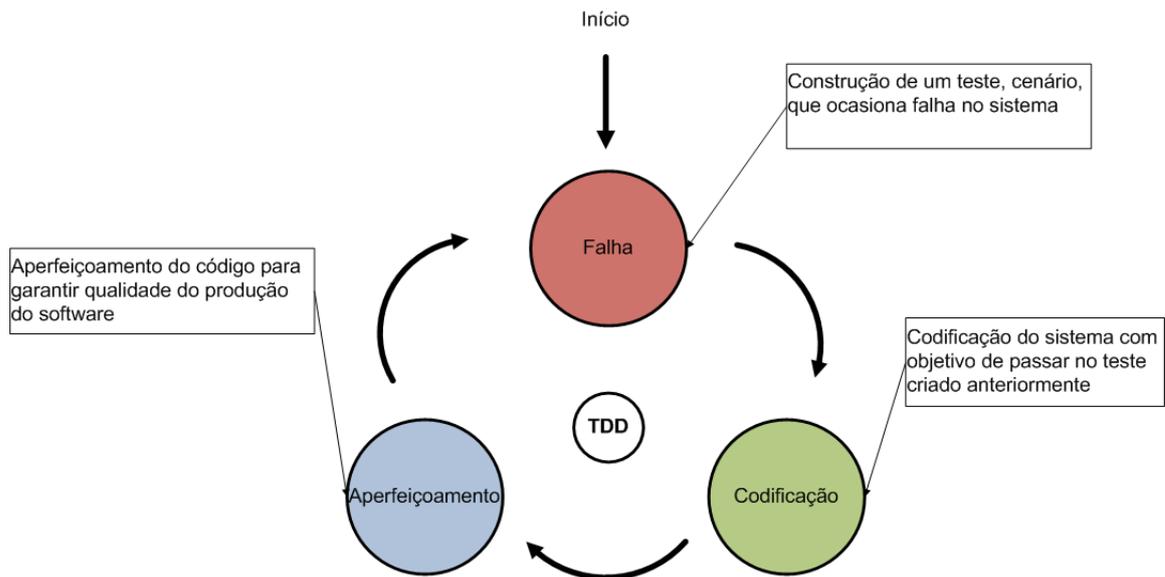


Figura 4.5: Ciclo de Desenvolvimento guiado por testes, TDD. Fonte: Paulk (2001)

Outro ponto central e marcante da metodologia de desenvolvimento XP é a Programação em Pares. A XP busca manter a comunicação fluida entre os membros da equipe de desenvolvimento, e a Programação em pares é uma das práticas que viabilizam a interlocução constante entre eles. A ideia básica da Programação em Pares, ilustrada na Figura ??, é que a codificação é compartilhada entre duas pessoas que utilizam o mesmo computador, assumindo papéis diferentes (piloto e copiloto) e se revezando entre essas duas responsabilidades. O piloto trabalha diretamente no código, construindo a estrutura lógica da funcionalidade que está sendo desenvolvida. O copiloto supervisiona o trabalho do piloto, verificando as estruturas de dados, padrões de projeto, nomenclatura de variáveis e objetos, já pensando na refatoração que ambos farão depois de executar todos os testes (BECK, 2009).

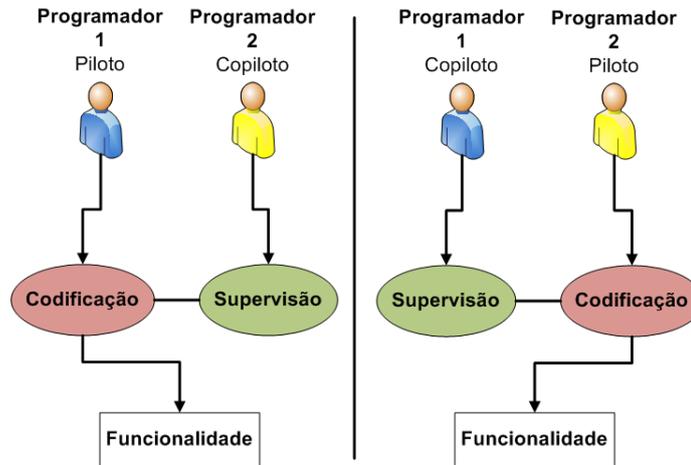


Figura 4.6: Exemplo de Programação em pares. Fonte: (BECK, 2009) ??)

Embora as empresas de desenvolvimento de JD normalmente não apliquem testes pré-definidos antes da criação dos JD (KASURINEN; SMOLANDER, 2014), acreditamos que seja possível a definição de uma rotina de verificações que objetiva analisar o resultado final da integração das criações da equipe de desenvolvimento no *sprint* (fase 1) e a opinião dos usuários finais (fase 2) sobre o produto criado. Essa prática tem o propósito de averiguar a aderência entre a concepção artística, a trilha sonora, jogabilidade e programação do JD, e com isso evitar o fatiamento das ideias geralmente causadas pela individualização do trabalho (HUNICKE; LEBLANC; ZUBEK, 2004; GODOY; BARBOSA, 2010). Para isso construímos um ciclo de verificações, ilustrado na Figura 4.7, similar ao TDD, que deve ser executado por, pelo menos, dois membros da equipe de desenvolvimento, com perfis diferentes (ex.: um programador e um desenhista) e um jogador (usuário final).

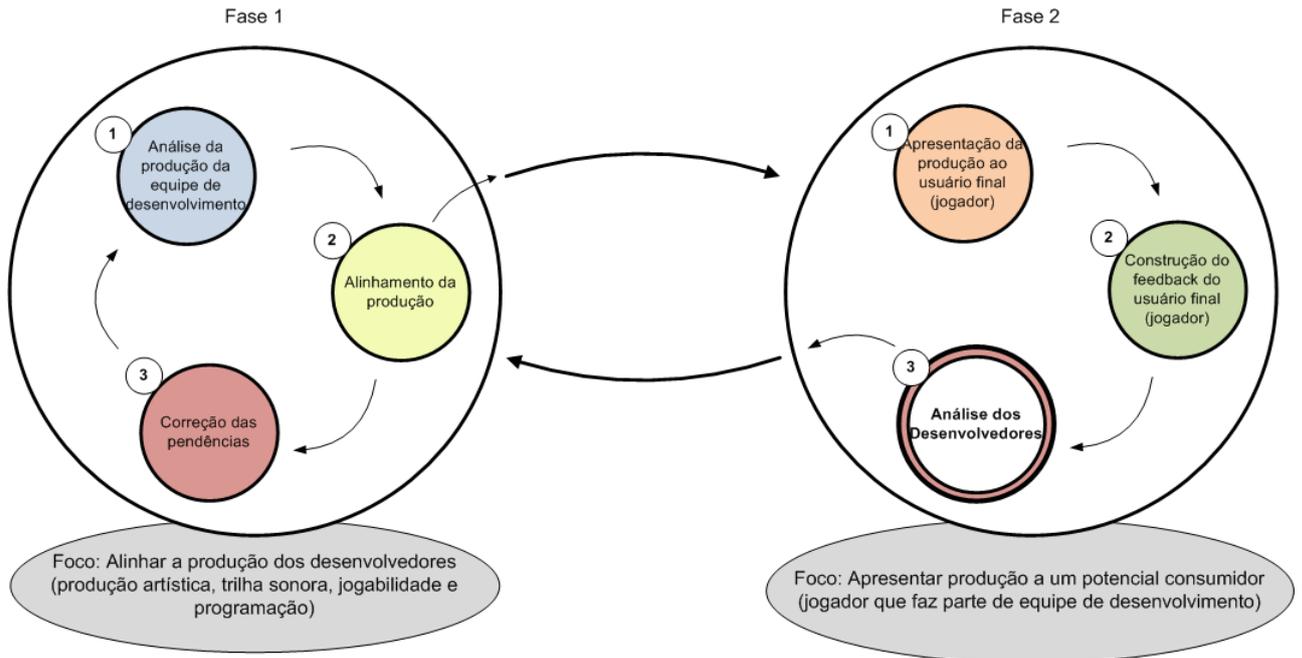


Figura 4.7: Ciclo de Verificações Multidisciplinar. Fonte: elaborado pelo autor

É muito importante ressaltar que o modelo apresentado na Figura 4.7 não se aplica a todos produtos criados em um *sprint*. Ele deve ser empregado a protótipos funcionais. Atividades que objetivam construção das estruturas funcionais como banco de dados, configurações e ajustes de IDE ou *frameworks* não devem adicionar esse item no documento de DOD.

Para facilitar o entendimento do modelo ilustrado na Figura 4.7, vamos considerar que uma equipe de três pessoas (dois desenvolvedores e um jogador) vai efetuar o Ciclo de Verificações Multidisciplinar. Na primeira fase, somente os desenvolvedores participam da verificação. A primeira atividade da Fase 1 é analisar o produto criado no *Sprint*. Os desenvolvedores, que possuem perfis diferentes, vão verificar se o produto final criado no *Sprint* está alinhado com as definições apresentadas na *Sprint planning* pelo *Product Owner*, a partir do ponto de vista da sua área de especialidade. Neste momento, os desenvolvedores não trocam ideias ou sugestões. Eles vão somente analisar se o produto gerado pela integração dos trabalhos dos desenvolvedores possuem, ou expressam as características e propriedades, artísticas ou técnicas, na sua *expertise*.

Na segunda atividade da Fase 1, os desenvolvedores vão apresentar as suas análises sobre o protótipo. É nesta etapa que ocorre o alinhamento entre a concepção artística e técnica do JD que está em desenvolvimento. Possíveis inconsistências na arte ou comportamento

de qualquer um dos artefatos do protótipo são apresentadas e discutidas entre os desenvolvedores. As observações que explicitam a necessidade de modificações no protótipo devem ser encaminhadas para a terceira etapa da Fase 1 (novos itens são adicionados na lista de atividade a fazer). Se não houver modificações, a Fase 1 é encerrada e passamos para Fase 2 da verificação.

A terceira etapa da Fase 1 é o momento de ajustes. As inconsistências que foram observadas na segunda etapa são corrigidas pela equipe de desenvolvedores. Depois de efetuar todas as modificações necessárias, uma nova versão do protótipo é gerada e a Fase 1 do Ciclo de Verificações Multidisciplinar se inicia novamente.

Na primeira etapa da Fase 2, ilustrada da Figura 4.7, o protótipo é apresentado a um jogador, um potencial consumidor do JD que está sendo produzido. Durante esse primeiro momento, o jogador vai avaliar o protótipo e construir uma opinião sobre o produto criado pela equipe de Desenvolvimento, sem a presença dos desenvolvedores. No próximo passo, a segunda etapa da Fase 2, os desenvolvedores e o jogador se reunirão para trocar experiências e construir o *feedback* do jogador. Nesta reunião, os desenvolvedores explicarão os objetivos artísticos e técnicos que eles queriam alcançar com o protótipo apresentado, e ouvirão a opinião do jogador sobre o produto desenvolvido.

Neste ponto, precisamos ressaltar a necessidade de colocar o usuário final, jogador, em contato direto com a equipe de Desenvolvimento, com uma função ativa, participando do processo de construção do JD. Essa necessidade já foi destacada por outros autores (SCHELL, 2014; SALEN; ZIMMERMAN, 2012; KEITH, 2010) e descrita por Fullerton (2014) ao apresentar uma abordagem completamente centrada no jogador, *A Playcentric Approach*. Nesta pesquisa, a necessidade de inserir a percepção do jogador foi descrita no modelo MDJD, ilustrado na Figura 3.1.

Na terceira e última etapa da Fase 2, Figura 4.7, os desenvolvedores analisam o *feedback* do jogador. O resultado dessa avaliação pode levá-los a caminhos diferentes. Se o jogador aprovar a criação da equipe de Desenvolvimento, a Verificação Multidisciplinar é finalizada e o protótipo estará pronto para ser apresentado na *Sprint Review*. Se o jogador fizer observações que explicitem a necessidade de modificações no protótipo, os Desenvolvedores podem fazer duas escolhas:

1. Reiniciar a Fase 1 da Verificação multidisciplinar, assumindo a responsabilidade e o risco de entregar o protótipo no *Sprint* atual;

2. Finalizar a Verificação multidisciplinar e apresentar na *Sprint Review* o protótipo e novas *user stories* que descrevam as observações feitas pelo jogador.

Por fim, chamamos a atenção para uma situação interessante: mesmo quando o jogador aprova o protótipo apresentado na Verificação Multidisciplinar, as observações feitas por ele durante a etapa de construção do *feedback* do jogador na Fase 2 podem dar origem a novas *user stories* que serão apresentadas ao *Product Owner* na *Sprint Review*. Essas observações podem oportunizar novas características e propriedades que o *Product Owner* e os *Stakeholders* não observaram durante a definição dos requisitos do JD.

#### 4.4 Visualizando as sugestões no *framework* Scrum

Depois de apresentar os produtos desenvolvidos nesta pesquisa, vamos demonstrar como eles poderiam ser aplicados no desenvolvimentos dos JD que foram construídos durante a execução dessa investigação. Lembramos que a produção dos quatro produtos apresentados nesta pesquisa não estavam finalizados durante o desenvolvimento dos JD que construímos. Para apresentar a aplicação das nossas sugestões, vamos considerar algumas cenas dos jogos LIPISpace e SimGE.

O primeiro exemplo diz respeito ao *Poker Planning* Colaborativo. Conforme apresenta a Figura 4.8, a nossa primeira sugestão seria aplicada no evento *Sprint Planning* pela equipe de desenvolvedores. Ela está na porta de entrada do ciclo de desenvolvimento e tem impacto direto na definição dos requisitos e funcionalidades que irão fazer parte do *Sprint*. Essa mensuração cooperativa proporciona mais fidedignidade ao planejamento e aumenta o engajamento da equipe de desenvolvedores aos prazos e compromissos assumidos durante o planejamento do *Sprint*.

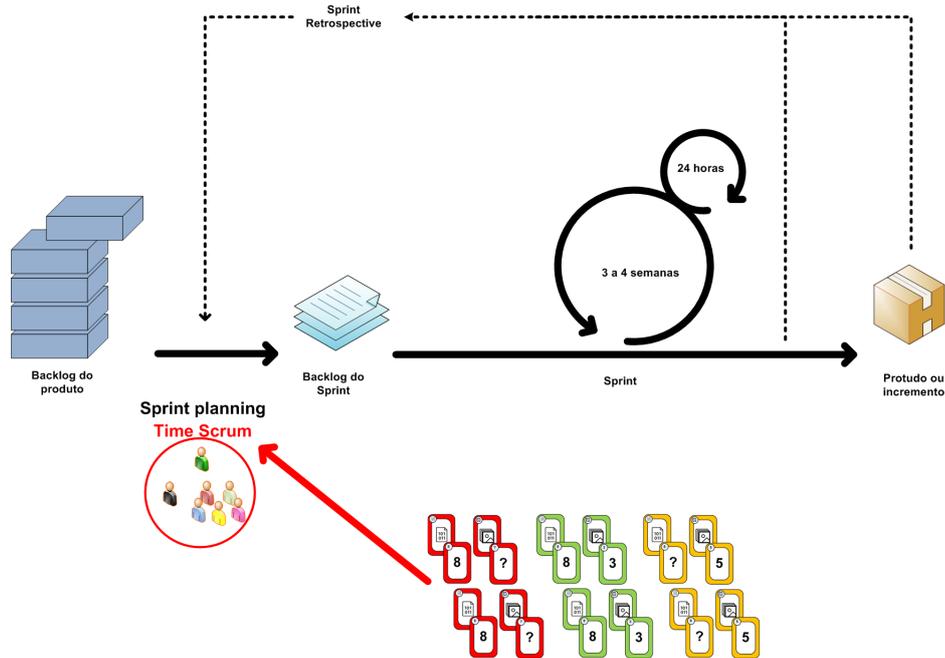


Figura 4.8: Aplicação do *Poker Planning* Colaborativo. Fonte: elaborado pelo autor

As cenas ilustradas na Figura 4.9 apresentam duas funcionalidades do LIPISpace. Na imagem "A", da Figura 4.9, observamos uma nave espacial, dois geradores e entre esses três objetos, um átomo. Os geradores produzem campos magnéticos que influenciam a direção das cargas elétricas que são liberadas pelo átomo, quando ele é estimulado por feixes de luz produzidos pela nave espacial. Além desses objetos, observamos também uma série de controles que comandam a movimentação da nave, o sentido e a direção dos campos elétricos e os três tipos de cargas elétricas (elétrons, prótons e nêutrons).

Na imagem "B", da Figura 4.9, observamos a representação de um átomo segundo o modelo atômico de Niels Bohr. Além disso, temos também as informações e a distribuição eletrônica do átomo e o diagrama de Linus Pauling. Na parte inferior da cena, estão os controles que possibilitam a adição e remoção de elétrons do átomo que está sendo apresentado na tela.

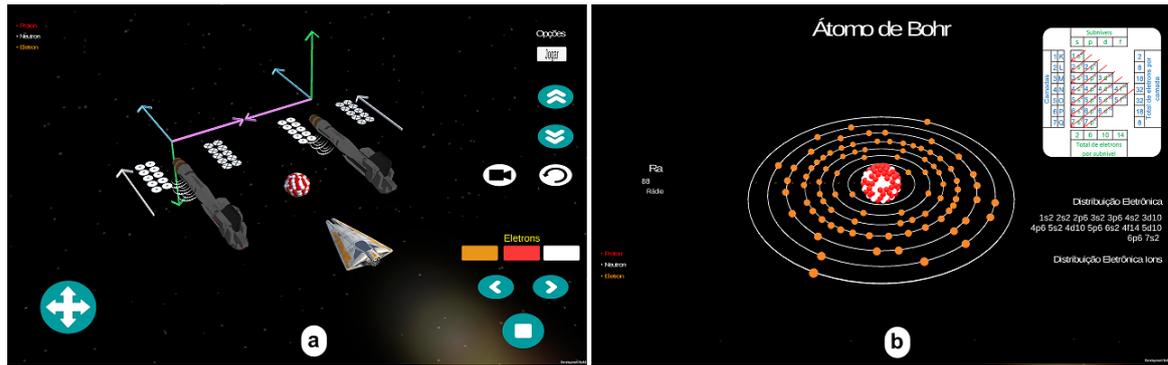


Figura 4.9: Cenas do Jogo Digital LIPISpace. Fonte: elaborado pelo autor

A construção das duas cenas apresentadas na Figura 4.9 foram definidas por duas *user stories* descritas na Figura 4.10. Lembramos que as cenas apresentadas tem o objetivo de unir o caráter lúdico dos Objetos de Aprendizagem aos elementos dos JD (DINIZ; MONTEIRO; CARNEIRO, 2016). Na Figura 4.9 "A", os dois geradores fazem parte da espaçonave. Eles seguem a mesma movimentação da nave. Na Figura 4.9 "B", utilizamos esferas para representar os elétrons, prótons e nêutrons. Essa imagem não está respeitando a proporção real de tamanho dessas partículas. O resultado final ilustrado na Figura 4.9 foi alcançado depois de dois *Sprints*. Todas as cenas foram testadas por professores que validaram os conceitos Físicos, a usabilidade e jogabilidade do LIPISpace.

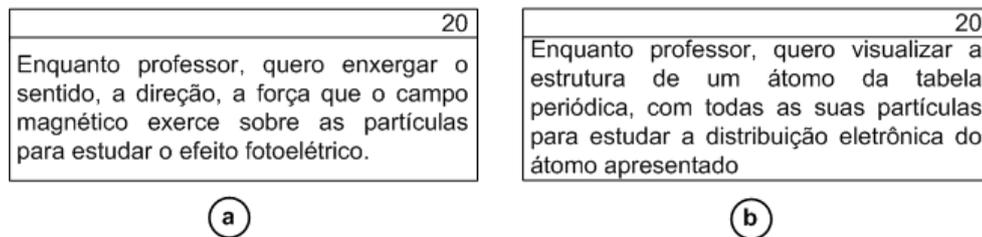


Figura 4.10: *User stories* das cenas ilustradas na Figura 4.9. Fonte: elaborado pelo autor

As duas *user stories* descritas na Figura 4.10 foram classificadas como grandes. Utilizamos o conjunto de cartas sugeridos por Cohn (2005), descrito na Tabela 4.1. A Figura 4.11 ilustra como poderiam ser as jogadas utilizadas para mensurar as *user stories* descritas na Figura 4.10.

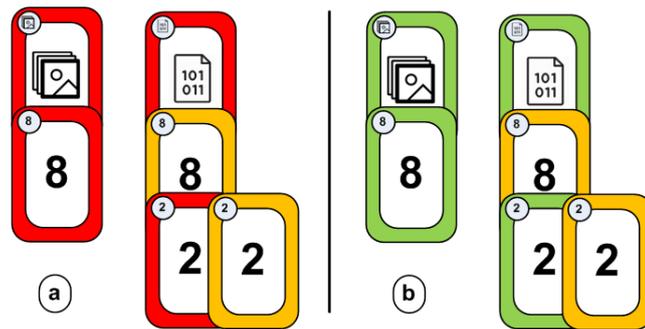


Figura 4.11: .

A imagem "A", representa a jogada da *user storie* descrita na Figura 4.10 "A", e a imagem "B", representa a jogada da *user storie* descrita na Figura 4.10 "B". Fonte: elaborado pelo autor

Para construir o segundo exemplo, iremos continuar utilizando as cenas ilustradas na Figura 4.9. Conforme exposto anteriormente, essas duas cenas foram construídas em dois *Sprints*. No final do primeiro *Sprint*, as duas cenas foram analisadas por dois professores de Física. Durante a análise, eles encontraram três erros conceituais e problemas relacionados a usabilidades.

Como ainda faltavam dois dias para terminar o primeiro *Sprint*, decidimos corrigir os três erros conceituais dentro do próprio *Sprint* e criamos uma nova *user storie*, ilustrada na Figura 4.12, para registrar as sugestões apresentadas pelos professores. Foi a partir dessa experiência que obtivemos as ideias necessárias para construirmos a Verificação Multidisciplinar conforme apresentado na seção 4.3. A Figura 4.13 ilustra o momento que a Verificação Multidisciplinar seria aplicada e a formação da equipe de trabalho responsável pela análise das cenas criadas.

20
Enquanto professor, quero manipular o sentido, a direção e a força do campo magnético para visualizar o efeito que ele exerce sobre as partículas

Figura 4.12: Nova *User storie* criada para adicionar uma nova funcionalidade na cena "A" da Figura 4.9. Fonte: elaborado pelo autor

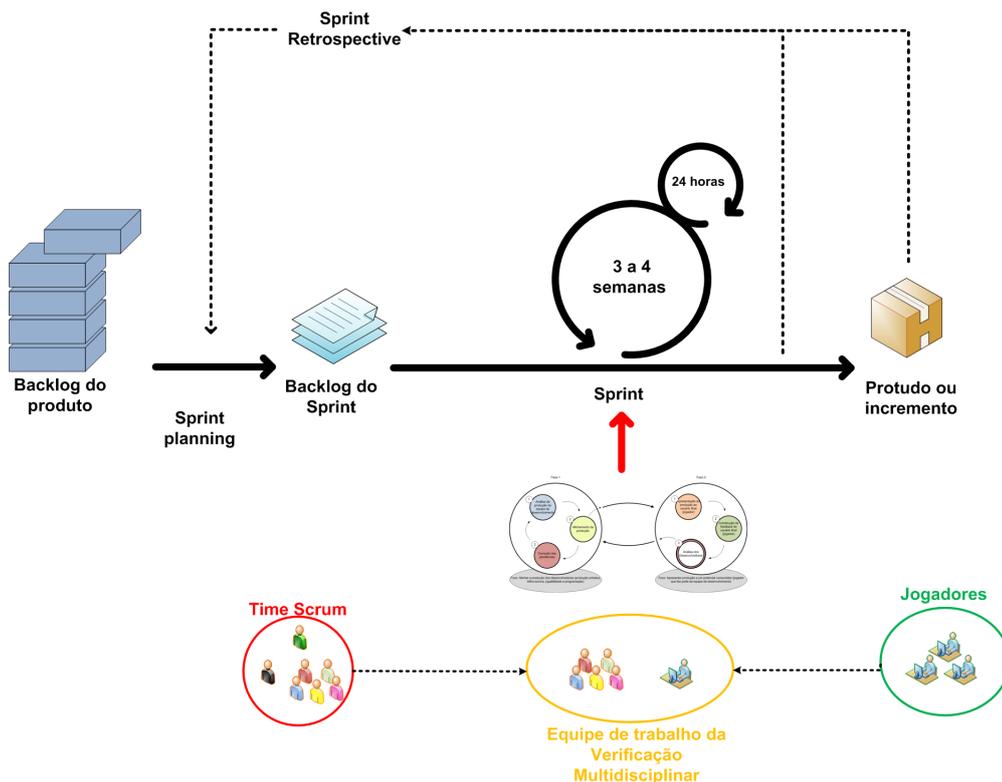


Figura 4.13: Aplicação da Verificação Multidisciplinar. Fonte: elaborado pelo autor

Para apresentar o terceiro exemplo, utilizaremos duas cenas do JD SimGE, ilustradas na Figura 4.14. O SimGE tem como principal objetivo expor os estudantes aos desafios do dia a dia de um gestor escolar. O jogador precisa utilizar os recursos disponíveis para equipar as diversas instalações de uma escola, enquanto responde uma série de perguntas que estão diretamente relacionadas com as leis, práticas e a rotina diária de um gestor escolar.



Figura 4.14: Cenas do Jogo Digital SimGE. Fonte: elaborado pelo autor

A cena ilustrada na Figura 4.14 "A" apresenta uma escola e os seus seis ambientes (secretaria, biblioteca, sala de aula, laboratório de informática, cozinha e o pátio). Na parte superior da cena, temos as informações sobre os recursos que o jogador tem para equipar a escola (recursos de capital e custeio), o valor atual da experiência do jogador e o tempo

que o jogador tem para utilizar os recursos. Ressaltamos que o tempo está diretamente relacionado com o início e fim ano letivo, as datas fiscais a que o gestor precisa obedecer e a experiência do jogador.

A segunda cena apresentada na Figura 4.14 "B" mostra um dos desafios que o jogador precisa resolver durante o jogo. Em intervalos aleatórios, perguntas de múltiplas escolhas são apresentadas para o jogador. Todas as vezes que ele acerta uma pergunta, ele ganha mais recursos de capital e custeio ou mais tempo para utilizá-los. As perguntas fazem parte do conteúdo didático das disciplinas do primeiro e segundo semestres do curso de Gestão Escolar do Programa Profucionário.

Para ilustrar a utilização do Protocolo para descrição de JD, apresentamos na Figura 4.15 uma parte da definição da seção Modelos Matemáticos do bloco Mecânica. Nela exibimos as relações matemáticas que existem entre os conceitos de Recursos de Custeio, Recursos de Capital, Experiência, Tempo e Tipo de Pergunta.

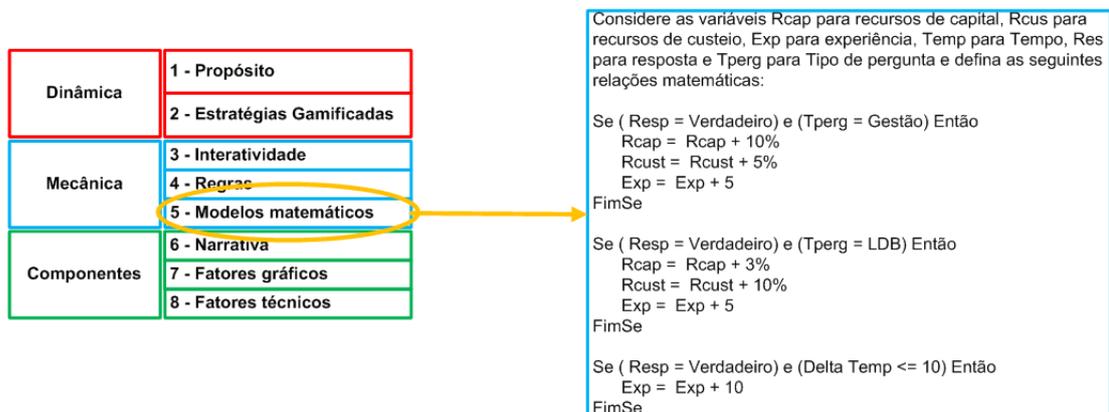


Figura 4.15: Exemplo de utilização de uma das seções do Protocolo para descrição de Jogos Digitais. Fonte: elaborado pelo autor

Conforme ilustrado na Figura 4.16, o Protocolo para descrição de Jogos Digitais dá origem a um documento de *designer* que pode ser utilizado durante a reunião de *Sprint Planning*, pode ser consultado durante a execução do *Sprint* e ainda pode ser atualizado para fazer parte do produto que será apresentado na reunião de *Review* do *Sprint*. A utilização de um protocolo que padronize a descrição dos JD possibilitará, a empresas que possuem processos de desenvolvimento com pouca maturidade ou equipes que possuem membros com pouca experiência no desenvolvimento de JD, um documento guia estruturado e que possui um conjunto de seções que são conceitualmente relacionadas, tendo como base o MDJD.

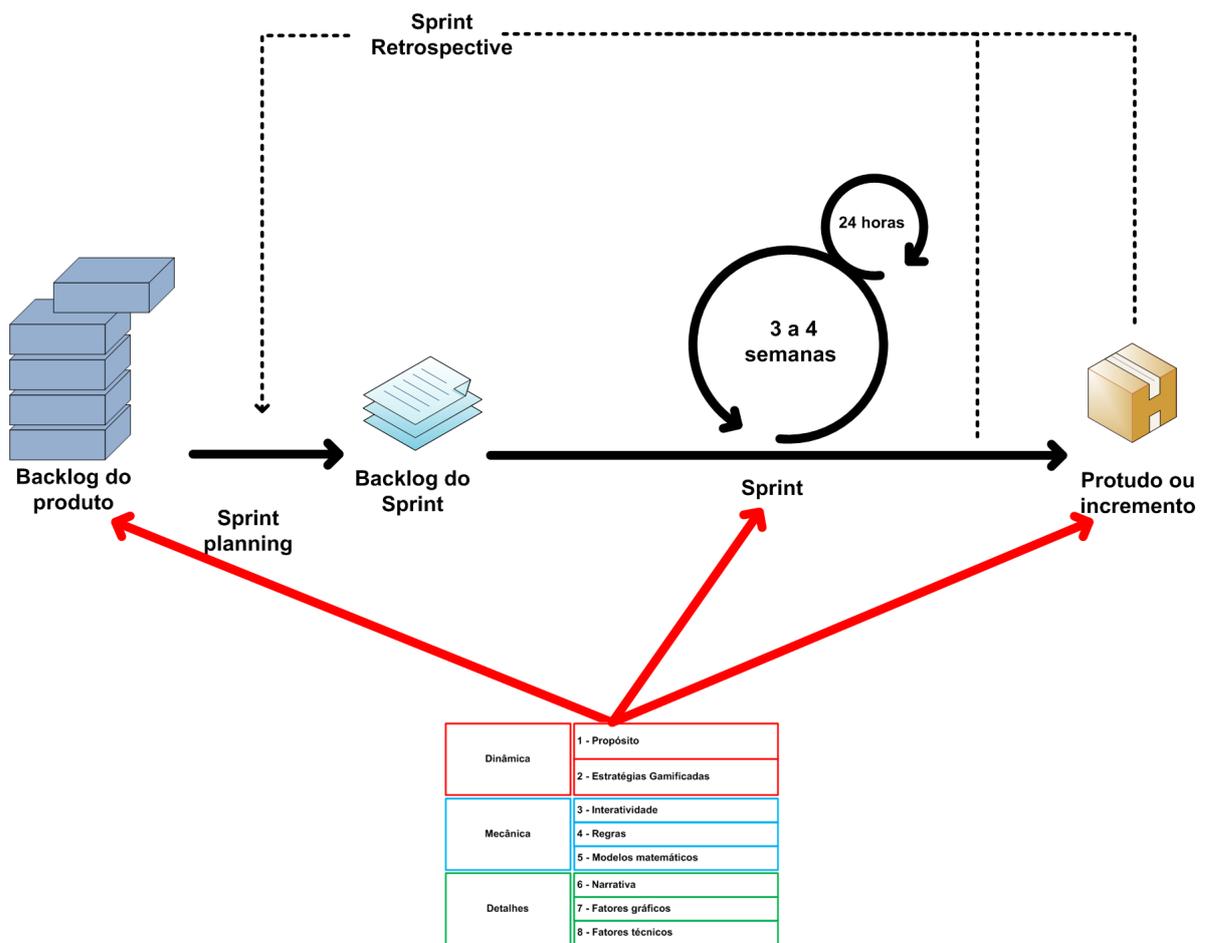


Figura 4.16: Aplicação do Protocolo para descrição de Jogos Digitais. Fonte: elaborado pelo autor

## Conclusões e Considerações finais

Para finalizar o trabalho, neste capítulo apresentam-se as considerações sobre o modelo MDJD, o protocolo para descrição de JD e as sugestões de modificação no *framework* Scrum. Apresentam-se também outras contribuições da tese e as perspectivas de novas investigações.

### 5.1 Conclusões

Com base no modelo construído, apresentamos um protocolo para descrição de JD. E, após identificar problemas em pontos cruciais do processo de desenvolvimento de JD (planejamento e desenvolvimento), definimos duas práticas que viabilizam a construção de JD sem deixar de lado duas características que são intrínsecas ao processo de desenvolvimento de JD: a incerteza e a aprendizagem constante.

O Modelo de Desenvolvimento de Jogos Digitais, ilustrado na Figura 3.1, aglutina e apresenta um conjunto de conceitos que norteiam o desenvolvimento de JD. Nele, podemos notar dois pontos importantes: O primeiro diz respeito aos atores envolvidos no processo. As diferentes perspectivas dos Desenvolvedores e Jogadores são apresentadas no modelo. A partir delas, podemos observar a importância do planejamento, do trabalho artístico e técnico (programação) no desenvolvimento de JD. O equilíbrio entre essas três forças viabiliza a construção de JD que atende às demandas e anseios dos jogadores. Porém, para alcançar esse estado de equilíbrio, é preciso utilizar uma metodologia de desenvolvimento que possibilite trabalhar com a incerteza dos constantes ajustes e modificações, que possibilite a adaptação das suas práticas e que possua o foco sempre voltado para o aprimoramento do produto. Essas características e necessidades justificam a escolha do *framework* Scrum como modelo de gestão dessa pesquisa.

O segundo ponto positivo que o MDJD proporciona está relacionado com o caráter prático que ele oportuniza durante tomadas de decisão em atividades de planejamento e na concepção dos JD. O relacionamento que existe entre os elementos das três categorias do MDJD orienta os *Game Designers* a definir uma estratégia de gamificação, e a relaciona com um conjunto de regras e componentes dos JD. Esse agrupamento irá compor o Sistema Fechado e Formal que irá gerar as respostas emocionais que o *Game Designers*

deseja proporcionar ao jogador. Para agentes cujas funções estão mais voltadas para o planejamento, como *Product Owner's* e *Scrum Master's*, o relacionamento entre os elementos das três categorias pode ajudar a definir a prioridade dos requisitos que irão entrar no *Sprint* e a ordem das atividades executadas.

O protocolo para descrição de JD é o segundo produto apresentado nesta tese. Ele foi criado a partir de uma leitura do MDJD e tem como principal objetivo a descrição formal dos JD. Embora diversos autores utilizem e defendam os JD como objetos de ensino e aprendizagem e realizem pesquisas para aperfeiçoar e potencializar a aplicação dos JD na Educação, não encontramos publicações que apresentem propostas para descrição formal de JD. Devido isso, os JD ainda são descritos de forma verbal, fato que dificulta a reprodução das experiências proporcionadas pelos JD, devido à ambiguidade e leitura pouco acessível. Um protocolo que padronize a descrição de JD resolve esse problema e aumenta a credibilidade científica desse artefato.

A grande vantagem que um protocolo padrão para descrição de JD nos traz está na estrutura que ele oferece aos escritores e leitores. Ela facilitará a absorção de informação graças ao significado de cada camada da sua estrutura. Devido a isso, acreditamos que um protocolo que apresente as explicações verbais separadas das especificações matemáticas, em uma estrutura básica e comum a todos os tipos de JD vai, ajudar a reprodução dos Sistemas de jogo e das respostas emocionais planejadas pelos *Game Designers*.

A primeira sugestão de mudança que fizemos sobre as práticas realizadas no ciclo de desenvolvimento do Scrum altera a dinâmica do *Poker Planning* e tem como principal objetivo aumentar a participação e, conseqüentemente, a colaboração entre os membros da equipe de Desenvolvimento durante a reunião de planejamento do *Sprint*. A ideia central é não deixar aspectos multidisciplinares de fora das estimativas através de uma dinâmica que possibilite que todos participantes (que possuem perfis profissionais diferentes) opinem sobre todos os requisitos apresentados na *Sprint Planning*.

Essa prática nos traz benefícios diretos no planejamento e na concepção dos JD devido a dois pontos importantes. A nossa proposta adiciona no baralho duas cartas, ilustradas na figura 4.2, que explicitam a necessidade de examinar os requisitos apresentados de uma forma multidisciplinar. Isso diminui a possibilidade do requisito ser subestimado, visto que as diferentes perspectivas estarão representadas nas cartas no momento do *showdown*. Além disso, a necessidade de apresentar estimativas para as duas grandes áreas que estão envolvidas no processo de desenvolvimento dos JD evidencia a necessidade de cooperação.

A estimativa é elaborada colaborativamente. Desenvolvedores com perfis diferentes irão construir a estimativa a partir dos seus diferentes pontos de vista.

A segunda sugestão apresentada nesta pesquisa, a Verificação Multidisciplinar, apresentada na Figura 4.7, é um ciclo de teste que dá origem a um novo item de análise para o documento de *Definition of Done* e tem como propósito maior corrigir inconsistências conceituais geradas durante o processo de desenvolvimento dos JD. Essa prática de avaliação testa o protótipo desenvolvido no *Sprint* a partir do ponto de vista de, pelo menos, três pessoas diferentes (dois desenvolvedores e um jogador) que possuem perspectivas e anseios distintos. Além disso, ela traz a figura do jogador, o usuário final, para dentro do processo de desenvolvimento dos JD.

A primeiro benefício que a Verificação multidisciplinar nos propicia é diminuir, ou eliminar, problemas causados pela individualização do trabalho. Ao colocar desenvolvedores com perfis profissionais distintos para analisar o protótipo que foi criado no *Sprint*, nós criamos uma grande oportunidade para a equipe de desenvolvimento aperfeiçoar o produto que está sendo desenvolvido sob um perspectiva multidisciplinar. Ou seja, problemas gerados pelo fatiamento do trabalho como ajuste da paleta de cores, ritmos e batidas da trilha sonora, comportamento dos personagens e dos objetos das cenas, sincronização entre a movimentação dos personagens e efeitos, serão observados e corrigidos.

Além disso, a análise que é feita pelo jogador na segunda fase da Verificação multidisciplinar aproxima a equipe de desenvolvimento do cliente final e do mercado consumidor, fato que possibilita o planejamento de novas características com base no usuário final. As novas possibilidades apontadas pelo jogador podem gerar jogos com mais conteúdo, valor e inovação para os JD. Essas características são importantes para o desenvolvimento de produtos que têm a criatividade como base de construção.

Acreditamos que as duas práticas sugeridas nesta pesquisa podem ser aplicadas em qualquer metodologia Ágil. No entanto, ressaltamos que elas são mais aplicáveis ao *framework* de desenvolvimento Scrum, devido ao ciclo de amadurecimento constante do processo, com o foco voltado para o aperfeiçoamento do produto que está sendo desenvolvido.

## 5.2 Contribuições diretas

Os quatro artefatos apresentado neste trabalho doutoral (o MDJD, o Protocolo para Descrição de JD, O *Poker Planning* colaborativo e a Verificação Multidisciplinar) têm o objetivo de auxiliar o planejamento e desenvolvimento de JD. Todo processo de construção desses artefatos foi gradual e teve suporte das experiências vivenciadas durante o desenvolvimento dos trabalhos descritos na Seção 5.3.

O artigo que publiquei, juntamente com os meus orientadores, foi um marco importante para construção do MDJD e deu início à produção dos outros artefatos dessa tese. [Diniz, Monteiro & Carneiro \(2016\)](#) apresenta um modelo que constrói um *link* direto entre os Elementos da Gamificação e os Objetos de Aprendizagem.

Essa pesquisa nasceu da vontade de apresentar uma proposta concreta e prática para construção de Objetos de Aprendizagem Gamificados. [Diniz, Monteiro & Carneiro \(2016\)](#) apresenta um modelo que relaciona os conceitos fundamentais dos Objetos de Aprendizagem com os componentes que viabilizam o engajamento nos JD. Além disso, esse artigo apresenta um passo a passo de como inserir os elementos da gamificação nos Objetos de Aprendizagem.

Essa publicação teve uma grande influência no desenvolvimento desta pesquisa porque, durante a sua realização, notamos a natureza estrutural que deveríamos aplicar na construção do modelo MDJD. Chegamos a essa conclusão porque qualquer sequência definida de passos ou atividades que especifiquem como devemos construir JD, limitará a idealização e concepção deles, devido a duas características que são implícitas ao processo de construção de JD: a incerteza e a aprendizagem ([KEITH, 2010](#); [SALEN](#); [ZIMMERMAN, 2012](#); [SCHELL, 2014](#)).

## 5.3 Contribuições indiretas

Além da construção dos artefatos que estão diretamente relacionados com o desenvolvimento desta pesquisa, eu tive a oportunidade de participar da produção de outros trabalhos acadêmicos, dois JD e três capítulos de livro, que contribuíram muito para o meu amadurecimento profissional.

Os dois jogos digitais desenvolvidos, SimGE <sup>1</sup> e o LIPISpace, <sup>2</sup> foram registrados no INPI pelo IFBA, devido ao financiamento que a instituição disponibilizou para desenvolvimento dos JD.

O SimGE é um JD inspirado nos jogos de simulação de vida real com o foco voltado para Gestão Escolar. Neste jogo, o estudante é responsável pela administração de uma escola, e, para obter recursos para equipá-la, ele precisa responder a uma série de questões que estão relacionadas com o dia a dia da administração escolar.

O LIPISpace é um *space shooter*, um jogo de naves especiais, no qual o estudante precisa destruir uma série de asteroides e naves inimigas utilizando descargas elétricas que sofrem influência de um campo magnético. Além disso, no LIPISpace, o estudante tem contato com simuladores que estimulam o estudo e a análise dos conceitos de campo magnético e do modelo atômico de Niels Bohr.

Os três capítulos de livros dos quais participei, [Alves, Minho & Diniz \(2014\)](#), [Santos, Souza & Diniz \(2015\)](#), [Hohenfeld, Lapa & Diniz \(2016\)](#), retratam momentos diferentes que vivenciei durante o processo de desenvolvimento de minha tese doutoral.

O primeiro fala sobre o conceito de gamificação e apresenta a aplicação dessas práticas em uma instituição de ensino da região metropolitana de Salvador ([ALVES; MINHO; DINIZ, 2014](#)). O segundo relata a construção de um site que utiliza a Realidade Aumentada como ferramenta de ensino e aprendizagem da língua brasileira de sinais ([SANTOS; SOUZA; DINIZ, 2015](#)). Ele é resultado de um projeto acadêmico realizado no IFBA para prover ferramentas de ensino para estudantes com necessidades especiais.

O terceiro capítulo de livro descreve os resultados obtidos em um projeto de cooperação entre Institutos Federais e o Ministério de Educação da Inglaterra, o projeto STEM. O IFBA aplicou os seus recursos, através do LIPI, no desenvolvimento de experimentos, Objetos de Aprendizagem e Jogos Digitais que viabilizavam o ensino de Física e Ciências a estudantes do ensino médio ([HOHENFELD; LAPA; DINIZ, 2016](#))

---

<sup>1</sup>Jogo Digital Educacional registrado no INPI: Número de Registro: 512015000928-1

<sup>2</sup>Jogo Digital Educacional registrado no INPI: Número de Registro: 512016001116-5

## 5.4 *Contribuições enquanto pesquisador*

Durante o meu processo de doutoramento (atividades, disciplinas e pesquisa), eu tive a oportunidade de coordenar um projeto educacional que me colocou dentro da equipe de desenvolvimento de um JD, o SimGE. As experiências que eu vivenciei durante a construção de um JD, que precisava apresentar um conteúdo educacional e atender aos anseios de um público-alvo, foram muito importantes para minha formação.

Além disso, a necessidade de trabalhar com uma equipe multidisciplinar que apresentava ideias diferentes sobre a utilização de Jogos Digitais como uma ferramenta educacional, colocou-me em contato direto com pessoas com formações distintas e diferentes perspectivas do processo de ensino e aprendizagem. A construção desse JD foi o trabalho colaborativo mais desafiador que vivenciei.

Porém, a experiência mais gratificante que tive durante o meu doutoramento foi observar um grupo de profissionais com pouca experiência de desenvolvimento de JD, superar limites e atingir os objetivos que inicialmente nós julgávamos improváveis.

## 5.5 *Atividades Futuras de Pesquisa*

O primeiro ponto que precisamos investigar depois da conclusão desta pesquisa doutoral é a aceitação dos produtos aqui apresentados (modelo, protocolo e as duas práticas sugeridas) junto ao meio acadêmico e perante a comunidade de desenvolvedores de JD. Desenvolveremos uma pesquisa aplicada a profissionais (professores e desenvolvedores) ligados à área de desenvolvimento e pesquisa de JD, para verificar a aderência dos resultados gerados nesta pesquisa no ambiente acadêmico e na indústria de desenvolvimento de JD. Os resultados gerados pela pesquisa de aceitação dos produtos apresentados neste trabalho serão utilizados para aprimorar o MDJD.

Além da ação que está diretamente relacionada com o desenvolvimento natural desta pesquisa doutoral, pretendo atuar na formação e aperfeiçoamento de professores (nas Licenciaturas em Computação, Física e Matemática e no Programa de Pós-Graduação Latus Sensu em Educação Profissional) no Instituto Federal da Bahia, utilizando os Jogos Digitais como objeto de estudo.

---

## Um exemplo de utilização do Protocolo para descrição de Jogos Digitais

---

Neste apêndice apresentaremos um exemplo de aplicação do Protocolo para descrição de Jogos Digitais. Para facilitar o entendimento, escolhemos um JD clássico e bem conhecido pelo público geral.

### *A.1 Apresentação do Jogo Digital Tetris*

Tetris é um jogo de estratégia criado por Alexey Pajitnov em 1984. Na década de 1980, Pajitnov era um programador e a sua principal atividade era testar a potencialidade de equipamentos construídos pela antiga União Soviética. Esse trabalho o colocava frente aos mais novos equipamentos criados pela URSS e em contato direto com os cientistas mais competentes da época e com suas ideias ([TETRIS, 1996](#)).

Esse ambiente produtivo deu a Pajitnov o conjunto de ferramentas e ideias para criar um JD no qual o jogador deveria arranjar as peças de um quebra-cabeça que iam caindo em tempo real. Pajitnov chamou o jogo de Tetris, a junção de duas palavras: Tetra, palavra grega que significa quatro, e Tennis, o esporte que ele mais gostava ([TETRIS, 1996](#); [RABIN, 2012](#)).

Atualmente, JD Tetris está disponível nas mais diversas plataformas computacionais (computadores, consoles de *video games*, *video games* portáteis, celulares, *smartphones*, etc), mas o seu lançamento comercial foi feito pela Nintendo em 1989, em uma exposição chamada *Consumer Electronics Show* em Las Vegas ([RABIN, 2012](#)).

O grande sucesso do lançamento ocorreu devido a dois fatores: o primeiro foi o gênero do jogo. Tetris cativou, e até os dias atuais cativa, jogadores de ambos os sexos e de uma larga faixa de idade. O segundo motivo foi a plataforma computacional utilizada pela Nintendo. O *Game Boy*, ilustrado na Figura [A.1](#), é um *video game* portátil lançado em 1989, que possibilitou que os jogadores levassem os seus jogos para qualquer lugar. A combinação desses dois fatores fez do Tetris o passatempo ideal para todos que tivessem um tempo livre entre as suas atividades ([RABIN, 2012](#)).



Figura A.1: *Game Boy*, console portátil da Nintendo lançado em 1989. Fonte: ([NINTENDO, 1989](#)).

## A.2 Descrição do Tetris

A descrição apresentada na Tabela [A.1](#) terá como base o jogo Tetris na sua versão original de 1989 (com sete peças de tipos diferentes) e a plataforma movel *Game Boy* de 1989, conforme ilustrado a Figura [A.1](#).

Bloco	Seção	Definição
Dinâmica	1- Propósito	O jogador deverá organizar as peças de quebra cabeça que cairão da parte superior da cena sem deixar espaços entre elas.
	2- Estratégias Gamificadas	1 - O jogador não pode deixar a pilha de peças alcançar a parte superior da cena; 2 - A cada nova fase as peças irão se mover mais rápido; 3 - Existem sete peças com formatos diferentes.
Mecânica	3- Interatividade	O jogador utilizará os botões direcionais para movimentar as peças para os lados e para baixo. O botão "A" rotaciona as peças, a tecla "Start" inicia ou para o jogo e a tecla "Select" dá ao jogador a opção de sair do jogo. Tetris pode ser jogado por dois jogadores simultaneamente se os dois consoles <i>Game Boy</i> estiverem conectados via cabo serial.
	4- Regras	1 - O jogador não pode deixar as linhas incompletas de peças empilhadas chegar até o topo da cena; 2 - Quando uma linha de peças é completada, ela desaparecerá e o jogador ganhará pontos extra.
	5- Modelos matemáticos	Utilize as seguintes definições matemáticas para pontuar, seguindo o número de linhas eliminadas pelo jogador. 1 - Uma linha eliminada = $(40 * (n + 1))$ ; 2 - Duas linhas eliminadas = $(100 * (n + 1))$ ; 3 - Três linhas eliminadas = $(300 * (n + 1))$ ; 4 - Quatro linhas eliminadas = $(400 * (n + 1))$ ; Em todas as quatro definições "n" representa o atual nível de dificuldade do jogo. O valor de "n" pode variar de 0 a 9.

Tabela A.1: Descrição do Jogo Digital Tetris. Fonte: elaborado pelo autor

Bloco	Seção	Definição
Componentes	6- Narrativa	Não se aplica. Tetris é um <i>casual game</i> .
	7- Fatores gráficos	As sete peças, ilustradas na Figura A.2, serão direcionadas e rotacionadas pelos controles do console. O jogador definirá como elas se combinarão para formar uma linha sem espaços de um lado a outro da cena.
	8- Fatores técnicos	O JD será desenvolvido para executar em um console com o processador Custom 8-bit Sharp de 4.19 MHz, 8 Kb de memória RAM, 8 Kb de memória de vídeo e uma paleta de cores de dois bits com quatro tons de cinza. Os JD ficarão armazenados em cartuchos do tipo On-CPU-Die de 256-bytes e um console pode se conectar a outro através de um cabo serial, conforme ilustrado na Figura A.3, para permitir que dois jogadores jogem juntos um mesmo jogo.

Tabela A.2: Continuação da Tabela A.1. Fonte: elaborado pelo autor

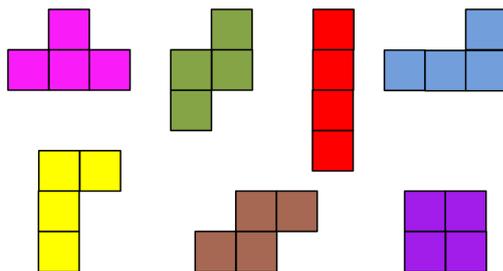


Figura A.2: As sete peças originais do Jogo Digital Tetris lançado em 1989. Fonte: (TETRIS, 1996).



Figura A.3: Conexão entre dois consoles *Game Boy*. Fonte: (TETRIS, 1996).

---

## Referências Bibliográficas

---

- ABT, Clark C. *Serious games*. [S.l.]: University Press of America, 1987.
- ÁGIL, Manifesto. Manifesto para o desenvolvimento ágil de software. *Disponível em: <http://manifestoagil.com.br/>*. Acessado em, v. 17, 2011.
- ALBINO, Raphael Donaire; SOUZA, Cesar Alexandre De; PRADO, Edmir Parada Vasques. Benefícios alcançados por meio de um modelo de gestão ágil de projetos em uma empresa de jogos eletrônicos. *Revista de Gestão e Projetos*, Universidade Nove de Julho (UNINOVE), v. 5, n. 1, p. 15, 2014.
- ALVES, Lynn. Videojogos e aprendizagem: mapeando percursos. *Carvalho, A.(2012) Aprender na era digital Jogos e Mobile-Learning*, p. 11–28, 2012.
- ALVES, Lynn RG; MINHO, Marcelle RS; DINIZ, Marcelo VC. Gamificação: diálogos com a educação. *Gamificação na educação. São Paulo: Pimenta Cultural*, p. 74–97, 2014.
- BAGNALL, Brian. *On the edge: the spectacular rise and fall of Commodore*. [S.l.]: Variant Press, 2005.
- BARTLE, Richard A. *Designing virtual worlds*. [S.l.]: New Riders, 2004.
- BECK, Kent. *Programação Extrema (XP) Explicada*. [S.l.]: Bookman Editora, 2009.
- BLESS, Marc. Distributed meetings in distributed teams. In: SPRINGER. *International Conference on Agile Software Development*. [S.l.], 2010. p. 251–260.
- BOEHM, Barry W. A spiral model of software development and enhancement. *Computer*, IEEE, v. 21, n. 5, p. 61–72, 1988.
- BRANSFORD, John D; BROWN, Ann L; COCKING, Rodney R. *How people learn: Brain, mind, experience, and school*. [S.l.]: National Academy Press, 1999.
- BRASIL, GAME. *Pesquisa Game Brasil 2016*. 2015.
- \_\_\_\_\_. *Pesquisa Game Brasil 2017*. 2016.
- CALLOIS, Roger. Os jogos e os homens: a máscara e a vertigem. *Lisboa: Cotovia*, 1990.
- CAMARGO, Caio. Interesting complexity: Sid meier and the secrets of game design. *Crossroads*, ACM, v. 13, n. 2, p. 4–4, 2006.
- CARNEIRO, Tereza. Redes de afinidade como estratégia de gestão pedagógica e difusão do conhecimento em cursos na modalidade a distância. Faculdade de Educação, 2015.
- CHOU, Yu-Kai. *Actionable Gamification: Beyond Points, Badges, and Leaderboards*. [S.l.: s.n.], 2015.
- COHEN, Scott. *Zap!: The Rise and Fall of Atari*. [S.l.]: McGraw-Hill Companies, 1984.
- COHN, Mike. *Agile estimating and planning*. [S.l.]: Pearson Education, 2005.

- COSTIKYAN, Greg. Don't be a vidiot: What computer game designers can learn from non-electronic games. *Retrieved September*, v. 26, p. 2005, 1998.
- COUGHLAN, Paul; COGHLAN, David. Action research for operations management. *International journal of operations & production management*, MCB UP Ltd, v. 22, n. 2, p. 220–240, 2002.
- CRAWFORD, Chris. The art of computer game design. Osborne/McGraw-Hill Berkeley, CA, 1984.
- CSIKSZENTMIHALYI, Mihaly. *Finding flow: The psychology of engagement with everyday life*. [S.l.]: Basic Books, 1997.
- DEANGELIS, DL; GROSS, LJ. *Individual based models and approaches in ecology: concepts and models*. [S.l.]: Routledge, Chapman and Hall, New York, 1992.
- DETERDING, Sebastian *et al.* Gamification. using game-design elements in non-gaming contexts. In: ACM. *CHI'11 Extended Abstracts on Human Factors in Computing Systems*. [S.l.], 2011. p. 2425–2428.
- DINIZ, Marcelo VC; MONTEIRO, Roberto LS; CARNEIRO, Tereza KG. Elementos da gamificação nos objetos de aprendizagem. *Revista Tecnologias na Educação*, p. 1–12, 2016.
- DUARTE, Luiz. *Scrum e Métodos Ágeis: Um Guia Prático*. [S.l.]: LuizTools, 2016.
- FARDO, Marcelo. A gamificação como método: estudo de elementos dos games aplicados em processos de ensino e aprendizagem. Dissertação (Mestrado)–Universidade Caxias do Sul, Programa de Pós-Graduação em Educação, 2013.
- FILHO, Marisardo Medeiros *et al.* A importância da prototipação no design de games. *SBC–Proceedings of SBGames*, 2013.
- FLEURY, Afonso; NAKANO, Davi; CORDEIRO, JHDO. Mapeamento da indústria brasileira e global de jogos digitais. *São Paulo: GEDIGames/USP*, 2014.
- FULLERTON, Tracy. *Game design workshop: a playcentric approach to creating innovative games*. [S.l.]: CRC press, 2014.
- GÅSLAND, Magne Matre. Game mechanic based e-learning: A case study. Institutt for datateknikk og informasjonsvitenskap, 2011.
- GEE, James Paul. What video games have to teach us about learning and literacy. *Computers in Entertainment (CIE)*, ACM, v. 1, n. 1, p. 20–20, 2003.
- GODOY, André; BARBOSA, Ellen F. Game-scrum: An approach to agile game development. *Proceedings of SBGames*, p. 292–295, 2010.
- GREGORY, David. *Building a Mindset for Rapid Iteration Part 1: The Problem*. 2010.
- GRENNING, James. Planning poker or how to avoid analysis paralysis while release planning. *Hawthorn Woods: Renaissance Software Consulting*, v. 3, 2002.
- GRIMM, Volker *et al.* A standard protocol for describing individual-based and agent-based models. *Ecological modelling*, Elsevier, v. 198, n. 1, p. 115–126, 2006.

\_\_\_\_\_. The odd protocol: a review and first update. *Ecological modelling*, Elsevier, v. 221, n. 23, p. 2760–2768, 2010.

HARRINGTON, Dan. *Harrington on Hold'em: Expert Strategy for No-Limit Tournaments; Volume II: the Endgame*. [S.l.]: Two Plus Two Publishing LLC, 2005. v. 2.

HENSE, Jan; MANDL, Heinz. Learning in or with games? quality criteria for digital learning games from the perspectives of learning, emotion, and motivation theory. In: *Proceedings of the IADIS International Conference on Cognition and Exploratory Learning in the Digital Age*. [S.l.: s.n.], 2012. p. 19–26.

HOHENFELD, D. P.; LAPA, J. M.; DINIZ, M. V. C. Comunidade de práticas de ensino de física. *Fernanda Helena Nogueira-Ferreira; Pâmela Billig Mello-Carpes; Cristiane Maria Sampaio Forte; Cristiane Mansur de Moraes Souza. (Org.). Caminhos do STEM. Um Programa de Educação Científica: Relatos e Produtos*, p. 285–300, 2016.

HUIZINGA, Johan. *Homo ludens: o jogo como elemento da cultura*. [S.l.]: Editora da Universidade de S. Paulo, Editora Perspectiva, 1931.

HUNICKE, Robin; LEBLANC, Marc; ZUBEK, Robert. Mda: A formal approach to game design and game research. In: *Proceedings of the AAAI Workshop on Challenges in Game AI*. [S.l.: s.n.], 2004. v. 4, p. 1.

KANODE, Christopher M; HADDAD, Hisham M. Software engineering challenges in game development. In: IEEE. *Information Technology: New Generations, 2009. ITNG'09. Sixth International Conference on*. [S.l.], 2009. p. 260–265.

KAPP, Karl M. *The gamification of learning and instruction: game-based methods and strategies for training and education*. [S.l.]: John Wiley & Sons, 2012.

KASURINEN, Jussi; SMOLANDER, Kari. What do game developers test in their products? In: ACM. *Proceedings of the 8th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*. [S.l.], 2014. p. 1.

KEITH, Clinton. *Agile game development with Scrum*. [S.l.]: Pearson Education, 2010.

KHATIB, Firas *et al.* Crystal structure of a monomeric retroviral protease solved by protein folding game players. *Nature structural & molecular biology*, Nature Publishing Group, v. 18, n. 10, p. 1175–1177, 2011.

KNAPP, Jake; ZERATSKY, John; KOWITZ, Braden. *Sprint: how to solve big problems and test new ideas in just five days*. [S.l.]: Simon and Schuster, 2016.

KOSTER, Raph. *Theory of fun for game design*. [S.l.]: "O'Reilly Media, Inc.", 2013.

LEBLANC, Marc *et al.* Mechanics, dynamics, aesthetics: A formal approach to game design. *lecture at Northwestern University*, 2004.

LUNDGREN, Sus; BJORK, Staffan. Game mechanics: Describing computer-augmented games in terms of interaction. In: *Proceedings of TIDSE*. [S.l.: s.n.], 2003. v. 3.

MASTROCOLA, Vicente Martin. *Game Design: Modelos de negócio e processos criativos. Uma trajetória do protótipo ao jogo produzido*. [S.l.]: Cengage Learning Nacional, 2015.

MCGONIGAL, Jane. *Reality is broken: Why games make us better and how they can change the world*. [S.l.]: Penguin, 2011.

\_\_\_\_\_. *Games*. 2014. URL: <https://janemcgonigal.com/play-me/>.

MELLO, Carlos Henrique Pereira *et al.* Pesquisa-ação na engenharia de produção: proposta de estruturação para sua condução. *Revista Produção*, SciELO Brasil, v. 22, n. 1, p. 1–13, 2012.

MICHAEL, David R; CHEN, Sandra L. *Serious games: Games that educate, train, and inform*. [S.l.]: Muska & Lipman/Premier-Trade, 2005.

MOLØKKEN-ØSTVOLD, Kjetil; HAUGEN, Nils Christian; BENESTAD, Hans Christian. Using planning poker for combining expert estimates in software projects. *Journal of Systems and Software*, Elsevier, v. 81, n. 12, p. 2106–2117, 2008.

MOORE, Gordon E. Cramming more components onto integrated circuits, reprinted from electronics, volume 38, number 8, april 19, 1965, pp. 114 ff. *IEEE Solid-State Circuits Newsletter*, v. 3, n. 20, p. 33–35, 2006.

MUSIL, Juergen *et al.* Improving video game development: Facilitating heterogeneous team collaboration through flexible software processes. In: SPRINGER. *European Conference on Software Process Improvement*. [S.l.], 2010. p. 83–94.

NEWZOO. *Top 100 Countries by Game Revenues*. 2016. Disponível em: <https://newzoo.com/insights/rankings/top-100-countries-by-game-revenues>. Acesso em: 13 mar. 2016.

NINTENDO. *A História da Nintendo*. 1989. Disponível em: <https://www.nintendo.pt/A-empresa/Historia-da-Nintendo/Game-Boy/Game-Boy-627031.html>. Acesso em: 20 mai. 2017.

ORACLE. *Distributed Planning Poker: Integrating IBM Rational Team Concert and Google Wave for distributed effort estimation*. 2014. Disponível em: [http://jazoon.com/history/Portals/0/Content/slides/we\\_a3\\_1600-1650\\_georg.pdf](http://jazoon.com/history/Portals/0/Content/slides/we_a3_1600-1650_georg.pdf). Acesso em: 20 mai. 2017.

PARLETT, David Sidney. *The Oxford history of board games*. [S.l.]: Oxford University Press, USA, 1999.

PAULK, Mark C. Extreme programming from a cmm perspective. *IEEE software*, IEEE, v. 18, n. 6, p. 19–26, 2001.

PETRILLO, Fábio *et al.* Houston, we have a problem...: a survey of actual problems in computer games development. In: ACM. *Proceedings of the 2008 ACM symposium on Applied computing*. [S.l.], 2008. p. 707–711.

PETRY, Arlete dos Santos *et al.* Parâmetros, estratégias e técnicas de análise de jogo. o caso a mansão de quelícera. In: *Simpósio Brasileiro de Jogos e Entretenimento Digital*. São Paulo, SP, Brasil: [s.n.], 2013.

PINTEREST. *O catálogo mundial de idéias*. 2010. Disponível em: <https://br.pinterest.com/pin/424112489882827959/>. Acesso em: 20 mai. 2017.

POPPENDIECK, Mary; POPPENDIECK, Tom. *Implementando o desenvolvimento Lean de Software: do conceito ao dinheiro*. [S.l.]: Bookman Editora, 2009.

RABIN, Steve. Introdução ao desenvolvimento de games. *São Paulo: Cengage Learning*, 2012.

RAILSBACK, Steven F; GRIMM, Volker. *Agent-based and individual-based modeling: a practical introduction*. [S.l.]: Princeton university press, 2011.

ROUSE, Richard. *Game design: Theory and practice*. [S.l.]: Jones & Bartlett Learning, 2010.

ROYCE, Winston W. Managing the development of large software systems. In: LOS ANGELES. *proceedings of IEEE WESCON*. [S.l.], 1970. v. 26, n. 8, p. 328–338.

SALEN, Katie; ZIMMERMAN, Eric. Regras do jogo: fundamentos do design de jogos. *São Paulo: Blucher*, v. 1, p. 69, 2012.

SANTOS, L. C. M.; SOUZA, A. C. S.; DINIZ, M. V. C. Desenvolvimento de um portal de animações em realidade aumentada para o processo de ensino-aprendizagem da língua brasileira de sinais. *Hipermídia e interdisciplinaridade na geração de conhecimento*, p. 154, 2015.

SCHANK, Roger C. *Dynamic memory revisited*. [S.l.]: Cambridge University Press, 1999.

SHELL, Jesse. *The Art of Game Design: A book of lenses*. [S.l.]: CRC Press, 2014.

SCHETINGER, Victor *et al.* User stories as actives for game development. *SBC-Proceedings of SBGames 2011*, p. 53, 2011.

SCHILD, Jonas; WALTER, Robert; MASUCH, Maic. Abc-sprints: adapting scrum to academic game development courses. In: ACM. *Proceedings of the Fifth International Conference on the Foundations of Digital Games*. [S.l.], 2010. p. 187–194.

SCHWABER, Ken; SUTHERLAND, Jeff. The definitive guide to scrum: the rules of the game. *Recuperado de: <http://www.scrumguides.org/docs/scrumguide/v1/scrum-guide-us.pdf>*, 2013.

SEBRAE. *Boletim 2014: Grandes players e pequenos negócios de Games*. 2014. Disponível em: <[http://sebrae2014.sebrae.com.br/Sebrae/Sebrae/202014/Boletins/2014\\_06\\_06\\_BO\\_Economia\\_Criativa\\_Grandes\\_players\\_pequenos\\_neg/C3/B3cios\\_de\\_Games.pdf](http://sebrae2014.sebrae.com.br/Sebrae/Sebrae/202014/Boletins/2014_06_06_BO_Economia_Criativa_Grandes_players_pequenos_neg/C3/B3cios_de_Games.pdf)>. Acesso em: 08 jul. 2016.

SHARP, Helen; ROGERS, Y; PREECE, J. Design de interação: além da interação homem-computador. *Artmed*, 2005.

SKINNER, Burrhus Frederic. *Science and human behavior*. [S.l.]: Simon and Schuster, 1953.

SOMMERVILLE, Ian. *Software engineering*. [S.l.]: Pearson, 2010.

STARFIELD, Anthony M. Qualitative, rule-based modeling. *BioScience*, JSTOR, v. 40, n. 8, p. 601–604, 1990.

SUITS, Bernard. *The Grasshopper-: Games, Life and Utopia*. [S.l.]: Broadview Press, 2014.

SUTHERLAND, Jeff. *Scrum-a arte de fazer o dobro de trabalho na metade do tempo*. [S.l.]: Leya, 2016.

TETRIS. *Tetris: Official Site*. 1996. Disponível em: <<http://tetris.com/>>. Acesso em: 20 jan. 2017.

THIOLLENT, Michel. Metodologia da pesquisa-ação. In: *Metodologia da pesquisa-ação*. [S.l.]: Cortez, 2011.

TRIPP, David. Pesquisa-ação: uma introdução metodológica. *Educação e pesquisa*, SciELO Brasil, v. 31, n. 3, p. 443–466, 2005.

UCHMAŃSKI, Janusz; GRIMM, Volker. Individual-based modelling in ecology: what makes the difference? *Trends in Ecology & Evolution*, Elsevier, v. 11, n. 10, p. 437–441, 1996.

WERBACH, Kevin; HUNTER, Dan. *For the win: How game thinking can revolutionize your business*. [S.l.]: Wharton Digital Press, 2012.

WESTBROOK, Roy. Action research: a new paradigm for research in production and operations management. *International Journal of Operations & Production Management*, MCB UP Ltd, v. 15, n. 12, p. 6–20, 1995.

WILKINSON, Phil. A brief history of serious games. In: *Entertainment Computing and Serious Games*. [S.l.]: Springer, 2016. p. 17–41.